Wireless and Mobile Network Security

Hakima Chaouchi
Maryline Laurent-Maknavicius

ISTE

WILEY

This page intentionally left blank

Wireless and Mobile Network Security

This page intentionally left blank

Wireless and Mobile Network Security

Security Basics, Security in On-the-shelf and Emerging Technologies

Edited by
Hakima Chaouchi
Maryline Laurent-Maknavicius

iSTE

WILEY

First published in France in 2007 by Hermes Science/Lavoisier in 3 volumes entitled: *La sécurité dans les réseaux sans fil et mobiles* © LAVOISIER, 2007
First published in Great Britain and the United States in 2009 by ISTE Ltd and John Wiley & Sons, Inc.

ISTE Ltd
27-37 St George's Road
London SW19 4EU
UK

www.iste.co.uk

John Wiley & Sons, Inc.
111 River Street
Hoboken, NJ 07030
USA

www.wiley.com

Library of Congress Cataloging-in-Publication Data

Sécurité dans les réseaux sans fil et mobiles. English.
Wireless and mobile network security: security basics, security in on-the-shelf and emerging technologies / edited by Hakima Chaouchi, Maryline Laurent-Maknavicius.
p. cm.
Includes bibliographical references and index.
English edition is a complete translation of the French three volumes ed. compiled into one volume in English.
ISBN 978-1-84821-117-9
1. Wireless communication systems--Security measures. 2. Mobile communication systems--Security measures. I. Chaouchi, Hakima. II. Laurent-Maknavicius, Maryline. III. Title.
TK5103.2.S438 2009
005.8--dc22
2009011422

British Library Cataloguing-in-Publication Data
A CIP record for this book is available from the British Library
ISBN: 978-1-84821-117-9

Printed and bound in Great Britain by CPI Antony Rowe, Chippenham and Eastbourne.

Table of Contents

Introduction

Wireless networks and security might be considered an oxymoron. Indeed it is hard to believe in security when it is so easy to access communication media such as wireless radio media. However, the research community in industry and academia has for many years extended wired security mechanisms or developed new security mechanisms and security protocols to sustain this marriage between wireless/mobile networks and security. Note that the mobile communication market is growing rapidly for different services and not only mobile phone services. This is why securing wireless and mobile communications is crucial for the continuation of the deployment of services over these networks.

Wireless and mobile communication networks have had tremendous success in today's communication market both in general or professional usage. In fact, obtaining communication services anytime, anywhere and on the move has been an essential need expressed by connected people. This becomes true thanks to the evolution of communication technologies from wired to wireless and mobile technologies, but also the miniaturization of terminals. Offering services to users on the move has significantly improved productivity for professionals and flexibility for general users. However, we cannot ignore the existence of important inherent vulnerabilities of these unwired communication systems, which gives the network security discipline a key role in convincing users to trust the usage of these wireless communication systems supported by security mechanisms.

Since the beginning of the networking era, security was part of the network architectures and protocols design even if it is considered to slow down the communication systems. Actually, network security is just a natural evolution of the security of stand-alone or distributed operating systems dealing with machine/network access control, authorization, confidentiality, etc. Even though the

Written by Hakima CHAOUCHI.

context has changed from wired to wireless networks, we are facing the same issues and challenges regarding security. More precisely, it is about preserving the integrity, confidentiality and availability of resources and the network. Other security issues that are more related to the users such as privacy and anonymity are also important from the user's point of view today, especially with the new need of tracking criminals, but in this book we are concerned only with network security, and as such, two chapters are included dealing with important security issues and solutions to secure downloaded applications in the mobile operator context and copyright protection by watermarking techniques.

Several security mechanisms have been developed such as authentication, encryption and access control others in order to offer secure communications over the network. According to the network environment, some security mechanisms are more mature than others due to the early stages of certain networking technologies such as wireless networks, ad hoc or sensor networks. However, even with maturity, and even if they are already widely implemented in marketed products, some security mechanisms still need some improvement. It is also important to consider the limited resources of mobile terminals and radio resources to adapt the wired network's security mechanisms to a wireless context. These limited resources have a direct impact on security design for this type of networks.

Chapter 1 offers a survey on current and emerging wireless and mobile communications coming from the mobile cellular communications such as 2G, 3G, 4G, IEEE wireless communication such as Wi-Fi, Bluetooth, WiMAX, WiMobile and WiRan, and the IP-based mobility communication such as Mobile IP or IMS. Even if security solutions always need to be improved, the deployment of these wireless and mobile networks is already effective and will tend to grow because of the growing needs of users in terms of mobility, flexibility and services. To do so, the industry and academic researchers keep on designing mobile and wireless technologies, with or without infrastructure, providing on the one hand more resources and security, and on the other hand autonomous and more efficient terminals (PDA phones, etc.).

This book is aimed at academics and industrialists, generalists or specialists interested in security in current and emerging wireless and mobile networks. It offers an up-to-date state of the art on existing security solutions in the market or prototype and research security solutions of wireless and mobile networks. It is organized into three parts.

Part 1, "Basic Concepts", offers a survey on mobile and wireless networks and the major security basics necessary for understanding the rest of the book. It is essential for novices in the field. In fact, this part describes current and emerging mobile and wireless technologies. It also introduces vulnerabilities and security mechanism fundamentals. It finally presents the vulnerabilities in wireless technology and an adaptation of copyright protection techniques in the wireless and mobile context.

Part 2, “Off-the-Shelf Technology”, looks at the issue of security of current mobile and wireless networks, namely Wi-Fi, WiMAX, Bluetooth and GSM/UMTS, and concludes with a description of the mechanisms for the protection of downloaded applications in the context of mobile operators.

Part 3, “Emerging Technologies”, focuses on the security of new communication technologies, namely the new generation of telecommunication networks such as IMS, mobile IP networks, and self-organized ad hoc and sensor networks. This last category of technologies offer very attractive applications but needs more work on the security side in order to be trusted by the users.

Finally, as we can see throughout this book, security solutions for wireless and mobile networks are either an extension of security solutions of unwired networks or a design of specific security solutions for this context. In any case, one thing is sure: at least four major constraints have to be considered in security design for wireless and mobile networks: limited radio and/or terminal resources, expected security and performance level, infrastructure or infrastructure-less architecture, and cost.

This page intentionally left blank

PART 1

Basic Concepts

Chapter 1

Introduction to Mobile and Wireless Networks

1.1. Introduction

Wireless networks in small or large coverage are increasingly popular as they promise the expected convergence of voice and data services while providing mobility to users. The first major success of wireless networks is rendered to Wi-Fi (IEEE 802.11), which opened a channel of fast and easy deployment of a local network. Other wireless technologies such as Bluetooth, WiMAX and WiMobile also show a very promising future given the high demand of users in terms of mobility and flexibility to access all their services from anywhere.

This chapter covers different wireless as well as mobile technologies. IP mobility is also introduced. The purpose of this chapter is to recall the context of this book, which deals with the security of wireless and mobile networks. Section 1.2 presents a state of the art of mobile cellular networks designed and standardized by organizations such as ITU, ETSI or 3GPP/3GPP2. Section 1.3 presents wireless networks from the IEEE standardization body. Section 1.4 introduces Internet mobility. Finally, the current and future trends are also presented.

Chapter written by Hakima CHAOUCHI and Tara ALI YAHIYA.

1.2. Mobile cellular networks

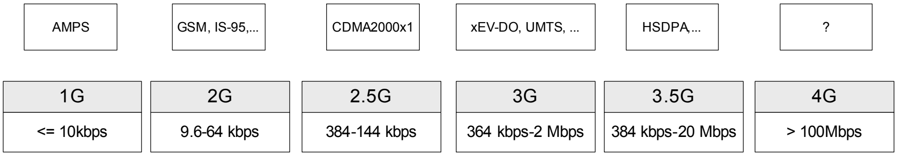
1.2.1. *Introduction*

The first generation (1G) mobile network developed in the USA was the AMPS network (Advanced Mobile Phone System). It was based on FDM (Frequency Division Multiplexing). A data service was then added on the telephone network, which is the CDPD (Cellular Digital Packet Data) network. It uses TDM (Time Division Multiplexing). The network could offer a rate of 19.2 kbps and exploit periods of inactivity of traditional voice channels to carry data. The second generation (2G) mobile network is mainly GSM (Global System for Mobile Communications). It was first introduced in Europe and then in the rest of the world. Another second-generation network is the PCS (Personal Communications Service) network or IS-136 and IS-95; PCS was developed in the USA. The IS-136 standard uses TDMA (Time Division Multiple Access) while the IS-95 standard uses CDMA (Code Division Multiple Access) in order to share the radio resource. The GSM and PCS IS-136 employ dedicated channels for data transmission.

The ITU (International Telecommunication Union) has developed a set of standards for a third generation (3G) mobile telecommunications system under the IMT-2000 (International Mobile Telecommunication-2000) in order to create a global network. They are scheduled to operate in the frequency band around 2 GHz and offer data transmission rates up to 2 Mbps. In Europe, the ETSI (European Telecommunications Standards Institute) has standardized UMTS (Universal Mobile Telecommunications Systems) as the 3G network.

The fourth generation of mobile networks is still to come (in the near future) and it is still unclear whether it will be based on both mechanisms of cellular networks and wireless networks of the IEEE or a combination of both. The ITU has stated the flow expected by this generation should be around 1 Gbps static and 100 Mbps on mobility regardless of the technology or mechanism adopted.

The figure below gives an idea of evolving standards of cellular networks. Despite their diversity, their goal has always been the same; to build a network capable of carrying both voice and data respecting the QoS, security and above all reducing the cost for the user as well as for the operator.



AMPS	GSM, IS-95,...	CDMA2000x1	xEV-DO, UMTS, ...	HSDPA,...	?
1G	2G	2.5G	3G	3.5G	4G
<= 10kbps	9.6-64 kbps	384-144 kbps	364 kbps-2 Mbps	384 kbps-20 Mbps	> 100Mbps

Figure 1.1. *The evolution of cellular networks*

1.2.2. *Cellular network basic concepts*

a) Radio resource

Radio communication faces several problems due to radio resource imperfection. In fact the radio resource is prone to errors and suffers from signal fading. Here are some problems related to the radio resource:

– Power signal: the signal between the BS and the mobile station must be sufficiently high to maintain the communication. There are several factors that can influence the signal (the distance from the BS, disrupting signals, etc.).

– Fading: different effects of propagation of the signal can cause disturbances and errors. It is important to consider these factors when building a cellular network.

To ensure communication and to avoid interference, cellular networks use signal strength control techniques. Indeed, it is desirable that the signal received is sufficiently above the background noise. For example, when the mobile moves away from the BS, the signal received subsides. In contrast, because of the effects of reflection, diffraction and dispersion, it can change the signal even if the mobile is close to the BS. It is also important to reduce the power of the broadcast signal from the mobile not only to avoid interference with neighboring cells, but also for reasons of health and energy.

As the radio resource is rare, different methods of multiplexing user data have been used to optimize its use:

– *FDMA* (Frequency Division Multiple Access) is the most frequently used method of radio multiple access. This technique is the oldest and it allows users to be differentiated by a simple frequency differentiation. Indeed, to listen to the user N, the receiver considers only the associated frequency fN. The implementation of this technology is fairly simple. In this case there is one user per frequency.

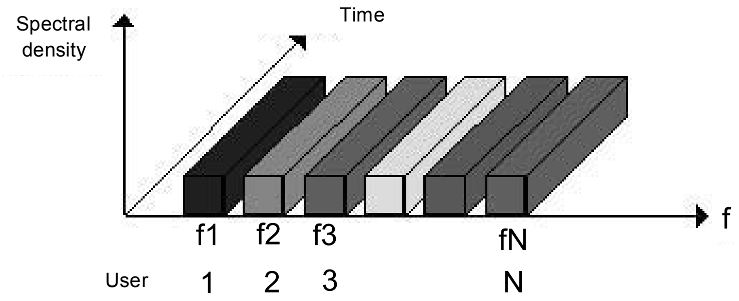


Figure 1.2. *FDMA*

– *TDMA* (Time Division Multiple Access) is an access method which is based on the distribution of the radio resource over time. Each frequency is then divided into intervals of time. Each user sends or transmits in a time interval from which the frequency is defined by the length of the frame. In this case, to listen to the user N, the receiver needs only to consider the time interval N for this user. Unlike FDMA, multiple users can transmit on the same frequency.

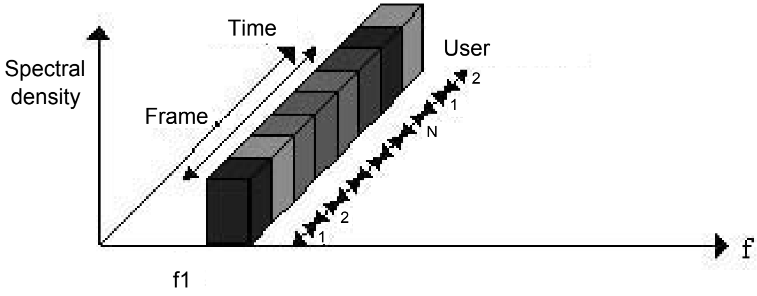


Figure 1.3. *TDMA*

– *CDMA* (Code Division Multiple Access) is based on the distribution code. It is spread by a code spectrum allocated to each communication. In fact, each user is differentiated from the rest of users with a code N allocated at the beginning of its communication and is orthogonal to the rest of the codes related to other users. In this case, to listen to the user N, the receiver has to multiply the signal received by the code N for this user.

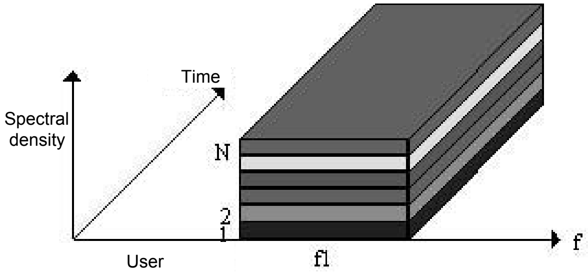


Figure 1.4. *CDMA*

The traffic uplink and downlink on the radio resource is managed by TDD (Time Division Duplex) or FDD (Frequency Division Duplex) multiplexing methods as the link is symmetric or asymmetric.

– *OFDM* (Orthogonal Frequency Division Multiplexing) is a very powerful transmission technique. It is based on the idea of dividing a given high-bit-rate datastream into several parallel lower bit-rate streams and modulating each stream on separate carriers, often called subcarriers. OFDM is a spectrally efficient version of multicarrier modulation, where the subcarriers are selected such that they are all orthogonal to one another over the symbol duration, thereby avoiding the need to have non-overlapping subcarrier channels to eliminate intercarrier interference. In order to have multiple user transmissions, a multiple access scheme such as TDMA or FDMA has to be associated with OFDM. In fact, an OFDM signal can be made from many user signals, giving the OFDMA multiple access [STA 05]. The multiple access has a new dimension with OFDMA. A downlink or uplink user will have a time and a subcarrier allocation for each of their communications. However, the available subcarriers may be divided into several groups of subcarriers called subchannels. Subchannels may be constituted using either contiguous subcarriers or subcarriers pseudorandomly distributed across the frequency spectrum. Subchannels formed using distributed subcarriers provide more frequency diversity. This permutation can be represented by Partial Usage of Subcarriers (PUSC) and Full Usage of Subcarriers (FUSC) modes [YAH 08].

b) Cell design

A cellular network is based on the use of a low-power transmitter (~100 W). The coverage of such a transmitter needs to be reduced, so that a geographic area is divided into small areas called cells. Each cell has its own transmitter-receiver (antenna) under the control of a BS. Each cell has a certain range of frequencies. To avoid interference, adjacent cells do not use the same frequencies, as opposed to two non-adjacent cells.

The cells are designed in a hexagonal form to facilitate the decision to change a cell for a mobile node. Indeed, if the distance between all transmitting cells is the same, then it is easy to harmonize the moment where a mobile node should change its cell. In practice, cells are not quite hexagonal because of different topography, propagation conditions, etc.

Another important choice in building a cellular network is the minimum distance between two cells that operate at the same frequency band in order to avoid interference. In order to do so, the cell's design could follow different schema. If the schema contains N cells, then each of them could use K/N frequencies where K is the number of frequencies allocated to the system.

The value of reusing frequencies is to increase the number of users in the system using the same frequency band which is very important to a network operator.

In the case where the system is used at its maximum capacity, meaning that all frequencies are used, there are some techniques to enable new users in the system. For instance, adding new channels, borrowing frequency of neighboring cells, or cell division techniques are useful to increase system capacity. The general principle is to have micro and pico (very small) cells in areas of high density to allow a significant reuse of frequencies in a geographical area with high population.

c) Traffic engineering

Traffic engineering was first developed for the design of telephone circuit switching networks. In the context of cellular networks, it is also essential to know and plan to scale the network that is blocking the minimum mobile nodes, which means accepting a maximum of communication. When designing the cellular network, it is important to define the degree of blockage of the communications and also to manage incoming blocked calls. In other words, if a call is blocked, it will be put on hold, and then we will have to define what the average waiting time is. Knowing the system's ability to start (number of channels) will determine the probability of blocking and the average waiting time of blocked requests.

What complicates this traffic engineering in cellular networks is the mobility of users. In fact, a cell will handle, in addition to new calls, calls transferred by neighboring cells. The traffic engineering model becomes more complex. Another parameter that is even more complicating for the model is that the system should accommodate both phone calls as data traffic, knowing that they have very different traffic characteristics.

d) Cellular system's elements

A cellular network is generally composed of the following:

– BSs: situated at the heart of the cell, a BS includes an antenna, a controller and a number of transmitters and receivers. It allows communications on channels assigned to the cell. The controller allows the management of the call request process between a mobile and the rest of the network. The BS is connected to a mobile switching center (MTSO: Mobile Telephone Switching Office). Two types of channels are established between the mobile and the BS: the data channel and the traffic control channel. The control channels are used for associating the mobile node with the BS nearest to the exchange of information necessary to establish and maintain connections. The traffic channels used to transport the user traffic (voice, data, etc.).

– Mobile switching center (MTSO): a MTSO manages several BSs generally bound by a wired network. It is responsible for making connections between mobiles. It is also connected to the wired telephone network and is thus able to establish connections between mobiles and fixed nodes. The MTSO is responsible for the allocation of channels for each call request and is also responsible for handover and recording the billing information of active call users.

The call process includes the following functions:

– Initializing a mobile: once the mobile node is turned on, it scans the frequency channels, then it selects the strongest control call channel (setup). Each cell regularly controls the information on the band corresponding to its control channel. The mobile node selects the channel whose signal is the most important. Then the phone goes through a phase of identification with the cell (handshake). This phase occurs between the mobile and the MTSO. The mobile is identified following an authentication and its location is recorded. The mobile continues to regularly scan the frequency spectrum and decides to change the BS if it has a stronger signal than the previous cell phone. The mobile node also remains attentive to the call notification.

– Call initiated by a mobile node: the mobile node checks that the call channel is free by checking the information sent by the BS on the downlink control channel. The mobile may then issue the call number on the uplink control channel to the BS that transmits the request to MTSO.

– Call notification: the phone number is received, the switching center tries to connect to BSs concerned by the number and sends a call notification message to the called mobile node (paging). The call notification is retransmitted by BSs in the downlink control channel.

– Acceptance of call: the mobile recognizes its number in the call control channel and then responds to the BS to relay the message to the switch that will establish a circuit between the BSs of the calling and the called nodes. The switch will also select an available traffic channel in each of the two cells involved and sends the information related to that call to the BSs. The phones will then synchronize the traffic channels selected by the BS.

– Active communication: this is the process of exchanging data or voice traffic between the calling and called mobiles. This is assured by both BSs and the switching center.

– Call blocking: if all channels of traffic in a BS are occupied, the mobile will try a number of pre-configured times to repeat the call. In case of failure, an "occupied" signal tone is returned to the user.

– Call termination: at the end of a communication, the switching center informs the BSs to free channels. This action is also important for billing.

– Abandonment of call: during a communication, if the BS fails to maintain a good level of signal (interference, low signal, etc.) it abandons the channel traffic of the mobile and notifies the switching center.

– Call between a fixed terminal and a mobile node: the switching center being connected to the landline or fixed network, it is then able to establish communication between these two networks. It can also join another mobile switching center through the fixed network.

– Handover (Handoff): when the mobile discovers a control channel where the signal is stronger than its current cell, the network will automatically change to the cell by transferring its mobile channel call to the new cell without the user noticing. The main criterion used to take the decision to transfer the mobile is the measured signal power of the mobile node by the BS. In general, the station calculates an average over a time window to eliminate the rapid fluctuations resulting from multipath effects. Various techniques can be used to determine the moment of transfer of the mobile. In addition, this transfer can be controlled by either the network or the mobile. The simplest technique of handover decision is one that triggers the transfer as soon as the mobile detects a new signal stronger than the cell where it is connected.

1.2.3. *First generation (1G) mobile*

First generation cellular networks such as CT0/1 (Cordless Telephone) for wireless and AMPS (Advanced Mobile Phone Service) for mobile communications were first characterized by analog communications. The first cellular networks are virtually non-existent today. The AMPS system was the 1st generation of the most widespread used network in the USA up to the 1980s. It has also been deployed in South America, Australia and China. In Northern Europe, the NMT (Nordic Mobile Telecommunications System) was developed. In the UK, the TACS (Total Access

Communication System) and Radio France in 2000 were deployed. All these cellular networks were 1G analog and used frequency bands around 450 and 900 MHz.

1.2.4. *Second generation (2G) mobile*

Cellular networks such as second generation DECT for wireless and mobile phones for mobile were characterized by digital communications networks, unlike the first generation, which were analog. During the 1990s several digital technologies were developed:

– GSM (Global System for Mobile Communication), developed in Europe, operating at 900 MHz.

– DCS 1800 (Digital Cellular System) equivalent to GSM but operating at higher frequencies (1,800 MHz).

– PCS 1900 (Personal Communication System) and D-AMPS (Digital AMPS) developed in the USA.

– Finally, PDC (Pacific Digital Cellular) developed in Japan.

The GSM and D-AMPS (also called IS-136) were based on the TDMA access method while the PCS 1900, also called IS-95 or cdmaOne, was based on CDMA technology.

A simple transmission of data is possible in addition to the voice but the rate remains low with less than 10 kbps and certainly did not make possible the deployment of multimedia services. Thus, HSCSD (High Speed Circuit Switched Data) and GPRS (General Packet Radio Service) are techniques that have helped increase the flow of 2G networks. These technologies are also known as 2.5 generation cellular networks. GPRS, unlike HSCDC, uses packet switching to optimize the radio resource transmission of data traffic that is sporadic in nature. The theoretical speed is 120 kbps while the real flow does not exceed 30 kbps. This generation cannot meet the needs of mobile users who want multimedia services comparable to fixed networks. The evolution of the GPRS network led to EDGE (Enhanced Data rates for GSM Evolution) or Enhanced GPRS (EGPRS), which has improved the reliability and speed of data transmission. It is generally known as 2.75G or 3G depending on its implementation. This is a simple evolution of GSM/GPRS to achieve average speeds of 130 kbps downstream and 60 kbps in transmission, 6 to 10 times greater than GPRS.

Mobility management is usually done using two databases: the HLR (Home Location Register) which maintains the data of the subscriber and the VLR (Visitor Location Register) which manages the customer in the visited cell. Using these two components, the network can manage the location of mobile node to be able to route its calls and also ensure the handover. These networks allow high mobility of the terminal but low personal mobility leading to the possibility of using the SIM (Subscriber Identity Module) in any terminal. Remember that personal mobility is the ability to change terminal while maintaining its working environment or session. We find such mobility for example in UPT (Universal Personal Telecommunication) networks.

1.2.5. *Third generation (3G) mobile*

3G cellular networks operate around the frequency band of 2 GHz, providing a range of multimedia services to fixed and mobile users with a Quality of Service almost comparable to that of fixed networks. The International Telecommunications Union (ITU) has selected five standards for 3G mobile under the symbol IMT-2000 (International Mobile Telecommunications system for the year 2000). This is the W-CDMA (Wideband CDMA), TD-CDMA and TD-SCDMA standard used in the European UMTS (Universal Mobile Telecommunication System) of CDMA2000, EDGE (Enhanced Data rate for GSM Evolution) and the third generation of DECT. The IMT-2000 are designed to include global roaming, a range of broadband services such as video and the use of a single terminal in different wireless networks (vertical mobility). Another objective is to make fixed services and mobile services compatible in order to be transparent to the user. These networks offer a comprehensive mobility which includes a terminal mobility, personal mobility and service mobility. The concept of VHE (Virtual Home Environment) is developed to support the service mobility. In addition to larger bandwidth, global mobility is another major difference compared to 2G networks.

UMTS based on the W-CDMA access method theoretically allows the transfer rates of 1.920 Mbps, almost 2 Mbps but at the end of 2004 rates offered by operators rarely exceeded 384 kbps. However, this speed is much higher than the base flow of GSM, which is 9.6 kbps. UMTS based on the TDD access method is not compatible with UMTS TD-CDMA. The 3G network development in China is based on a TD-SCDMA (Time Division-Synchronous Code Division Multiple Access) local standard to avoid paying for the rights of other 3G standards.

In the family of CDMA2000 standards, we find CDMA2000 1x, CDMA2000 1xEV-DO and CDMA2000 1xEV-DV which are direct successors of CDMA 2G (cdmaOne, IS-95); these are 3GPP1 standards. CDMA2000 1x, known under the terms 1x, 1xRTT, IS-2000, CDMA2000 1X, 1X and cdma2000 (CDMA lowercase), double the capacity of the voice compared to IS-95. The data transmission could reach 144 kbps. 1xRTT is considered to be 2.5G, 2.75G or 3G under implementation. CDMA2000 3x was specified on another frequency band – this standard has not been deployed. Finally, 1xEV-DO or IS-856 and 1xEV-DV were designed to increase the speed of data transmission and support mobile video. In the HSDPA (High Speed Access Protocol) family which is the evolution of the UMTS to a new wireless broadband network. Data transmission protocols are the HSDPA, HSUPA and HSOPA, which are the successors of UMTS. HSUPA (High-Speed Uplink Packet Access) could bear a rate of 5.76 Mbps. HSDPA (High-Speed Downlink Protocol Access) in the first phase of its development could attain 14 Mbps. In the second phase of its development HSDPA could support up to 28.8 Mbps using MIMO (Multiple Input Multiple Output) technology and beam forming. HSOPA (High Speed OFDM Packet Access), HSDPA's successor, is also known as 3GPP LTE (Long Term Evolution), the goal of which is to reach 100 Mbps downlink and 50 Mbps on the uplink through access technology OFDMA. It is in direct competition with technologies such as WiMAX IEEE. HSOPA is a new air interface incompatible with W-CDMA and therefore with the previous developments of 3G networks.

1.3. IEEE wireless networks

1.3.1. *Introduction*

Many standards for wireless communication are being developed day after day and the price of their equipment becomes increasingly attractive. This will contribute to the success of these technologies. In this section, we introduce the standards that are the basis of many wireless networks.

Standard	Description
802.11a	This standard is an amendment to the IEEE 802.11 specification that added a higher throughput of up to 54 Mbit/s by using the 5 GHz band. IEEE 802.11a specifies 8 operating channels in this frequency band.
802.11b	This standard uses the radio signaling frequency (2.4 GHz) as the original 802.11 standard with 13 channels in France. This standard allows a range of 300 m in an outdoor environment.
802.11e	This standard defines a set of Quality of Service enhancements for wireless LAN applications through modifications to the Media Access Control (MAC) layer. Such enhancement allows the best transmission quality for voice and video applications
802.11f	This standard (also known as the Inter-Access Point Protocol) is a recommendation that describes an optional extension to IEEE 802.11, which provides wireless access-point communications among multi-vendor systems. This protocol allows the users to change their access point when handover occurs.

802.11g	This is a set of standards for wireless local area network (WLAN) computer communications operating in the 5 GHz and 2.4 GHz public spectrum bands.
802.11i	This, is an amendment to the IEEE 802.11 standard specifying security mechanisms for wireless networks. IEEE 802.11i makes use of the Advanced Encryption Standard (AES) block cipher, whereas WEP and WPA use the RC4 stream cipher. It proposes different type of encryption protocols for transmission.
802.11k	This is an amendment to the IEEE 802.11-2007 standard for radio resource management. It defines and exposes radio and network information to facilitate the management and maintenance of a mobile wireless LAN. In a network conforming to 802.11k, if the access point (AP) has the strongest signal is loaded to its full capacity, a wireless device is connected to one of the underused APs. Even though the signal may be weaker, the overall throughput is greater because more efficient use is made of the network resources.
802.11n	This is a proposed amendment which improves upon the previous 802.11 standards by adding MIMO and many other newer features. It improves significantly network throughput increase in the maximum raw (PHY) data rate from 54 Mbit/s to a maximum of 600 Mbit/s.
802.15.1	This covers Bluetooth technology.
802.15.3	IEEE 802.15.3a is an attempt to provide a higher speed UWB (Ultra-Wide Band) physical layer enhancement amendment to IEEE 802.15.3 for applications which involve imaging and multimedia.
802.15.4	This is the basis for ZigBee, WirelessHART and MiWi specification, which further attempts to offer a complete networking. It offers a low data rate with a low price.
802.16a	This specifies the global deployment of broadband Wireless Metropolitan Area Networks. It delivers a point to multipoint capability in the 2-11 GHz band. The standard is extended to include OFDM and OFDMA.
802.16d	This is the revision standard for the 802.16 and 802.16a.
802.16e	This standard adds the mobility capability to IEEE 802.16d by adding advanced features to the MAC and PHY layers.
802.20	This standard (also known as Mobile Broadband Wireless Access (MBWA)) enables worldwide deployment of affordable, ubiquitous, always-on and interoperable multi-vendor mobile broadband wireless access networks that meet the needs of business and residential end-user markets.
802.21	This standard (also known as Media Independent Handover (MIH)) is developing standards to enable handover and interoperability between heterogenous network types including both 802 and non-802 networks.

802.22	This standard (also known as Wireless Regional Area Networks (WRAN)) aims to develop a standard for a cognitive radio-based PHY/MAC/air interface for use by license-exempt devices on a non-interfering basis in a spectrum that is allocated to the TV broadcast service.

Table 1.1. *The different IEEE 802 standards*

1.3.2. *WLAN: IEEE 802.11*

The IEEE 802.11 standard describes the wireless area network characteristics. Wi-Fi (Wireless Fidelity) corresponds initially to the name give to a certification delivered by the Wi-Fi Alliance which is a consortium of separate and independent companies that agrees on a set of common interoperable products based on the family of IEEE 802.11 standards.

The IEEE 802.11 can operate in two modes: infrastructure and ad-hoc. In the ad hoc mode or infrastuctureless mode, two WLAN stations can communicate directly with each other whenever they are in the same range spectrum without the intervention of the access point. Each WLAN station can be considered as an access point and a client station at the same time. However, in the infrastructure mode, the wireless network is controlled by the access point which is equipped with two interface networks:

– One wireless interface by which it receives all the exchanged frames in the cell and over which it retransmits the frames to the destination station in the cell.

– The second interface, which is ethernet, is used for communication with other access points or used for accessing the Internet.

The set of all WLAN stations that can communicate with each other is called the basic service set (BSS). The distribution system (DS) connects more than one BSS and forms an extended service set. The concept of a DS is to increase network coverage through roaming between cells.

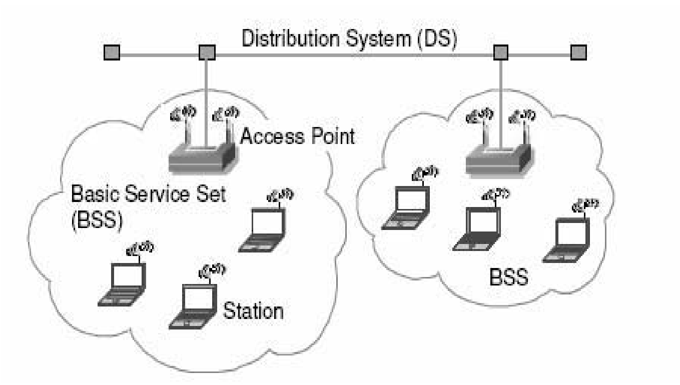


Figure 1.5. *WLAN-infrastructure mode*

a) Wi-Fi architecture

Similarly to all IEEE standards, the IEEE 802.11 specifications address both the Physical (PHY) and Media Access Control (MAC) layers and are tailored to resolve compatibility issues between manufacturers of WLAN equipment. The MAC layer can be a common layer for the different types of physical layer adopted by this standard. This can be done without any modification to the MAC layer.

b) The PHY layer

Three PHY layers were defined initially for IEEE 802.11:

1) DSSS (Direct Sequence Spectrum): the principle of this is to spread a signal on a larger frequency band by multiplexing it with a signature or code to minimize localized interference and background noise. To spread the signal, each bit is modulated by a code. In the receiver, the original signal is recovered by receiving the whole spread channel and demodulating with the same code used by the transmitter. The 802.11 DSSS PHY also uses the 2.4 GHz radio frequency band.

2) FHSS (Frequency Hopping Spread Spectrum): this utilizes a set of narrow channels and "hops" through all of them in a predetermined sequence. For example, the 2.4 GHz frequency band is divided into 70 channels of 1 MHz each. Every 20 to 400 ms the system "hops" to a new channel following a predetermined cyclic pattern. The 802.11 FHSS PHY uses the 2.4 GHz radio frequency band, operating at a 1 or 2 Mbps data rate.

3) Infrared: the Infrared PHY utilizes infrared light to transmit binary data either at 1 Mbps (basic access rate) or 2 Mbps (enhanced access rate) using a specific modulation technique for each. For 1 Mbps, the infrared PHY uses a 16-

pulse position modulation (PPM). The concept of PPM is to vary the position of a pulse to represent different binary symbols. Infrared transmission at 2 Mbps utilizes a 4 PPM modulation technique.

c) MAC layer and channel access method

The principal function of the MAC layer is to control the access to the medium. The IEEE 802.11 adopted two algorithms of controlling access to the channel: DCF (Distributed Coordination Function) and PCF (Point Coordination Function).

The default method of access is DCF, which is designed to support asynchronous best effort data. Nowadays, the IEEE 802.11 works on this mode only. Fundamentally, the DCF deploys the CSMA/CA (Carrier Sense Multiple Access/Carrier Avoidance) algorithm. The most important part of this algorithm is the process of backoff which is applied before any frame transmission.

Whenever a WLAN station wants to sent data, it first senses the medium. If the later is idle, then the WLAN station will transmit its data, otherwise it changes its transmission. After detecting the medium being idle over a period of time DIFS (Distributed Interframe Spaces), the WLAN station will continue to listen to the medium during a supplementary random time called the backoff period. The frame then will be transmitted if the medium is idle after the expiration of the backoff period.

The duration of backoff is determined by the CW (Contention Window) which has a value bounded by [CWmin, CWmax] maintained separately in each WLAN station in the BSS. A slotted backoff time is generated randomly by each WLAN station in the interval of [0, CW]. If the medium is still idle, the backoff time will be decremented slot by slot and this process will be continued as long as the medium is idle. When the backoff time reaches 0, the WLAN station will transmit the frame. If the medium is occupied during the process of backoff, the countdown to backoff will be suspended. There it restarts with the residual values when the medium is idle for one consecutive DIFS.

Whenever the frame received well by the recipient, the latter will send an acknowledgement (ACK) message to the sender. If the WLAN station does not receive the ACK, it deduces that there were a collision and in order to avoid consecutive collisions, it will retransmit the same frame. The value of the CW will be doubled in the case of transmission failure.

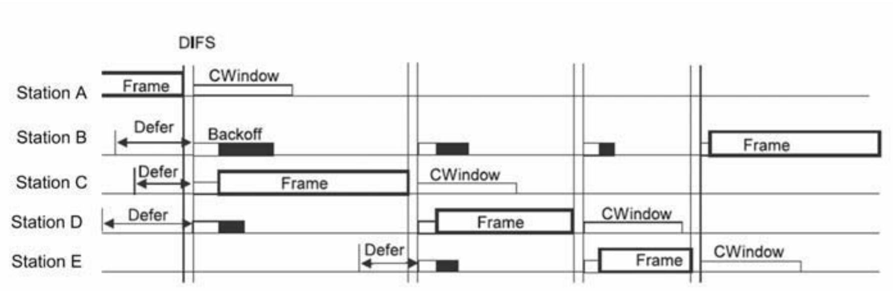


Figure 1.6. *Backoff algorithm*

The PCF method, also called the controlled access mode, is based on a polling method which is controlled by the access point. A WLAN station cannot transmit if it is not authorized and it cannot receive only if it is selected by the access point. This method is conceived for the real-time applications (voice and video) that demand delay management when transmitting data. This system is reservation-based access. However, this method of operation is optional and not mandatory, just like DCF, and it is applicable only in the infrastructure mode. Thus, the access point controls the access to the medium and authorizes or not the WLAN station to send data. It defines also the Point Coordination (PC) which determines two types of time periods, with or without contention:

– **Contention Period (CP):** corresponding to a period of time with contention in which the DCF method is used to access the medium.

– **Contention Free Period (CFP):** corresponding to a period of time without contention in which the PCF method is used to access the medium.

The duration of CFP-MaxDuration is defined by the access point. The CFP periods are initialized when the beacon is emitted by the access point. During CFP-Max, the OCF method will be active, while in the residual time, the DCF method is used. In order to switch between the PCF and DCF method, a super frame is used in order to make it possible to mote the repetition period within the mode without contention (PCF).

– **IEEE 802.11a, b, g:** the IEEE 802.11 standard is published in four phases. Firstly, it is called 802.11, which included MAC and three specifications of physical layers (two of them operating in the 2.4 GHz band, and one using infrared). The IEEE 802.11b standard was then published. This operates in the 2.4 GHz band with the data rate of 5.5 and 11 Mbit/s. Afterwards, the IEEE 802.11g standard is specified in the 2.4 GHz band, but with a data rate of 54 Mbit/s. The wireless network based on 802.11b and 802.11g is compatible in the uplink direction. Thus, a 802.11g wireless card can be connected to the 802.11b network using the data rate of 11 Mbit/s, while the contrary is not possible. For the physical part, the following propositions are kept for the wireless network based on 802.11a: frequency band of 5 GHz without license use, OFDM with 52 subcarriers, which has a very good performance in terms of multipath resistance and high data rate from 6 to 54 Mbit/s. The higher layer is represented by the MAC layer which controls the CSMA/CA algorithm.

– **IEEE 802.11e and f:** the IEEE 802.11 standard is intended to support only best effort service; however, IEEE 802.11e introduced basic QoS support by defining four different access categories (ACs), namely AC_VO (voice) with highest priority, AC_VI (video), AC_BE (best effort) and AC_BK (background) with lowest priority. Actually, in CSMA/CA all WLAN stations compete for the channel with the same priority. There is no differentiation mechanism to provide better service for real-time multimedia traffic than for data applications. This is the reason behind introducing the hybrid coordination function in IEEE 802.11e which consists of two different methods of medium access, which uses the concepts of Traffic Opportunity (TXOP), referring to a time duration during which a WLAN station is allowed to transmit a burst of data frames: EDCA (Enhanced Distributed Channel Access) and HCCA (Controlled Channel Access).

The EDCA method is where each AC behaves as a single DCF contending entity with its own contention parameters (CWmin, CWmax, AIFS and TXOP), which are announced by the AP periodically in beacon frames. Basically, the smaller the values of CWmin, CWmax and AIFS[AC], the shorter the channel access delay for the corresponding AC and the higher the priority for access to the medium. In EDCA a new type of IFS is introduced, the Arbitrary IFS (AIFS), instead of DIFS in DCF. Each AIFS is an IFS interval with arbitrary length as follows: AIFS = SIFS + AIFSNx slot time, where AIFSN is called the arbitration IFS number. After sensing the medium has been idle for a time interval of AIFS[AC], each AC calculates its own random backoff time (CWmin[AC] <= backoff time<= CWmax[AC]). The purpose of using different contention parameters for different queues is to give a low priority class a longer waiting time than a high priority class, so the high-priority class is likely to access the medium earlier than the low-priority class.

The polling-based HCCA method is where different traffic classes called traffic streams (TSs) are introduced. Before any data transmission, a TS is first established, and each WLAN station is allowed to have no more than eight TSs with different priorities. In order to initiate a TS connection, a WLAN station sends a QoS request frame containing a traffic specification (TSPEC) to the AP. A TSPEC describes the QoS requirements of a TS, such as mean/peak data rate, mean/maximum frame size, delay bound and maximum Required Service Interval (RSI). On receiving all these

QoS requests, the AP scheduler computes the corresponding HCCA-TXOP values for different WLAN stations by using their QoS requests in TSPECs (TXOP1, TXOP2, etc.) and polls them sequentially.

IEEE 802.11f treats the problem of interoperability among access points of different manufacturers. This standard facilitates the handover process of WLAN stations from one access point to another while maintaining the current traffic transmission.

– **IEEE 802.11k:** this is a proposed standard for how a WLAN should perform channel selection, roaming and transmit power control (TPC) in order to optimize network performance. It is intended to improve the way traffic is distributed within a network. In a WLAN, each device normally connects to the AP that provides the strongest signal. Depending on the number and geographic locations of the subscribers, this arrangement can sometimes lead to excessive demand on one AP and underutilization of others, resulting in degradation of overall network performance. In a network conforming to 802.11k, if the AP having the strongest signal is loaded to its full capacity, a wireless device is connected to one of the underutilized APs. Even though the signal may be weaker, the overall throughput is greater because more efficient use is made of the network resources.

– **IEEE 802.11i:** also called WP2, this is a standard for WLANs that provides improved encryption for networks that use the popular 802.11a, 802.11b (which includes Wi-Fi) and 802.11g standards. The 802.11i standard requires new encryption key protocols, known as the Temporal Key Integrity Protocol (TKIP) and Advanced Encryption Standard (AES). The 802.11i standard was officially ratified by the IEEE in June 2004 and thereby became part of the 802.11 family of wireless network specifications. The 802.11i specification offers a level of security sufficient to satisfy most government agencies. However, AES requires a dedicated chip, and this may mean hardware upgrades for most existing Wi-Fi networks. Other features of 802.11i are key caching, which facilitates fast reconnection to the server for users who have temporarily gone offline, and pre-authentication, which allows fast roaming and is ideal for use with advanced applications such as Voice-over Internet Protocols (VoIPs).

– **IEEE 802.11n:** in January 2004, IEEE announced that it would constitute a new working group (TGn) 802.11 for developing a new amendment to the IEEE 802.11 standard for wireless network. They estimated that the data rate would attain 540 Mbit/s. This is done by adding MIMO and channel-bonding/40 MHz operation to the PHY layer, and frame aggregation to the MAC layer. MIMO uses multiple transmitter and receiver antennas to improve system performance. MIMO is a technology which uses multiple antennas to coherently resolve more information than possible using a single antenna. Two important benefits are provided by 802.11n: antenna diversity and spatial multiplexing.

1.3.3. *WPAN: IEEE 802.15*

The 802.15 WPAN efforts focus on the development of consensus standards for Personal Area Networks or short distance wireless networks. These WPANs address wireless networking of portable and mobile computing devices such as PCs, Personal Digital Assistants (PDAs), peripherals, cell phones, pagers and consumer electronics, allowing these devices to communicate and interoperate with one another in a small range. Initially, this standard was developed in 1999 with an aim of enabling communication over short distances. In this group, three subgroups were initiated in parallel:

1) IEEE 802.15.1, the most well known standard which is the basis of Bluetooth technology.

2) IEEE 802.15.3, which defined UWB technology.

3) IEEE 802.15.4, the basis of Zigbee specification; the aim of this work group was to provide a solution for WPAN with a low data rate also considering the power consumption issue.

a) Bluetooth

Bluetooth was scheduled to operate in environments involving many users. There may be up to eight pieces of equipment which communicate with each other in a small network called piconet. Two or more piconets that include one or more devices participating in more than one piconet are called scatternets. The communication between the peices of equipment is coded and protected against intruders and interference. The Bluetooth equipment uses a 2.4 GHz band, available universally without license in almost all European countries and the USA. A 179 MHz channel is allocated, while only 23 channels are allocated in France, Spain, and Japan. Channel access is made by the FHSS technique with a data rate of 1 Mbps using only one type of modulation, Gaussian-shaped frequency shift keying (GFSK).

Connection scheme

Bluetooth equipment can operate either in master or slave mode. The first connection is to interconnect a maximum of eight pieces of equipment in which seven of them are slaves and one of them is the master. All of them operate together to form a piconet which is the basic and the simplest configuration of a bluetooth network.

The second connection is to interconnect piconets in order to connect one to another, forming a scatternet. The scatternet is a topology in which a multi-hop wireless network can be created. Two piconets can communicate with each other using a common node. A node can be master in a piconet and a slave in another.

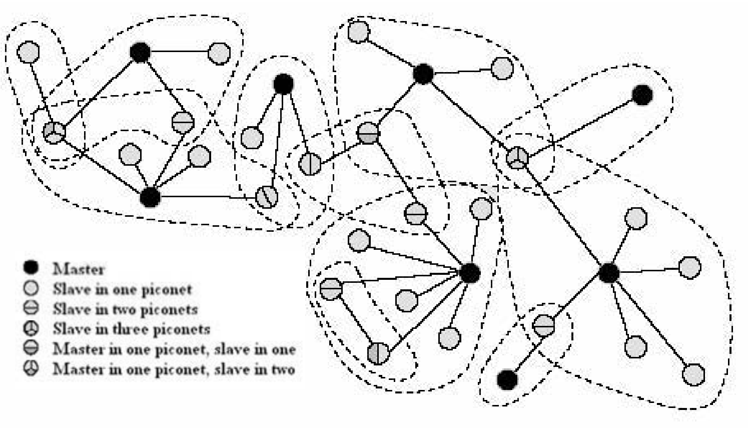


Figure 1.7. *A complex configuration of a scatternet*

Communications

When a piece of equipment (slave) enters the piconet, it waits for an inquiry message from the master node in order to obtain the address of the master and the phase clock, which is used to calculate the hop sequences. The time is divided into slots, with 1,600 slots per second. One slot is 625 ms. A Bluetooth slave uses all the frequency bands in a cyclical manner. The slaves of the same piconet possess the same frequency sequence and when a new slave is connected, it should start knowing the set of frequency hops in order to respect the timing. The master starts its transmissions in the pair slots, while the slaves use the odd slots. Message duration can be between three and five consecutive slots.

Two different types of communications are defined in Bluetooth: asynchronous connectionless links (ACLs) and synchronous connection-oriented links (SCOs). The SCO provides a guaranteed delay and bandwidth. One slave can open three SCOs with the same master or two SCOs with two different masters, while the master can open three SCOs with three different slaves. The SCO provides a symmetric channel with CBR, which is suitable for application with symmetric bandwidth constraints.

SCOs provide a limited reliability; no retransmission is achieved and no CRD is applied on the payload, although they are optionally protected with forward error correction (FEC) of 1/3 or 2/3 and convolutional code. SCOs allow a synchronous data rate of 64 Kbit/s. ACL links are convenient for real time traffic. One slave can exchange a packet with the master according to the scheduling between slaves which is calculated by the master. An ACL can exist only between slaves and the master, which signifies that the application requirements of different parameters of QoS cannot be accommodated. The ACL link can reach 732.2 kbps.

b) UWB and Zigbee

The purpose of IEEE 802.15.3 is to provide low complexity, low cost, low power consumption and high data rate wireless connectivity among devices within or entering the personal operating space. The data rate is high enough (20 Mb/s or more) to satisfy a set of consumer multimedia industry needs for WPAN communications. This standard also addresses the QoS capabilities required to support multimedia data types. This standard is the basis of WiMedia that adopted UWB technology for multimedia personal wireless networks. The objective of Wireless USB is to replace the metallic interface of USB2 with a wireless interface data rate of 480 Mbit/s.

One of the most important characteristics of UWB is to enable the communication among devices that move at a slow speed. The topology of UWB is similar to that of Bluetooth consisting of piconet and scatternets. Also, the UWB can work in ad hoc mode and provide the QoS using the TDMA technique and using determined number of slots for the different simultaneous connections.

The Zigbee network is conceived to consume less energy but only with low mobility, unlike the UWB network. This standard specifies two layers: the physical layer based on DSSS in the frequency band of 868/915 MHz with a data rate of 20 and 40 kbps, and the physical layer based on DSSS in the frequency band of 2.4 GHz with a data rate of 250 kbps.

1.3.4. *WMAN: IEEE 802.16*

Emerging technologies such as WiMAX (Worldwide Interoperability for Microwave Access) which is based on IEEE 802.16 are profoundly changing the landscape of wireless broadband. This is because providing last mile connectivity to a backbone network (such as the Internet) continues to be a challenge of fundamental importance for the evolution of next generation wireless networks. This is due to the variety of fundamentally different design options. For example, there are multiple physical layer (PHY) choices: a single-carrier-based physical layer called Wireless-MAN-SCa, an OFDM-based physical layer called Wireless MAN-OFDM, and an OFDMA-based physical layer called Wireless-OFDMA. Similarly, there are multiple choices for MAC architecture, duplexing, frequency band of operation, etc.

However, for practical reasons of interoperability, the scope of the standard needs to be reduced, and a smaller set of design choices for implementation need to be defined. The WiMAX Forum does this by defining a limited number of system and certification profiles.

a) The MAC layer

The MAC layer of Mobile WiMAX provides a medium-independent interface to the PHY layer and is designed to support the wireless PHY layer by focusing on efficient radio resource management. The MAC layer supports both Point-to-Multipoint (PMP) and mesh network modes and is divided into three sublayers: the *service-specific convergence* sublayer, the *common part* sublayer and the *security* sublayer. The primary task of the *service-specific convergence* sublayer is to classify external Service Data Units (SDUs) and associate each of them with a proper MAC service flow (SF) identifier and connection identifier. The common part sublayer function is to (i) segment or concatenate the SDUs received from higher layers into the MAC Protocol Data Units (PDUs), (ii) retransmit MAC PDUs that were received erroneously by the receiver when Automated Repeat Request (ARQ) is used, (iii) provide QoS control and priority handling of MAC PDUs belonging to different data and signaling bearers, and (iv) schedule MAC PDUs over the PHY resources. The *security* sublayer handles authentication, secure key exchange and encryption.

Channel Access Mechanism

In WiMAX, the MAC layer at the BS is fully responsible for allocating bandwidth to all Mobile Stations (MSs), in both uplink and downlink. It supports several mechanisms by which an MS can request and obtain uplink bandwidth. Depending on the particular QoS and traffic parameters associated with a service, one or more of these mechanisms may be used by the MS. The BS allocates dedicated or shared resources periodically to each MS, which it can use to request bandwidth. This process is called polling. Mobile WiMAX defines a contention access and resolution mechanism for the case when more than one MS attempts to use the shared resource. If it already has an allocation for sending traffic, the MS is not polled. Instead, it is allowed to request more bandwidth by (i) transmitting a stand-alone bandwidth request or (ii) piggybacking a bandwidth request on generic MAC packets.

Quality of Service

Support for QoS is a fundamental part of the mobile WiMAX MAC layer design strong QoS control is achieved by using a connection-oriented MAC architecture, where all downlink and uplink connections are controlled by the serving BS. Before any data transmission happens, the BS and the MS establish a unidirectional logical link, called a connection, between the two MAC-layer peers. Each connection is identified by a Connection Identifier (CID), which serves as a temporary address for data transmission over the particular link. Mobile WiMAX also defines a concept of a *service flow.* An SF is a unidirectional flow of packets with a particular set of QoS parameters and is defined by a service flow identifier (SFID). To support a variety of applications, mobile WiMAX defines four SFs:

1) Unsolicited grant services (UGS): this is designed to support fixed-size data packets at a Constant Bit Rate (CBR). Examples of applications that may use this service are T1/E1 emulation and VoIP without silence suppression. The SF parameters that define this service are maximum sustained traffic rate, maximum latency, tolerated jitter and request/transmission policy.

2) Real-time polling services (rtPS): this service is designed to support real-time SFs such as MPEG video, that generate variable-size data packets on a periodic basis. The mandatory SF parameters that define this service are minimum reserved traffic rate, maximum sustained traffic rate, maximum latency and request/transmission policy.

3) Non real-time polling service (nrtPS): this service is designed to support delay-tolerant data streams, such as an FTP, that require variable-size data grants at a minimum guaranteed rate. The mandatory SF parameters to define this service are minimum reserved traffic rate, maximum sustained traffic rate, traffic priority and request/transmission policy.

4) Best-effort (BE) service: this service is designed to support data streams, such as Web browsing, that do not require a minimum service-level guarantee. The mandatory SF parameters to define this service are maximum sustained traffic rate, traffic priority and request/transmission policy.

b) The physical layer

The first characteristic of the physical layer is to have a different structure of channel in the uplink and downlink directions. Since the physical layer is based on Wireless MAN-OFDM A 256-carrier orthogonal-frequency division multiplexing (OFDM) scheme. Thus, the multiple access of different subscriber stations (SSs) is TDMA-DAMA (Time Division Multiple Access-Demand Assignment Multiple

Access). With the TDMA-DAMA the allocation of time slots will be achieved dynamically. While in the downlink, the transmission mode will be in two modes: traffic flow continue and sporadic flow. In the first mode, the TDM technique is used for channel access. The mechanism used for duplexing is FDD in order to share resources between downlink and uplink channel. In the second mode, the access to the channel is done using TDMA-DAMA in which three methods are employed for duplexing traffic of downlink and uplink: FDD, FSDD and TDD:

– **IEEE 802.16a:** after completing the IEEE 802.16 standard, the group started work on extending and modifying it to work in both licensed and license-exempt frequencies in the 2 GHz to 11 GHz range, which would enable NLOS deployments. This amendment, IEEE 802.16a, was completed in 2003, with OFDM schemes added as part of the physical layer for supporting deployment in multipath environments. Besides the OFDM physical layers, 802.16a also specified additional MAC-layer options, including support for OFDMA.

– IEEE 802.16d: this is a revised standard that replaces 802.16, 802.16a and 802.16c with a single standard. Note that this standard offers a variety of fundamentally different design options. For example, there are multiple PHY choices: a single-carrier-based PHY called Wireless-MAN-SCa, an OFDM-based physical layer called Wireless MAN-OFDM, and an OFDMA-based PHY called Wireless-OFDMA. Similarly, there are multiple choices for MAC architecture, duplexing, frequency band of operation, etc. This standard was developed to suit a variety of applications and deployment scenarios, and hence offer a plethora of design choices for system developers. In fact, it could be said that IEEE 802.16 is a collection of standards, not one single interoperable standard. The primary frequency bands suggested by this standard are as follows:

1) The 10-66 GHz band provides a physical environment where, due to the short wavelength, line-of-sight (LOS) is required and multipath is negligible. The channel bandwidths of 25 MHz or 28 MHz are typical with a raw data rate in excess of 120 Mb/s, which is suited to PMP access mode.

2) A frequency below 11 GHz provides a physical environment where, due to the longer wavelength, LOS is not necessary: this environment is well suited to the mesh access mode.

3) License-exempt frequencies below 11 GHz (5-6 GHz) are similar to those of the licensed band described in 2). However, it introduces mechanisms such as dynamic frequency selection to detect and avoid interference.

As a summary, the following table shows the different interfaces introduced by this standard.

Designation	Applicability	Duplexing
WirelessMAN SC	10-66 GHz	TDD FDD
WirelessMAN-SCa	< 11 GHz	TDD FDD
WirelessMAN-OFDM	< 11 GHz	TDD FDD
WirelessMAN-OFDMA	< 11 GHz	TDD FDD
Wireless HUMAN	< 11 GHz (5-6 GHz)	TDD

Table 1.2. *Air interface in IEEE 802.16*

– **IEEE 802.16e :** this is an amendment of the IEEE 802.16-2004 standard that added mobility support. IEEE 802.16e forms the basis for the WiMAX solution for nomadic and mobile applications and is often referred to as mobile WiMAX. It is expected that the mobile WiMAX will not only compete with the broadband wireless market share in urban areas with DSL, cable and optical fibers, but also threaten the hotspot-based Wi-Fi and even the voice-oriented cellular wireless market.

New features are introduced to this standard:

(i) a new scheduling service that builds on the efficiencies of UGS and rtPS. This is called extended real-time polling service (ertPS). In this case, periodic uplink allocations provided for a particular MS can be used either for data transmission or for requesting additional bandwidth. This feature allows ertPS to accommodate data services whose bandwidth requirements change with time;

(ii) three types of handover are introduced: hard handoff, fast BS switching (FBSS) and macro-diversity HO;

(iii) finally, a scalable OFDMA-based physical layer is introduced. In this case, the FFT sizes can vary from 128 bits to 2,048 bits.

1.3.5. *WMAN mobile: IEEE 802.20*

IEEE 802.20 or MBWA enables worldwide deployment of affordable, ubiquitous, always-on and interoperable multi-vendor MBWA networks that meet the needs of business and residential end-user markets. It specifies physical and

MAC layers of an air interface for interoperable MBWA systems, operating in licensed bands below 3.5 GHz, optimized for IP-data transport, with peak data rates per user in excess of 1 Mbps. It supports various vehicular mobility classes up to 250 km/h in a MAN environment and targets spectral efficiencies, sustained user data rates and numbers of active users that are all significantly higher than those achieved by existing mobile systems.

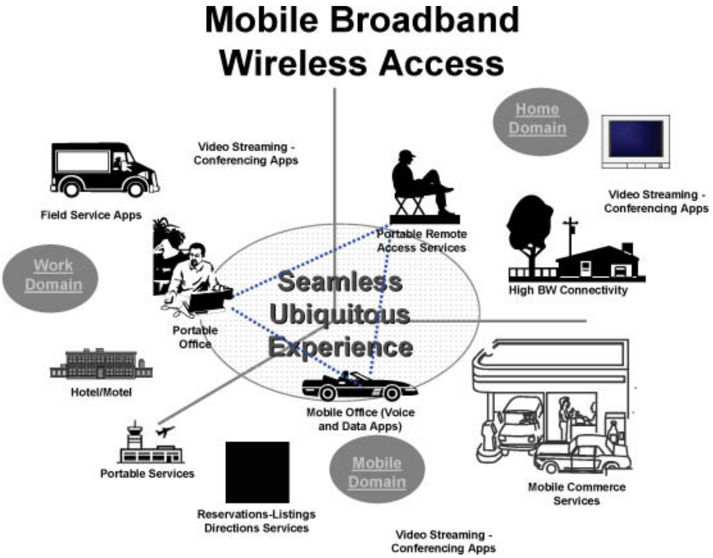

Mobile Broadband
Wireless Access
Home Domain
Video Streaming - Conferencing Apps
Field Service Apps
Video Streaming - Conferencing Apps
Portable Remote Access Services
Work Domain
Seamless Ubiquitous Experience
High BW Connectivity
Portable Office
Hotel/Motel
Mobile Office (Voice and Data Apps)
Portable Services
Mobile Domain
Mobile Commerce Services
Reservations-Listings Directions Services
Video Streaming - Conferencing Apps

Figure 1.8. *The vision of a seamless integration of the three user domains: work, home and mobile, with various scenarios. The IEEE 802.20 standard should form the basis for systems that support this vision*

The IEEE 802.20-based air interface will be optimized for high-speed IP-based wireless data services. It will support compliant mobile terminal (MT) devices for mobile users and will enable improved performance relative to other systems targeted for wide-area mobile operation. The air interface shall be designed to provide best-in-class performance attributes such as peak and sustained data rates and corresponding spectral efficiencies, capacity, latency, overall network complexity and QoS management. Applications that require the user device to assume the role of a server, in a server-client model, may be supported as well:

– **Applications:** the air interface will support applications that conform to open standards and protocols. This allows applications including, but not limited to, video, full graphical Web browsing, e-mail, file uploading and downloading without size limitations (e.g., FTP), streaming video and streaming audio, IP Multicast, Telematics, location-based services, VPN connections, VoIP, instant messaging and online multiplayer gaming.

– **Always on:** the air interface shall provide the mobile user with an “always-on” experience similar to that available in wireline access systems such as Cable and DSL while also taking into account and providing features needed to reserve battery life. The connectivity from the wireless MT device to the BS will be automatic and transparent to the user.

1.3.6. *MIH: IEEE 802.21*

IEEE 802.21 is an emerging IEEE standard. The standard supports algorithms enabling seamless handover between networks of the same type as well as handover between different network types, also called media independent handover (MIH) or vertical handover. The standard provides information to allow handing over to and from cellular, GSM, GPRS, Wi-Fi, Bluetooth, 802.11 and 802.16 networks through different handover mechanisms. This is done by introducing a new layer specified by MIH which provides three main functionalities: Media Independent Event Service (MIES), Media Independent Command Service (MICS) and Media Independent Information Service (MIIS).

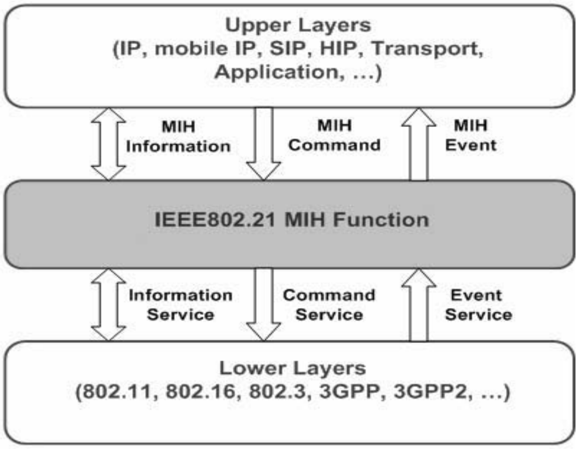


Figure 1.9. *IEEE 802.21 framework*

Thus, the heart of the IEEE 802.21 framework is the Media Independent Handover Functions (MIHFs), which provide abstracted services to higher layers by means of a unified interface. This unified interface exposes service primitives that are independent of the access technology and are called Service Access Points (SAPs). Figure 1.9 illustrates an example showing how the MIHF communicates with access-specific lower layer MAC and PHY components, including 802.16, 802.11 and cellular networks, using lower layer interfaces, and with upper layer entities. The services provided by MIHF are described as follows:

1) Media Independent Event Service (MIES): the event service is used to facilitate handover detection. Events inform the condition of the present network and transmission behavior of the data links, radio resource management, etc. The defined events include Pre-trigger (L2 Handover Imminent), Link Available, Link Up, Link Parameter Change, Link Going Up, Link Down, Link Going Down, etc.

2) Media Independent Command Service (MICS): higher layers use the MICS primitives to control the functions of the lower layers. MICS is used to gather information about the status of connected links, as well as to execute higher layer mobility and connectivity decisions on the lower layers. The MIH command can be both local and remote. These include commands from the upper layers to MIH and from MIH to the lower layers.

3) Media Independent Information Service (MIIS): as a mobile node is about to move out of its current network, it needs to discover the available neighboring networks and communicate with the elements within these networks so as to optimize the handover. MIIS provides a framework and corresponding mechanisms by which an MIHF entity can discover and obtain network information within a geographical area. MIIS primarily provides a set of information elements, the information structure and its representation as well as a query/response mechanism. The information service provides access to static information as well as dynamic information.

The MIH can be considered as a middleware at the link layer level in the components of a network that controls the mobility and the mobile device. Such middleware provides functions to a higher layer level. MIH extends the MIP since it can treat the information of layer 2, especially information concerning the status of the network and the use of the multiple technological interfaces. In order to do, the standard uses the triggers in multiple layers and proposes a 2.5 sublayer with an optimization of handover for the mobility in heterogenous interconnected networks.

IEEE 802.21 also provides the intelligence in the link layer and all information concerning the network to the higher layers in order to optimize the handover between the heterogenous mediums. The standard can support the handover for stationary and mobile users. For the mobile user, the handover can occur when the wireless condition channel changes, while handover for stationary users occurs when users prefer to use another type of network which has, for example, less load, best QoS, etc.

IEEE 802.21 supports a cooperative usage of mobile users and also the network infrastructure. The mobile user is capable of detecting the available network, and the infrastructure may store information demanded by then network, just like the list of cells in the neighborhood and the localization of mobile equipment. In general, the equipment of the client and the points of attachment (access point in Wi-Fi or BS in WiMAX) may support multimodal interface and in some cases it can support more than one interface simultaneously.

The standardization of IEEE 802.21 is in progress and it supports a framework for seamless continuity of service for mobile users. The function of MIH introduced by the standard enables a good decision of handover. The higher layers take the decision of handover depending on the entry and the context of the MIH function. The principal components of the IEEE 802.21 framework help an efficient and optimized handover.

1.3.7. *WRAN: IEEE 802.22*

In October 2004, IEEE set up a working group to develop the 802.22 standard for WRANs. The standard specifies a cognitive air interface for fixed, point-to-multipoint, wireless regional area networks that operate on unused channels in the VHF/UHF TV bands between 54 and 862 MHz. This is an ideal spectrum for deploying regional networks to provide broadband service in sparsely populated areas, where vacant channels are available.

The 802.22 system specifies a wireless air interface whereby a BS manages its own cell and all associated Consumer Premise Equipment (CPE). The BS controls the medium access in its cell and transmits in the downstream direction to the various CPE (which can be user-installable), which respond back to the BS in the upstream direction. In order to ensure the protection of incumbent services, the 802.22 system follows a strict master/slave relationship, wherein the BS performs the role of the master and the CPE is the slave. No CPE is allowed to transmit before receiving proper authorization from a BS, which also controls all the RF characteristics (e.g., modulation, coding and frequency of operation) used by the CPEs. In addition to the traditional role of a BS, which is to regulate data transmission in a cell, an 802.22 BS manages a unique feature of distributed sensing. This is needed to ensure proper incumbent protection and is managed by the BS, which instructs the various CPEs to perform distributed measurement activities.

Thus, as a conclusion, this standard is to cover the rural and faraway region which has a small density of population and to provide services which have the same efficiency as those provided by other broadband technologies such as xDSL. The second aim of this standard is to introduce the scalability in order to serve regions with a larger density of population where the spectrum is available. The typical range of this system is of 33 km for a region with 1.25 person density per /km^2, while the maximum range can reach 100 km.

1.4. Mobile Internet networks

1.4.1. *Introduction*

IP routing was designed without support for mobile nodes and was defined for fixed nodes. IP mobility has been made possible thanks to developments in wireless networks as well as developments in the miniaturization of portable and mobile terminals. IP mobility introduces new features in the network to ensure continuity of routing for mobile nodes on the move. These features are addressing, location management, re-routing and handover of the mobile's node:

– *Addressing*: in an IP network, support for mobile nodes requires two IP addresses: a fixed address of the mobile node, which is related to the home network that serves as an identification of the mobile node, and a temporary address that is related to the visited network. The temporary address changes as the mobile node moves from one temporary network to another. The temporary address is produced each time by a visited network.

– *Location management*: a correspondence is maintained in the network between the fixed address and the temporary address of the mobile node. This correspondence is conducted by a new entity in the network, a mobility agent. The mobile node must securely send its new temporary address for the mobility agent to maintain the correspondence between the temporary address and the permanent address of the mobile node and thus can locate it in order to forward its traffic to its current location.

– *Re-routing*: when the mobile node has an active session during its trip, it is the responsibility of the network to route the traffic to its new destination without interrupting the session.

– *Handover*: the handover is the process of changing the point of attachment to the network. It contains the *discovery phase* of the new visited network and *attachment* to this new network. The handover is difficult when there is an ongoing session because the whole issue is to change the point of attachment without interrupting the session.

From a performance protocol developed by the Internet Engineering Task Force (IETF) to support IP mobility (Mobile IP) creates too much latency in micro mobility. In order to do this, supporting IP mobility has been divided into two categories: support for macro mobility and support for micro mobility.

Macro mobility happens when a mobile node moves between two different areas. Macro mobility can take place during an active session of a mobile user, or during a new session initiated by the user from a visited network in a new domain which is known as a nomadic or roaming user. Micro mobility concerns a mobile node moving between two points of attachment belonging to the same area. Active research on this subject has raised several propositions; however, all lack efficient standards. The basic features of a mobility management are mainly management of the location and management of handovers. These functions are fully or partly necessary in a mobility management protocol [CHA 04]:

– Authentication and authorization.

– Packet transfer.

– Path update.

– Handover management.

– Support for inactive mobile.

– Address management.

– Support for security.

The selection criteria of the mobility management protocol are mainly the total duration of the handover and the rate of lost packets during the handover procedure. Thus, we speak of “smooth handover” to refer to a handover with minimal packet loss, “fast handover” with minimum delay, and “seamless handover” with a minimum delay and packet loss. The major problem of managing a handover in IP networks is that the layer 3 handover (network layer) only happens at the end of the layer 2 handover (link layer), which is due to the non-communicating layer principle in the TCP/IP model. This means that the mobile node has disconnected from its old access point and connects to a new access point, but only at the physical layer, and must wait for the layer 3 handover to ensure connectivity to the network, thus using network resources. The goal is to minimize the time between the executions of layer 2 and layer 3 handovers. To do so, we must improve the detection techniques of movement at layer 3 and try to synchronize early layer 2 and layer 3 handovers.

In the case of macro mobility management, the Mobile IP protocol is accepted by the community of the IETF as the management standard of the macro mobility.

Mobile IP allows continuity of service for moving terminals; this is a challenging issue for IP networks.

The classification of micro mobility protocols are mainly two broad categories: proxy-agent architecture (PAA) and localized enhanced-routing schemes (LERS) [MAN 02]. Architecture-based agents are a proxy proposal to extend Mobile IP and use a hierarchy of mobility agents which are variations of Mobile IP foreign agents (FA). Architectures based on the modified localized routing protocol uses a dynamic routing especially in a localized area.

Figure 1.10 below summarizes the various proposals supporting mobility in IP networks. This section will introduce one or two protocols of each category.

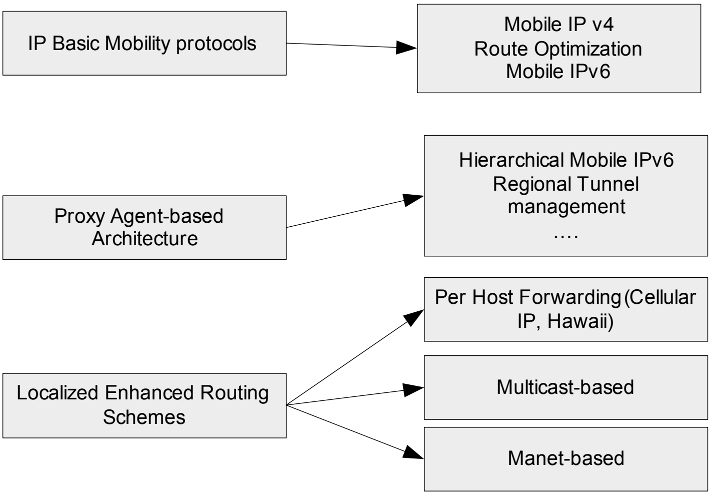

IP Basic Mobility protocols
Mobile IP v4
Route Optimization
Mobile IPv6
Proxy Agent-based Architecture
Hierarchical Mobile IPv6
Regional Tunnel management
....
Localized Enhanced Routing Schemes
Per Host Forwarding (Cellular IP, Hawaii)
Multicast-based
Manet-based

Figure 1.10. *Classification of IP mobility support*

1.4.2. *Macro mobility*

Proposed by the IETF, Mobile IP is the standard to support macro mobility in IP networks [PER 97], [PER 02]. It aims to make the mobility transparent to the upper layers of the TCP/IP model as well as, in terms of IP routing, the disconnection due to the change of point of attachment. To do so, Mobile IP proposed a mechanism for managing location and a mechanism for re-routing or transfer packets to the mobile node. Mobile IP defined a mobile home network where there is a home agent (HA) and visited network with a foreign agent (FA).

The HA is responsible for maintaining the location information updated regularly by the mobile node. It also transfers packets to the mobile node through an IP tunnel (IP-in-IP) via the FA in case the mobile node has a non-routable IP address (Care of Address (CoA) allocated by the FA). Otherwise the IP tunnel might be established between the HA and the mobile node directly if it has a routable IP address (co-located CoA-CCOA that can be allocated by the Dynamic Host Configuration Protocol (DHCP)). When the mobile node cannot obtain a new address from the HA or DHCP, the FA will provide a new IP address (temporary IP address) in the advertisement message field and also communicates with the HA to transfer the request to register the current location (temporary IP address) sent by the mobile node. The FA will also retrieve the packets sent to the mobile node in the IP tunnel established with the HA and transfer them to the mobile node.

The basic operation of Mobile IP is the same for Mobile IPv6 and IPv4. However, Mobile IPv6 provides new features of IPv6 solutions to some problems in Mobile IPv4 such as the known triangular routing where the correspondent node should send packets to the HA that transfers them to the mobile node using the correspondence between the permanent address and the temporary address.

The mobile node, after completing its temporary address (CoA or CCOA), will send a request to the HA to record its new location. The HA will keep a correspondence between the permanent address of the mobile node that is linked to its HA and CoA. The correspondent nodes (CN) will send the traffic using the permanent address of the mobile node. The HA will intercept these packets and will be able to redirect it to the mobile node by building an IP tunnel [PER 96]. The encapsulated packets will be redirected and an additional IP header will be added which contains the destination address, the CoA address of the mobile node. The IP tunnel is established with the FA if the mobile node has a CoA or directly with the MN if it has a routable CCOA address. In Mobile IPv4, we call this redirection triangular routing. An improvement of this proposed routing was called route optimization (RO) [PER 01], where the mobile node informs the HA and directly the correspondent node of its new location so that packets will be sent directly from correspondent node to the mobile node. This has unfortunately experienced security problems that have been resolved in Mobile IPv6. Thus, in Mobile IPv6, there is no triangular routing and the FA is not necessary since the mobile node can always build its own address to be routable through the IPv6 addressing plan.

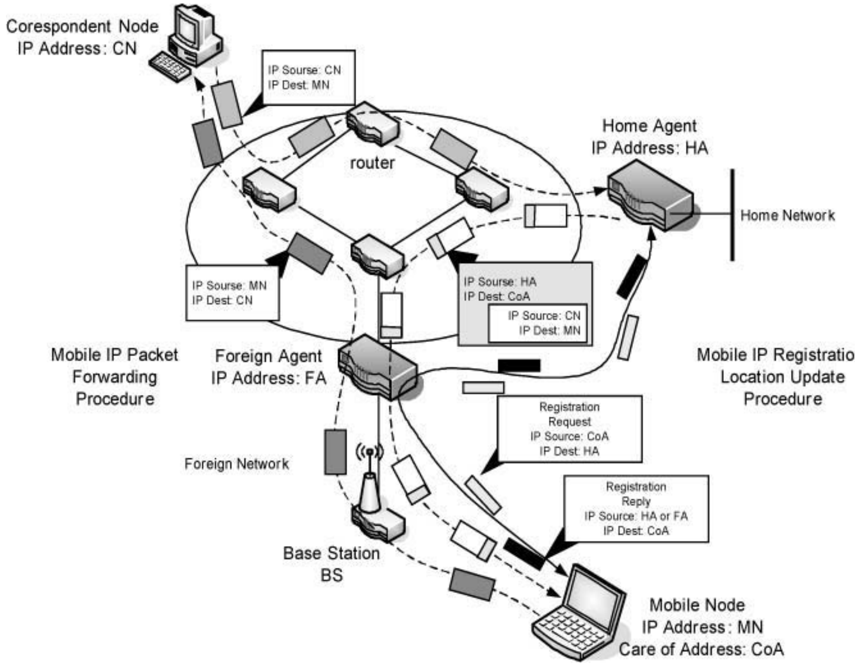


Figure 1.11. *Mobile IP [CHA 04]*

1.4.3. *Micro mobility*

The performance of Mobile IP in the case of micro mobility where the mobile node moves between two BSs in the same area showed Mobile IP's inability to support this type of mobility [CAM 02], [REI 03]. In order to do so, a large amount of work has produced other approaches such as hierarchical approaches (PAA), or the enhanced routing approach (LERS) [MAN 02]. In the first category, there are hierarchical Mobile IP (HMIP) [CAS 00], [MAL 00b], Fast Handoff (FMIP) [MAL 00a and c] and other improvements for Mobile IP [MAL 01]. Other micromobility working groups are active at the IETF, such as Netlmm [NETLMM] where Proxy Mobile IP and Netlmm approaches are ongoing and Mext [MEXT] where IP mobility extensions are proposed. In the second category, there are Cellular IP [CAM 00b], HAWAII [RAM 00] and other approaches based on the multicast and ad hoc routing (MANET) approach [MAN 02].

a) Proxy-based architecture

This type of architecture introduces the concept of hierarchy of mobility agents (FA and/or HA) to localize the update messages and minimize the time needed to complete the handover process.

One of the proposed changes of Mobile IP is hierarchical Mobile IP [CAS 00] which tries to improve the performance of Mobile IP in micro mobility. An FA will be installed at the gateway of the visited network forming a GFA (Gateway Foreign Agent), which it will be responsible for the regional registration procedure [GUS 01], [MAL 00 2], thus hiding from the HA all movements of the mobile node inside the same visited network covered by a GFA. The mobile node will, in addition to the permanent address (HA), have a temporary address CoA to be attached to the gateway GFA, and a local address CoA to be attached to the FA visited network. Thus, the HA retains the correspondence between the permanent address (HA) and the gateway CoA (GFA); however, the GFA keeps correspondence between the local CoA and the gateway CoA (GFA). The registration procedure is identical to that of Mobile IP, the only difference being that registration with the HA is only necessary if the mobile node changes its GFA, otherwise the registration within the visited network is done with the GFA, which plays the role of a local HA. Packets to the mobile node are redirected by the HA to the GFA, which then transfers them to the mobile node.

Another improvement of Mobile IP is Fast Handoff [MAL 00a and c] which attempted to improve further handover delay of HMIP in micro mobility. To do so, it proposes to improve the detection of movement of the mobile node by using information from the link layer on top of the layer 2 handover (link layer). This information is used to precisely predict the layer 3 handover and thus can use network resources in the new cell as soon as possible after completing the layer 2 handover. In order to do this, the mobile node will begin its registration procedure (layer 3 handover) with the new FA through the old FA before the layer 2 handover is completed. In addition, a tunnel will be established between the old FA and the new FA to deliver packets that continued to arrive at the former FA during the handover. This is possible through the mechanism of "route optimization". Indeed, the mobile node sends its new location at the same time to the HA and to all the correspondent nodes, so that they send packets directly to the mobile node. Other proposals to improve the quality of the handover in Mobile IP have been proposed, such as proactive handover or Telemip [REI 03]. Other micromobility working groups are active at the IETF such as Netlmm [NETLMM], where proxy Mobile IP and Netlmm approaches are proposed, and Mext [MEXT], where IP mobility extensions are proposed.

b) Enhanced localized routing-based architecture

To manage micro mobility, this approach introduces a dynamic routing in certain parts of the network. Three categories of solutions are identified [MAN 02]. Per host forwarding architectures such as Cellular IP or HAWAII are based on Multicast and other architectures based on MANET.

Cellular IP [CAM 00b] was designed to replace IP in an access network. A Cellular IP area is composed of a MA (mobility agent) which is a gateway to the Internet and acts as a FA and running Mobile IP. Each MA contains a cache that contains the routing node next to the mobile node and an index to reach the gateway. This cache is used by the MA to transfer packets from the gateway to the mobile node or the mobile node to the gateway. Routes are established and maintained through the transmission of two control packets. A "beacon" message is sent periodically by a gateway to precisely create the routes to the gateway at all MAs. The packet route update message is sent by the mobile node at its first network connection when it changes its point of attachment, and also periodically. These packets are transferred hop by hop to the gateway to creating or updating and entries in the routing cache of each MA. Cellular IP offers two types of handover: hard handover where a packet route update message to the gateway is sent after layer 2 handoff, and semi-soft handoff where the mobile node uses the information of the link layer in advance of the layer 2 handoff. In this case, the mobile node requests to begin the bicasting process to the old and the new cell in order to minimize the loss of packets. Cellular IP offers connectivity support for passive nodes (paging) using "paging cache".

Other proposals based on dynamic routing similar to Cellular IP have been introduced as HAWAII. On the other hand, Multicast architectures that were designed to withstand the point-to-multipoint connections regardless of location, addressing and routing have also been used to support micro mobility. In the proposed architecture based on Multicast for mobility, the mobile node will obtain a multicast-CoA address. The mobile node can ask the neighbor access routers to join his Multicast group either before or during the handover. "Dense Mode Multicast and Sparse Mode Multicast" are examples of Multicast proposals to support micro mobility [MAN 02], [MYS 97], [MIH 00].

Finally, there is MANET for the management of mobility that has also been proposed as MER-TORA [MAN 02]. MANET protocols have been designed to support ad hoc networks where the mobile node and routers are mobile. In the case of IP mobility management, we consider that the access network is part of the ad hoc network which is fixed, and there are terminals that are mobile. Thus, the proposal by the ad hoc approach is applicable in this case too where only mobile nodes are the moving parts.

1.4.4. *Personal mobility and SIP*

The IETF has developed a signaling protocol SIP (session initiation protocol) [RFC 3261], which can also be used to support the so-called personal mobility. Personal mobility allows a user to change terminal and recover their session. Unlike Mobile IP, SIP acts at the transport layer and not at the network layer of the TCP/IP model. SIP is independent of the transport protocol (UDP, TCP, ATM, etc.). It uses a logical address instead of IP addresses. It controls a multimedia session with two or more participants. It is a lightweight protocol based on the text and is not complex with little load in the network. In November 2000 SIP was accepted by the 3GPP as the signaling protocol and permanent element of the next generation network IMS (IP Multimedia Subsystem). SIP terminals are already on the market for applications such as VoIP. Several conversation clients via the Internet also use SIP (Windows Messenger, AOL Instant Messenger, etc.).

SIP mainly proposes adding a "user agent" in the terminal user who plays the role of SIP client, a registrar or registration server; it keeps the location information provided by the "user agent" and a proxy between two "user agents" that can relay SIP requests and ask the right "registrar" to locate the corresponding "user agent". These components are separated logically and not necessarily physically. SIP can operate in peer-to-peer mode, but in the context of deployment of public services, registration servers and proxies are necessary.

1.4.5. *Identity based mobility*

In today's Internet architecture, IP addresses are used both as locators and identifiers. This dual role of IP addresses has several problems. Firstly, IPv4 is still more widely used than IPv6, so address space of IPv4 becomes insufficient due to increasing Internet usage and the number of hosts. Furthermore, as the mobility of devices increase, the dual role of IP addresses makes mobility management complicated.

In order to solve these problems, the Host Identity Protocol (HIP) was proposed by the IETF and IRTF (Internet Research Task Force). It simply proposes to separate the locators and identifiers. In HIP, IP addresses act only as locators while host identities identify themselves. This situation requires adding a new layer in the TCP/IP stack between the transport layer and the IP layer. The role of this layer is to compensate host identities with upper layer protocols.

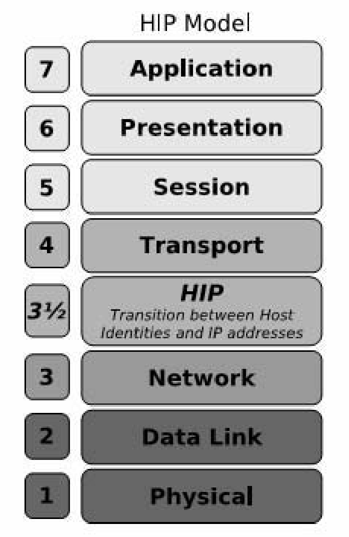


Figure 1.12. *Host Identity Protocol in ISO layers [HIP]*

One of the issues completely defined in HIP is that the Host Identity (HI) is the public key from a public/private key pair. This key can be represented by the Host Identity Tag (HIT), a 128-bit hash of the HI, and has to be globally unique in the whole Internet universe. Another representation of the HI is the Local Scope Identity (LSI) which is 32 bits in size and can only be used for local purposes.

The HIP Base Exchange is a cryptographic key-exchange procedure performed at the beginning of the HIP communication establishment. It is built around a classic authenticated Diffie-Hellman key exchange. The BE is a four-way packet exchange between the Initiator (I) and the Responder (R). The initial IP address of a HIP host should be stored in order to make the host reachable. Traditionally, the DNS is used for storing this information. The problem with the DNS system is the latency: when updating the location information each time the MN moves, the update is not fast enough. The Rendezvous Mechanism is designed to solve this problem. The Rendezvous Server (RVS) keeps all the information of HIP communication. The location information of RVS is just stored in DNS. If a MN wants to communicate with other MNs, all nodes have to register with their RVS.

HIP enabled R to register to the RVS with its HIT and current IP address. When I wants to establish a connection with R, it first sends the I1 packet to one of R’s RVSs or to one of IP addresses (if it can be learnt via DNS). I gets the IP address of R’s RVS from DNS and send the I1 packet to the RVS for Base Exchange. RVS checks whether it has the HIT of the I1 packet. If HIT belongs to itself, it sends the I1 packet to a related IP address. R sends the R1 packet directly to I without RVS. For more details, the reader is invited to visit the HIP working group website [HIP] and [AYD 08].

1.4.6. *NEMO and MANET networks*

a) MANET

MANET brings the concept of dynamic IP routing. Unlike simple IP routing where all nodes are fixed, ad hoc routing applies to a network where all nodes can be mobile or can join or leave the network dynamically. There is no concept of infrastructure in an ad hoc network. The network is self-organized but provides connectivity nodes through ad hoc routing. The IETF has defined two main categories of ad hoc routing proactive and reactive routing according to network density and mobility of nodes.

Proactive routing distributes routing information regularly to update the knowledge of nodes on the network structure and allows the routing of traffic to the correct destination. Reactive routing searches for the right route before sending traffic to the destination. It does not maintain information routing throughout the network in the way proactive routing does. The proactive routing does not scale well in high mobility networks; it is good only in low mobility and/or low density networks. Other approaches to ad hoc routing are related to hybrid routing, which combines a proactive and reactive or geographic routing. Multicast routing is also possible but not mature enough compared to the reactive and proactive approaches.

Support for mobility is inherent in this type of network since routing adapts to the new network setup (arrival or departure of nodes, mobility of nodes, etc.). Some MANET network applications are today possible such as VANETs (vehicular ad hoc networks) used for communications between vehicles or between vehicles and equipment on the road. MANETs or VANETs are unfortunately not yet deployed to support billable services since the network still needs to mature on several fronts such as security, but above all it is necessary to define a profitable business model for this type of technology.

b) NEMO

The concept of NEMO (Network Mobility) is introduced to support groups of nodes that form an entity in relation to mobile network infrastructure, for example, a set of nodes in a train could form a NEMO network. It supports the mobility of one or more nodes that are in charge of the NEMO network connectivity to the network infrastructure. Such networks also present new application opportunities of IP mobility. They also open new opportunities for services to service providers. That being said, there are not yet applications in the market based on these technologies. This is certainly due to the fact that they are not yet ready in terms of QoS and security.

1.5. Current trends

1.5.1. *All-IP, IMS and FMC*

IMS (IP Multimedia Subsystem) is the result of the efforts of 3GPP and 3GPP2 to define a wireless network All-IP unlike the existing different networks such as conventional voice, data and control networks. It was initially defined by the industrial forum 3GPP in 1999 before being brought in 2000 to a draft in 3GPP release 5 where SIP was adopted for signaling.

In 3GPP release 6, interworking with WLAN has been added. Then in release 7, support for fixed networks was also added with TISPAN R1. This is known as FMC (Fixed Mobile Convergence). The IMS network is the core that can be used by different access networks (GPRS, UMTS, CDMA2000, WLAN, WiMAX, DSL, cable, etc.). IMS being in the beginning designed for mobile networks, FMC also became a priority in 2005. This is especially interesting for fixed operators to enable them to continue to exist in the mobile market. It will mainly support packet switching communications; however, gateways to support circuit switching systems are used. Open interfaces between control plans and services are developed to help new services to be built in a more optimal way. The mobility of the terminal is provided by the network access while personal or user mobility will be provided by the IMS network through SIP. IP-based services will be easily deployed on IMS since it is all an IP network, for example, VoIP services, push to talk over cellular (POC), network games, videoconferencing, messaging, community services, presence information and content sharing.

From a user point of view, the interest of IMS is for instance to enable them to have only a single phone number on their phones, both fixed and mobile. The transition from one access network to another of a different technology will be transparent to the user. The operator can also provide various services to the user regardless of its location, its access technology or its terminal. It could also be suggested that the development of this system is only a means to enable fixed operators to go on the mobile market and vice versa because their telephone related revenues are lower because of the VoIP. Moreover, from a realization point of view, the cost of such integration will certainly not be negligible, so we can then ask why not just use the Internet to enable this convergence and this

All-IP? This certainly avoids all the problems of integration between the 3GPP and the IETF for the SIP world. However, the Internet network is known to be less controllable than telecommunication-based networks.

1.5.2. *B3G and 4G*

4G technology refers to the future standard networks and wireless terminals. Initially, 4G meant the integration of heterogenous wireless networks building a wireless network supporting heterogenous mobility or vertical handovers. This combined the advantages of each wireless technology (bandwidth, security, cost, etc.). The ITU has specified only the expected bandwidth for 4G networks, so the field is open to competition, in particular between the standards developed by the cellular networks and IEEE wireless networks. The ITU has nevertheless stated that a 4G network is a network that can provide a flow rate of 100 Mbps uplink and 1 Gbps downlink on the move. The 4G network is also known as the B3G (Beyond 3G). NTT DoCoMo and Samsung have tested their prototype 4G appointed VSF-OFCDM, which provided 100 Mbps and 1 Gbps downlink on the move. NTT DoCoMo intends to market its first services in 2010 in Japan and Samsung are also marketed in 2010 in Jeju-do and South Korea. This also means an evolution of 3G terminals.

Initially, infrastructure and terminals will all support current technologies and will be generic enough to accommodate and easily evolve to new technologies such as 4G. The candidate technologies to implement the 4G are WiMAX, WiBro and 3GPP Long Term Evolution (HSOPA). OFDM is the multiplexing technology in 4G combined with other access methods such as dynamic TDMA or OFDMA. Technologies such as the SDR (Software Defined Radio) for the reconfiguration of terminals and network service function or state of the network will be used. Moreover, unlike 3G, where there are still circuit switching functionalities, 4G will only support packet switching.

1.5.3. *Applications*

Technology can die as a result of lack of application or usage even if it is efficient technology. Wireless and mobile networks have largely evolved in order to provide QoS comparable to fixed networks but also provide other services that are necessary in response to user mobility. The increased flow of wireless and mobile networks made it possible today to carry voice and data but also video and even real-time video (teleconferencing). Audiovisual services (TV, TV on demand, etc.) on mobiles seem to be a way to generate revenues from these new broadband wireless and mobile networks. Mobile networks have clearly evolved from simple telephone applications to a variety of services. Unlike a simple call that has a beginning and an end, the concept of session represents a continuum of communication in which all kinds of events can happen, successively or combined in terms of messages (voice, images, video, text, etc.) or number of correspondents entering or leaving the session. And all this regardless of access mode: the integrated operators should not only operate the network convergence, but also usage on all terminals: PC, mobile, fixed terminals. Moreover, applications developed in open system will better serve customers. It will not break them down according to their numbers, but will unite and enrich their user profile in a dynamic directory to better direct calls or sessions, incorporating, for example, notions such as presence. For example, if a user wants to be contacted after 8 pm, they could do so depending on the availability that would have defined, in their MMS on mobile instant messaging in their PC, voice mail or mail on their smartphone, on their fixed line or personal line of their second professional line, etc. Applications being limited, the management of the presence opens up new prospects for billable services to clients.

The convergence of services, such as voice, data and images, and networks, such as fixed and mobile networks, always aimed to optimize network resources, maximize income and minimize effort for users. We again encounter an old debate, which is to put intelligence in the network or the terminal. The IMS network has clearly chosen the first approach, which is to introduce intelligence in the network. Issues such as quality applications for mobile users depend not only on network resources but also on the capacity of mobile terminals, which would be decisive for the success of these new wireless networks and mobile broadband. The problem will be to provide attractive services from a financial point of view to the user. In addition to technical challenges to be solved, it clearly depends on the cost of achieving these new technologies, and the political strategy of countries on the role of new technologies in the evolution and equilibrium of societies.

1.6. Conclusions

Different technologies have been developed by different organizations that come together in two main families: the family of IEEE wireless networks (Wi-Fi, WiMAX, etc.) and the family of cellular mobile networks (1G, 2G, 3G, etc.). Current trends are developing an all-IP core network to provide mobile services to users regardless of their location or terminal. On the other hand, the wireless network evolves to provide more bandwidth; this is the 4G network, which will probably be a combination of the different technologies above or simply a new technology offering very high bandwidth comparable to fixed networks. The goal remains to maximize the radio resource, support mobility, providing multimedia services (voice, data, image), transparency to the user and ensuring the security of communications.

1.7. Bibliography

[802.20] www.IEEE 802.org/20/.

[802.21] www.IEEE 802.org/21/.

[AYD] GURKAS AYDIN, Z. and CHAOUCHI, H., "Micro Mobility with HIP", *Research Report,* 2008.

[CAM 00a] CAMPBELL, A.T. and GOMEZ, J., "IP Micromobility Protocols", *ACM Mobile Comp. and Commun. Review*, vol. 4, no. 4, October, 2000

[CAM 00b] A. T. CAMPBELL *et al.*, "Cellular IP," Internet draft, draft-ietfmobileip-cellularip-00.txt, January, 2000, IETF draft document.

[CAM 02] CAMPBELL, A.T. *et al.*, "Comparison of IP Micromobility Protocols," *IEEE Wireless Commun.*, vol. 9, no. 1, February, 2002.

[CAS 00] CASTELLUCCIA, C. and BELLIER, L., "Hierarchical Mobile IPv6", Internet Draft (IETF draft document), July 2000 (draft-castelluccia-mobileip-hmipv6-00.txt).

[CHA 04] CHAOUCHI, H., "On Global Mobility Control in IP networks", PhD, Paris VI University, 2004.

[GUS 01] GUSTAFSSON E., JONSSON A. and PERKINS C., "Mobile IP Regional Registration," Internet draft, draft-ietf-mobileip-reg-tunnel-04.txt, March, 2001, IETF draft document.

[HIP] http://www.ietf.org/html.charters/hip-charter.html.

[MEXT] http://www.ietf.org/html.charters/mext-charter.html.

[NETLMM] http://www.ietf.org/html.charters/netlmm-charter.html.

[MAL 00a] EL MALKI, K. and SOLIMAN, H., "Fast Handoffs in Mobile IPv4", Internet Draft (IETF draft document), September, 2000 (draft-elmalki-mobileip-fasthandoffs-03.txt).

[MAL 00b] MALINEN, J. and PERKINS, C., "Mobile IPv6 Regional Registrations", Internet Draft (IETF draft document), July, 2000 (draft-malinen-mobileip-regreg6-00.txt).

[MAL 00c] EL MALKI, K. and SOLIMAN, H., "Hierarchical Mobile IPv4/v6 and Fast Handoffs". Internet Draft (IETF draft document), March, 2000 (draft-elmalki-solimanhmipv4v6-00.txt).

[MAN 02] MANNER, J., LÓPEZ, A. and MIHAILOVIC, A. *et al.*, "Evaluation of Mobility and QoS Interaction", *Computer Networks*, vol. 38, no. 2, 137-163, February, 2002.

[MIH 00] MIHAILOVIC, A., SHABEER, M. and AGHVAMI, A.H., "Multicast for Mobility Protocol (MMP) for emerging Internet networks", Proceedings of PIMRC 2000, London, UK, September 2000.

[MYS 97] MYSORE, J. and BHARGHAVAN, V., "A New Multicasting-based Architecture for Internet Host Mobility", *Proceedings of ACM Mobicom*, September, 1997.

[OHR 05] OHRTMAN, F., *WiMAX Handbook*, McGraw-Hill Communications, 2005.

[ORE 05] O'REILLY, F., "IMS: une nouvelle architecture qui va révolutionner les télécoms?", http://www.itrmanager.com.

[PER 96] PERKINS, C., "IP encapsulation within IP," *Internet RFC, RFC 2003*, Oct. 1996.

[PER 97] PERKINS, C., "Mobile IP," *IEEE Commun. Mag.*, vol. 35, no. 5, pp. 84–99, May, 1997.

[PER 01] PERKINS, C. and JOHNSON, D., "Route optimization in Mobile IP," Internet draft, draft-ietf-mobileip-optim-11.txt, Sept. 2001, IETF draft document.

[PER 02] PERKINS, C. (Ed.), "IP mobility support for IPv4," *Internet RFC RFC 3220*, January, 2002.

[PUJ 06] PUJOLLE G., *Les réseaux*, Eyrolles, 2006.

[RAM 00] RAMJEE, R. *et al.*, "IP Micro-mobility Support using HAWAII", Internet draft, draft-ietf-mobileip-hawaii-01.txt, July 2000, IETF draft document.

[REI 03] REINBOLD, P. and BONAVENTURE, O, "IP micro-mobility protocols", *IEEE Communication, Surveys*, vol. 5, no. 1, 2003.

[STA 05] STALINGS, W., *Wireless Communications Networks*, 2nd edition, Pearson Education, 2005.

[TAN 99] TAN, C., PINK, S. and LYE, K., "A fast handoff scheme for wireless networks", *Proceedings of the Second ACM International Workshop on Wireless Mobile Multimedia*, ACM, August, 1999.

[URL] http://bertrand.fievet.free.fr/cdma_principes.html.

[WIKI] http://en.wikipedia.org/wiki.

[YAH 08] YAHIYA, T.A., "Cross Layer Design for ressource allocation in mobile wimax networks", PhD, Paris VI University, 2008.

Chapter 2

Vulnerabilities of Wired and Wireless Networks

2.1. Introduction

This chapter synthesizes the vulnerabilities common to modern telecommunications systems. The presented synthesis is general in the sense that it does not depend on particularities of any specific system like topology, form, used media, implementation, etc.

Section 2.2 introduces the definitions of security and trust in the digital age. Most notably, this section introduces the terminology commonly used in this context. The needs in terms of security mechanisms are illustrated using dependencies and relations between an asset, its owner and the environment.

Section 2.3 presents threat models for diverse modern telecommunications systems. This section discusses various aspects with impact on the security situation and architecture, such as heterogenity and homogenity of systems, used medium and the extraordinary role of the Internet. This section introduces a classification of typical vulnerabilities by distinguishing infrastructural and personal risks.

Finally, section 2.4 shows how the previously described typical vulnerabilities evolve when introducing wireless communications. This last section makes reference to the proposed taxonomy and insists on additional difficulties caused by wireless communications.

Chapter written by Artur HECKER.

2.2. Security in the digital age

2.2.1. *Private property: from vulnerabilities to risks*

> *Everyone has the right to the protection of the moral and material interests resulting from any scientific, literary or artistic production of which he is the author.*
>
> *Article 27.2, Universal Human Rights Declaration (1948)*

The principle of private property is one of the pillars of modern society. The protection of such property, in the sense of physical and virtual assets, patrimony, investments, as well as the respect of related areas such as privacy and human rights, is the moral and legal obligation of the State, businesses and citizens.

The cultural and industrial development of the late 19th and 20th centuries and ongoing globalization have changed the perception of assets and their values, pushing the latter progressively from a purely materialized form (real estate, basic products) to more abstract forms. This is underlined for example through the introduction of international laws on intellectual property (copyright laws [BERCN]), the evolution and rapid development of the service sector, etc. This process has reached its apogee with the beginning of the digital age, which, through the broad introduction of the digital patrimony, erases the last frontiers between real assets and virtual assets. Finally the notion of soft products appears (of which the main representative is software, but also multimedia, games, etc.), and, consequently, with the emergence of the Internet, entirely digital commerce. Indeed, today, we have commercial activities that are entirely based on selling digital through digital to digital (e.g. iTunes).

Yet, such digitized and therefore virtualized information is particularly sensitive and vulnerable to volatility, modifications and uncontrollable duplication. Indeed, digital products do not allow the notion of authenticity since every instance is, essentially, a clone.

In addition, globalization and development of telecommunications technologies and services results in the standardization and openness of information systems (ISs). The proprietary part in ISs is declining steadily, and we see trivialization of access, amplified interconnection of systems and a strong trend towards convergence of previously separate sectors, as with the multimedia, conventional telecommunications and computing industries, for example. The ease of access amplifies the exchanges of assets with its environment. Given the innate vulnerability of digital goods, the potential for abuse increases with such exposure to multiple uses, while protection becomes harder.

In the digital era, virtual assets (software products, know-how, algorithms, knowledge, information, multimedia, data, etc.) become integral parts of IS infrastructures designed to provide different services. Faced with such a distribution of property in the digital infrastructure of interconnected information systems, operators of such systems, users (businesses and individuals) and the State must face questions on the issue of protection of the exchanged and contained information. Such protection should cover:

– all assets, i.e. both the transported content and the parts used or stored in a given infrastructure;

– all types of actors;

– the entire lifetime, especially respecting the particular aspects of the actual use but also the legal issues (expiration and forced deletion vs. audit needs).

Today providing many critical services (air traffic control, defense, emergency services, trading, etc.) telecommunications systems have become indispensable for all actors in the information society. The relationships between these infrastructures and conventional classical infrastructures (energy, transport, water) create new systems whose complexity and vulnerabilities are greater than those of the constituting systems. New provisions become necessary to take into account the interdependences of modern critical infrastructures (strength, resilience, safety, etc.).

At the other end of the spectrum, the democratization of information technologies, driven by astonishing technological progress, creates proper personal IT infrastructures in the private space of individuals. In most cases, these systems are no longer isolated, but on the contrary become more and more open, often overlapping in several dimensions and as such difficult to delimit in practice. Today, various actors, most notably citizens and the State, must master both content and operations on these infrastructures (transport, processing and storage) and the infospheres per se, i.e. the virtual infrastructures created in part within the respective private space by the interconnections of the ISs and in part by the sharing of digital assets (see Figure 2.1).

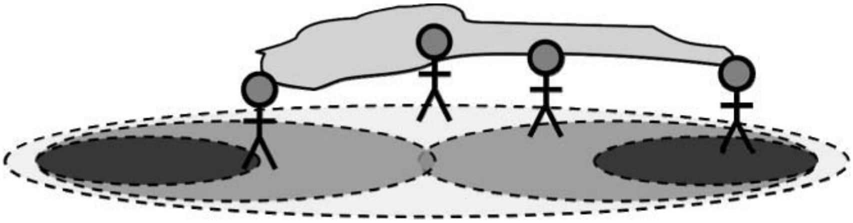


Figure 2.1. *The interconnection of information systems creates new virtual infrastructures*

2.2.2. *Definition of security*

It is necessary to find a definition of security common to an asset, a service, an infrastructure and an infosphere for any concerned owner. We can find several definitions of security:

> **Security**
>
> *noun (pl. securities) 1 the state of being or feeling secure. 2 the safety of a state or organization against criminal activity such as terrorism or espionage. 3 a thing deposited or pledged as a guarantee of the fulfillment of an undertaking or the repayment of a loan, to be forfeited in case of default. 4 a certificate attesting credit, the ownership of stocks or bonds, etc.*
>
> *(Compact Oxford English Dictionary)*

Several aspects discussed below are visible in these definitions. In definitions 1 and 2, security is seen as a situation characterized by the absence of any danger or risk to persons involved ("I feel safe"). Although this definition is at a sufficiently high level and is, in particular, applicable to ISs, it merely states the situation sought but does not mention means to achieve it. Definitions 3 and 4 describe specific mechanisms to achieve certain security understanding in the economic sector, namely through a deposit or through certification of liquidity. These two are too specific from the point of view of information systems.

According to another vision, security is often seen as the art of sharing secrets. Cryptologists often use this definition of security. It is at a very low level, generally necessary and true, but insufficient in many contexts today. It is rather difficult to apply to modern information systems in a "top-down" approach. Moreover, the definition depends on the implementation of possible security measures and is necessary in some but not all cases.

We define security in the digital age as a quest for the protection of digital assets and the protection of systems treating such assets against any act which is unwanted or perceived as abuse by the respective owners. Such unwanted acts are typically possible because of vulnerabilities present in the ISs. The exploitation of vulnerabilities creates threats and thus represents a risk from the point of view of the owner. Conversely, in the security methodology the perception of risks to assets by the owner leads to the implementation of a set of counter-measures within the IS.

Our definition is at the crossroads of the usual definition, aiming to install peace of mind, and of the military definition [DOD03], which insists on measures to be taken. Our definition takes as its starting point the existence of an asset, good or value that deserves to be protected in a processing environment. The term "quest", used in this definition, stresses the continuity of the process and the uncertainty, which are typical for security: counter-measures must evolve over time, and usually it is not known whether they are sufficient; counter-measures might have their own vulnerabilities – their presence and the reliance on their function leads to new threats against which the owners must protect themselves. Moreover, the notion of "unwanted" in the definition implies the presence of at least two separate players, called the owner and attacker respectively. The attacker is typically presumed to be malicious and creates threats exploiting vulnerabilities in or around the asset. The owner wants to minimize risks and imposes counter-measures that he considers necessary to protect the asset (see Figure 2.2). He therefore describes the security objective.

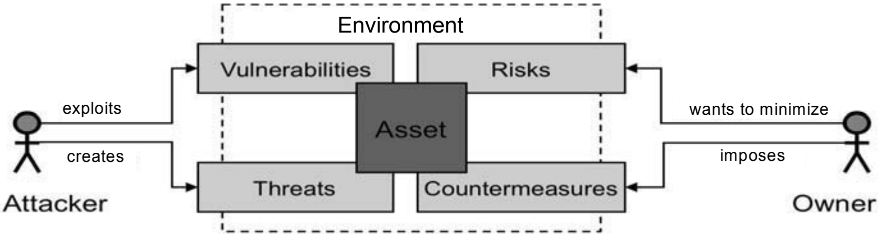


Figure 2.2. *Relationships between asset, attacker and owner*

The complexity of this issue is due to several factors. Given the architectural and technological complexity and the dynamism of assets in the context of IS, it is difficult to identify all potential vulnerabilities. Moreover, it is often difficult (too expensive, too limiting) to implement all counter-measures that are considered necessary: in most cases, the owner must assess the practical trade-off between his estimate of the seriousness of a risk and the cost of implementation of counter-measures. This is a procedure called risk assessment. The installation of all counter-measures deemed necessary increases the complexity of the original IS. Indeed, this new system, resulting from the addition of counter-measures to the initial system, should be re-evaluated. The trade-offs accepted by the owner introduce residual risks, which, over time, often result in new vulnerabilities.

Therefore, the resulting implemented set of counter-measures is usually insufficient, partly because of the ignorance of certain vulnerabilities due to the complexity of the interactions between the asset and its environment, and partly because of the applied risk evaluation methodology, typically linked to probabilistic models (usage statistics and the comfort of usage vs. perception of risks and importance of services). Obviously, there is no sufficient model, because an attacker uses his intelligence to find vulnerabilities.

Hence, as stressed by our definition, the security of ISs is a continuous process [HAL05] and not a finished product. In the ideal case, the re-evaluation of the environment and risks, estimation of asset values and research into vulnerabilities in the system (initial system and counter-measures) must be repeated periodically and systematically. There is currently no standard system to meet the requirements of each of these phases for different targets, nor there are precise specifications on timing. However, several methods for risk assessment exist ([EBIOS] [MEHARI] [CC04]), and some work has been completed within the ETSI (European Telecommunications Standards Institute) to specify these aspects in the framework of IS [TVRA].

Traditionally the security process is broken down into three aspects referring to the object of the security preoccupations and specifying what should be protected. This view of IS security is known as the CIA (confidentiality, integrity and availability) trinity. Today, this breakdown is normally insufficient because it is difficult to apply to certain new threats such as viruses, unsolicited messages or abusive usage. Thus, in the CIA classification, these threats are often classified as an availability issue. Originally, the availability was defined as resilience to attacks, but gradually the definition of availability has changed; today it is often confused with availability in the sense of reliability of operation, which, traditionally, is not a security but a Quality of Service field. This natural migration shows that extra-functional properties such as security and Quality of Service cannot be studied separately from their object or separately from one another. In general, the new system resulting from the addition of extra-functional to functional parts has to be in the center of any discussion.

Another approach refers to the question of how to protect assets and breaks the security process into phases of prevention, detection and reaction, typically referred to as PDR It is obvious that the mechanisms for achieving these phases will also be linked to aspects of Quality of Service.

Common to both approaches is the assessment of vulnerabilities, threats, risks and assets that must be done first.

2.2.3. *Trust and subjectivity in security*

There are two important aspects inherent to any security definition.

The first aspect is the notion of trust. It is quite clear that trust in any player in the studied environment removes the need for security in the same way as the total distrust prohibits any exposure of an asset to its environment and questions the notion of private property. Indeed, if any possible action on assets is perceived as a risk, the system inevitably converges to a total closure. This underlines the interdependence between trust and security: sharing a secret presumes existing trust in the same manner as the notion of comfort towards certain risks. Similarly, the existence of counter-measures presumes trust in certain actors and/or certain parts of the system. Yet, in general, there is no direct vector from trust to security or vice versa. Despite their mutual influence, we must clearly distinguish security and trust since there are very secure systems that we do not trust (nuclear plants, secret services), although the opposite is also true (credit cards, civilian aircraft).

The second important aspect is subjectivity. Indeed, for the same asset in the same environment, the risk assessment of different owners can produce radically different results. It not only depends on the presumed trust (based for example on knowledge and experience of the owner), but also on the investment and the position of the owner in relation to the object (goals, interests, anticipated usage).

Subjectivity and trust must be evaluated in the context of the targeted environment (military/hostile, civil/courteous). By definition, the subjectivity does not raise any fundamental problems regarding security assessment if the asset is isolated from the outside world, i.e. if the environment of the asset owner does not overlap with the environments of other actors (closed, proprietary systems, etc.). However, in the digital age, more often the opposite is true: the digital assets of different owners are treated by different IT systems belonging to various owners; it is often normal that during its treatment an asset traverses dozens of systems providing different services. The complexity of interactions, very different kinds of assets *per se* and different threat evaluations pose a huge problem regarding the evaluation of risks to the whole resulting system.

In general, it is impossible to compare two sets of counter-measures. These can be dictated by requirements originating from a different security policy. Therefore, an evaluation of counter-measures usually only makes sense within the scope of a given security policy. Furthermore, there is a natural interdependence between subjectivity and trust. Because of this interdependence, the countermeasure sets imposed by two different owners for the case of an IT exchange may contain contradictory or semantically non-recoverable requirements, a measure of protection required for an asset X may be unfeasible with regard to the composition of a series of sequential treatments, etc.

2.2.4. *Services and security*

The definition of security given above explicitly considers service as a potential security target. Indeed, in general any service, as an immaterial equivalent to a product, has a clearly defined value for its provider. This value is justified by the initial investment in the service infrastructure, the cost of daily maintenance and possible developments, and by the commercial or other objectives of this offer. Moreover, each service implies an interaction of its provider (as part of the service contract) with at least one additional actor, the user. In this manner, each service implies an opening to the outside represented by the user access interface. More generally, all users of a service are a true subset of the total set of actors within an environment. Therefore, each service is naturally exposed to threats in the absence of counter-measures (like access control) and the use of each service – regardless of its semantics – is closely linked to the abuse, i.e. to a non-contractual usage. Service semantics add other risks for the provider and the user: the data exchanged within the service should normally be reserved for rightful recipients; the fact that a service has been used should also be considered confidential information (privacy). Accordingly, each service requires an analysis of the system used for service provision, taking into account any actor involved in the service execution. The service contract is useful in helping to produce a homogenous security policy for the actors and to create a basis of trust between partners.

On the contrary, in general, security cannot be proposed as a service. The problem is in the definition of security, highlighted by the intrinsic and intimate link of its measures to their target. First of all, the notion of "security service" suggests securing an asset (for example, another service) that is not sufficiently secured by its owner by some third party. Yet, it would be preferable to consider the security needs and problems before the exposure of the asset to its environment (for example, before the deployment of service, and notably in the design phase). Further, the subjectivity of the security assessment of the very same asset is usually impossible to resolve, even assuming a high flexibility of service (e.g., by personalization). The protection of an asset by a security service adds at least one actor, i.e. the provider of the security service, which may already contradict the security requirements of some owners. A typical example for this situation is a service infrastructure provider. Such an infrastructure requires protective measures, but they are integrated into the infrastructure and cannot belong to any third person. Generally, however, security remains an extra-functional property, often invisible but intrinsic to each service. It is not possible to activate and deactivate security, as it is not possible to subscribe to security[1]. In other words, in every service system, there should be a security subsystem, even if the latter remains invisible to the user of the service.

[1] Although in practice we can subscribe to a "more secure" version of a given service, we must understand that the operator is obliged to implement security measures for both versions of the provided service, the more and the less secure. Paradoxically, from the service provider's point of view, security measures for the least secure version are typically more difficult to implement (blur, approximate, more subtle). This is especially true for access control.

Nevertheless, in practice, security can be offered as a service in certain scenarios. This seems particularly applicable for static, well tested, deployed and accepted services, i.e. for situations where threat models are approved and protective measures are considered sufficient (for example, through the daily experience with service provisioning: observation of the real risk under the applied protection).

New services typically provoke an unwinding of a socio-technological spiral of a mutual development of vulnerabilities and protective measures. New services define new usage and therefore set new scenarios; they are subject to potential new abuse forms. In addition, given a strong commercial pressure for the deployment of new services (effects of market competition), new services are often provided with an insufficient security analysis: vulnerabilities are naturally ignored, and threat models do not reflect the new reality. In this situation, the service is often deployed with an initial focus on its purely functional form. However, in the progress of the service deployment, various vulnerabilities are discovered. Exploited by the attackers, these become threats and result in series of attacks. Perceived as a risk by providers and users, they hinder further service deployment. Eventually, the necessary investment in counter-measures becomes inferior to the estimated losses of the hindered deployment.

Several recent services have witnessed a radical change in the understanding of their security needs. As examples, we can cite the Internet, mobile services, wireless communications, and new applications and forms of communications (peer-to-peer and peer-to-multipeer communications).

2.3. Threats and risks to telecommunications systems

2.3.1. *Role of telecommunications systems*

By forming the common denominator of any interconnection of modern information systems, telecommunications systems are at the heart of the digital age. They form the crucial interface and constitute a critical point in terms of security.

Telecommunication services, historically provided by a closed infrastructure under State control, were subject to the most radical changes in recent decades. The proliferation of the Internet and data services, the mobile revolution, the convergence of computing and telecommunications, the paradigm shift in the traditional media (television, radio, press) and market deregulation initiated in most European countries during the 1990s stress both the importance of the telecommunications systems for society and their new fragility.

Today, telecommunications systems are rightfully regarded as a critical infrastructure. Their protection becomes more a political than a commercial or a personal concern [COECRM]. All involved actors (State, enterprises, individuals) must bear their part of responsibility for the telecommunications systems that are central to their daily use. These responsibilities can be due to the actually perceived risks (and in this case depend on the position of the actor with regard to the system in question); they can also be of a legal nature.

The networks (optical, wired and wireless) are major components of modern telecommunications systems. These networks and their users are exposed to several risks. We classify these risks by using a model that distinguishes the roles of data owners and owners of infrastructures treating such data.

In the next section, we present the threat models typically used in security analysis of telecommunications systems. We then discuss several factors impacting the real risks. Finally, we present risk classification depending on the role of the actor.

2.3.2. *Threat models in telecommunications systems*

Threat models first describe the system, all actors in this system and their position in the system (for example, link, node). Then, the threat model introduces an attacker in the system and demonstrates the attacker's capacities, i.e. topological position in the system, resources, possible access, etc.

The traditional threat model to a communication channel is based on the minimal communication model minimalist involving two participants called Alice and Bob, and a communication channel (see Figure 2.3).

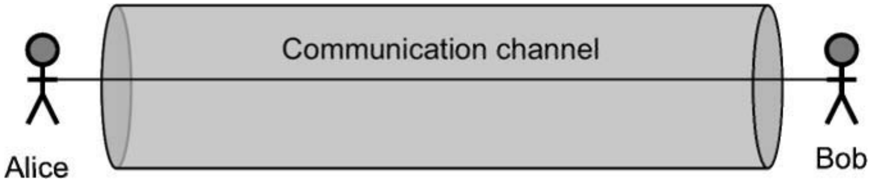


Figure 2.3. *A minimal communications model*

This model usually presumes an initial trust relationship between Alice and Bob. It is often used in cryptography, since it effectively limits possible attacks to the attacks against the communication channel between Alice and Bob. Yet, in the context of telecommunications systems, this model is not exhaustive, since other elements and vulnerabilities are present. Figure 2.4 introduces a more appropriate model, distinguishing between the two communicating parties (Alice and Bob) and at least one telecommunications infrastructure and its authority crossed by the communication channel. In general, this authority is neither Alice nor Bob but a real third party.

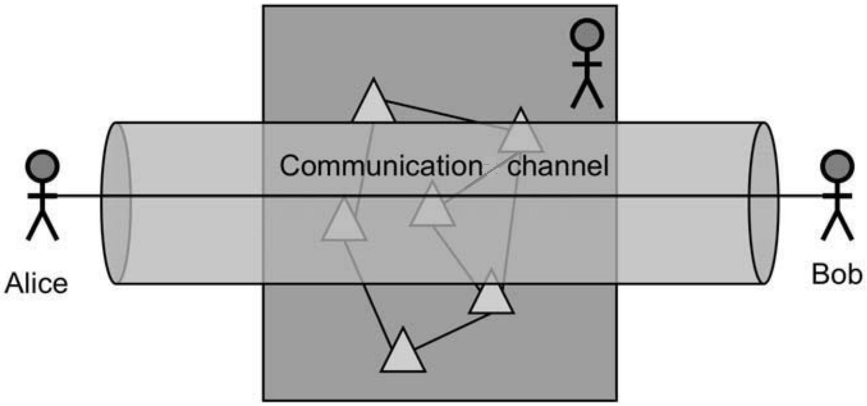


Figure 2.4. *Communications model with a telecommunications system*

The emergence of such a third party increases the complexity of the system, introduces new interfaces and vulnerabilities and may require a more complicated trust chain. It therefore widens the spectrum of possible threats.

The trust model of Figure 2.4 may have very different forms, but in practice we assume one of the following:

– Alice and Bob trust each other in the sense of the intended communications, and they both trust the used telecommunications system to correctly provide the services (private network).

– Alice and Bob trust each other, but do not trust the crossed infrastructure (public network).

– Alice and Bob trust the telecommunications infrastructure but do not trust one another; they will use the infrastructure as a trusted third party (TTP) to establish a new trust relationship.

Figure 2.5 presents from left to right the typical threats against the actors and parts of this model. In the following, the attacker is denoted as Eve:

– Eve can attack one of communicating parties (Alice in the example) using vulnerabilities in the software and protective measures used by Alice. Strictly speaking, this threat is not related to the telecommunications system. However, a terminal with a connection interface to a telecommunications system is a more open entity and is thus more vulnerable. Often, attacks are possible because of vulnerabilities in the terminal and the visibility of the terminal involved in a telecommunications service. A typical example is the execution of malicious code on the platform used by Alice through a virus or by the overflow of reception buffers.

– Alternatively, Eve can attack the communication channel linking Alice to the telecommunications system. This attack may be non-intrusive (reading the exchanged data) or intrusive (modification of exchanged data, injection of data, replay of old data). The possibility of such an attack depends on the channel. For example, a wireless channel is potentially more vulnerable against passive listening by a third person than a network cable, which normally at least requires physical access to the medium.

– Another possible attack against the channel exists within the telecommunications system. To do this, a form of access to the telecommunications system is normally required. If Eve is not the owner of the system, Eve may try to masquerade as a legitimate part of the infrastructure to attract Alice (or Bob) to use its services. In some cases, Eve can get physical access to communication channels forming part of the system or use system vulnerabilities to gain access to system components. These forms of access can allow Eve to collect information on communications between Alice and Bob and to manipulate the data flow between the two.

– The intrusion into the infrastructure permits to mount “man in the middle” attacks. In this scenario, Eve positions as a junction point between Alice (or Bob) and the infrastructure such that all communications of Alice (or Bob) to the infrastructure traverse Eve. Without reliable and mutual authentication (i.e. identity verification) between Alice (or Bob) and the infrastructure, Alice and the infrastructure cannot find this kind of intrusion. However, “man in the middle” attacks are also possible if Eve can usurp the identity of the communicating opponent. To counter these attacks, mutual and reliable authentication between Alice and Bob becomes necessary.

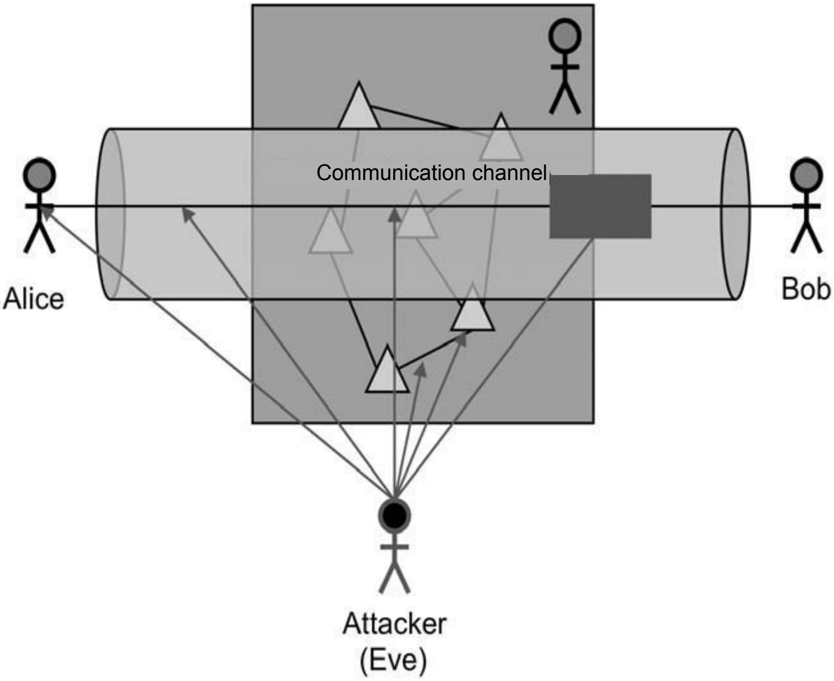


Figure 2.5. *Threat model for a telecommunications system and its participants*

In each scenario, the attack can have an intrusive or a purely destructive character aiming at the unavailability (at least temporal) of the attacked element. An attacker typically uses a combination of targeted and destructive attacks to achieve their goals.

Finally, we recall Figure 2.4: the communication channel between Alice and Bob typically traverses several telecommunications systems, operated by different authorities, and often by a superposition of information systems and different authorities. In each traversed system, all previously described threats are possible. Moreover, interconnection systems introduce new interfaces and further complicate the trust chain.

2.3.3. *Homogenity vs. heterogenity*

The heterogenity of information systems is a major obstacle to the deployment of consistent security policies. Indeed, the implementation of security mechanisms in a heterogenous environment is naturally more difficult: care must be taken to consistently achieve the globally defined goals at each subsystem level despite the diversity of the mechanism instantiation, for instance on various links and connections, on different pieces of equipment with different properties, capabilities, vulnerabilities and of varying usage. This normally results in an explosion of additional specifications and conditions. Such a deployment requires a deep, transversal understanding and good engineering of the target IS.

Moreover, assuming that the probability of presence of vulnerabilities in a realization of a function is constant, the heterogenity increases the chances of an attacker finding the vulnerability by multiplying the number of different realizations.

In addition, the management of a heterogenous infrastructure is also more complicated, and contributes considerably to the complexity of the IS in practice, which in turn introduces new vulnerabilities.

Therefore, heterogenity is seen as an important vulnerability of an information system: heterogenous IS are more difficult to protect but easier to attack. Traditionally, systems engineers are tempted to introduce ways to overcome this problem: standardization, centralization, overlays, translation (application level gateways), etc.

While heterogenity can be reasonably perceived as vulnerability for any single, determined IS, under the control of an authority suffering from the resulting high complexity of the IS, at the global level the opposite is true. Homogenity at a global scale is a major vulnerability, because it globally exposes any vulnerability. The exploitation of these vulnerabilities becomes almost certain, but in particular, the search for such vulnerabilities becomes a highly attractive task. The attacker uses a compendium of attack tools, potentially usable everywhere. By exploiting the differences in IT cultures, management strategies (e.g. periods of application patches) and security policies deployed by different authorities, an attacker is almost certain to find vulnerable subsystems and take over their control.

The best example for such a worldwide Esperanto is the Internet with its unique protocol suite TCP/IP, designed to allow access from anywhere to anywhere in the world. Today, the Internet has become the primary platform, choice number 1, for attacks against IS: everybody is accessible, everybody is exposed, and everybody is standard and compliant. With a number of services typically provided by the same implementations (Bind, Apache, Sendmail/Exim, IIS, etc.), everybody is also vulnerable. Finally, add to that fact the homogenity of user platforms, underlined by the de facto monopoly in operating systems (Windows, Internet Explorer/Firefox, etc).

2.3.4. *The Internet and security*

In a system like the Internet, interconnected, standard, open and managed by different authorities (typically by large operators) under different law systems, attacks are normal. They are different in nature (malicious, failures, oversights, bad configurations, etc.) and represent different implications, roles and judgments of players with regard to the targeted resource. The Internet is vulnerable both as an infrastructure, at different levels (routing and its convergence in BGP, name resolution through the DNS, transport over IP, UDP and TCP) and as a sum of services offered to end-users (mostly Mail/SMTP and Web/HTTP). Internet security today is a problem on multiple levels: it involves politics, technology and personal space.

In the information society of the 21st century, the Internet is the common denominator. Increasingly, it is seen increasingly as a critical infrastructure by many states (e-Government, e-Voting, e-Learning, source of information, etc.). Thus, this infrastructure requires rigorous protection. However, the application of a security policy must be coordinated and simultaneous across the whole network, among actors from different levels and countries: today, it is a step that, especially politically, seems almost impossible. It is a problem of different IT cultures of the major stakeholders, but also a problem of externality: typically, an operator of an infrastructure of the Internet is not directly concerned by outgoing attacks (unless he is directly responsible for this). In some cases, the operator is not even concerned by the incoming attacks because, unlike the target, his infrastructure is often transparent to the attack (pure transport) and, compared to any border equipment, its capabilities are enormous. In other words, the addition of control and auditing systems for the operator is expensive, but the operator does not gain anything as long as other operators do not react in the same way.

For users, the Internet is probably the biggest security threat among all available ISs. By connecting (often ignorant) users to a global information system of low security, it makes them for the first time universally accessible and universally vulnerable. For an individual citizen, the Internet provides huge opportunities by opening the door to the information society. Yet, we should not fool ourselves by thinking that this door is a one-way passage. Applications using the services offered by access and service providers and, generally, on the Internet often originate from dubious sources. They may be of low quality and be generally vulnerable. Moreover, even assuming sufficient quality of applications, users should express doubts regarding the content, the displayed addresses and the sources accessed through this. Thus, users need to be sensitized and trained. They should ask questions about the protection of their privacy, their reputation and in regard to the confidentiality of their data.

Conversely, for attackers, the Internet represents a veritable support platform by offering unsupervised, free forums for exchanging information on new discovered vulnerabilities, the source code of various exploits, lists of vulnerable destinations, stolen credentials, etc. It is even possible and indeed a current practice to collaborate in developing new attack tools.

2.3.5. *The role of the medium*

Naturally, the medium plays a central role in any distributed IS and especially in a telecommunications system. The performance, the service quality and the complexity of implementation directly depend on transmission properties and the nature of the medium. In this context, it is not surprising that the reflections on security must take into account the medium.

We must consider two aspects in particular. First, each medium brings its own properties of transmission, from which we can derive the inherent vulnerabilities. In this category, we should also add vulnerabilities in the additional management and control systems, which may be necessary, in practice or in principle, in order to use a given medium as a reliable transport. Second, new media very often justify the emergence of new services related to the nature of the new medium. These new services are again vulnerable, as has already been discussed in section 2.2.4.

This can be demonstrated through a comparison of existing transport media such as those used in wireline, optical, wireless or quantum networks. The transmission parameters of these media such as theoretical throughput, transmission error rates, signal attenuation, transmission delay, channel separation (spatio-temporal co-existence of systems, separation of flows and destinations, etc.), the nature of communications (broadcast, line of sight, shared, point-to-point) and medium access (shared, coordinated, physically impossible) are very different and normally reflect specific expected usage. Different management systems (signaling, protocols, infrastructure) are used to achieve a reliable and secure transport. Very often the usage changes over time beyond the expectations of designers. The actual use of these media must be taken into account when designing a telecommunications system employing such media, including management and security subsystems.

In the past, we have seen several examples of serious omissions around media changes. Very often, the inherent medium characteristics are not sufficiently taken into account. In other cases, designers try to adapt a security system originally designed for another medium to a given problematic. Section 2.4 includes a discussion of vulnerability evolution when changing from wire-based to wireless communications for example.

Without limiting ourselves to a specific type of communications system in the following text, we classify the risks of actors according to their positioning in relation to the digital assets in the IS in general. In practice, actors can have multiple roles and, for legal, contractual and other reasons, exhibit considerations going beyond their actual positioning towards the asset. In a risk assessment for an actor of a real telecommunications system, all these aspects must be taken into account.

2.3.6. *Risks to the infrastructure*

The owner of an infrastructure treating or carrying some digitized content not owned by him is mainly concerned with the protection of his own investment in the infrastructure and its maintenance (internal visibility). The primary preoccupation of this owner is therefore not data security but infrastructure security. Typical concerns include the following risks:

– Unauthorized access: this involves any type of unauthorized (qualitative) or extra-contractual (quantitative) access to the infrastructure, its main elements or an unauthorized access through the infrastructure. As examples we can cite the access of an unauthorized person to the internal network of a company, a non-contractual service access, access to certain content through the network of a network operator, etc. Identity usurpation is fatal in systems that control access based solely on identity.

– Infrastructural espionage: the information on the infrastructure *per se* represents a value to the owner and the attacker, because such knowledge discloses potentially critical points (weaknesses, vulnerabilities) for its operation and its use. The publication of various statistics, access to network metrology elements, the disclosure of the information about the exact topology, sampling equipment and their types are significant risks, since the data can be used to prepare and mount attacks. A typical example in a telecommunications network is espionage of services available on all components by the method called “port scan”.

– Infrastructural intrusions: the change of a “normal” behavior of an infrastructure element is obviously a risk for an operator, since it involves taking control of a private property of the operator. However, even an addition of a new, unauthorized equipment to the infrastructure is an intrusion and represents a high risk for infrastructure operators: it makes it possible to mount attacks against ignorant users or against parts of the infrastructure, and to collect key statistics. As an example, we can cite the rogue access point problem in wireless networks, or the unwanted interconnection of a secure network with a public network by a terminal bridging its connections to both physical infrastructures. We should not forget that any terminal, once connected, becomes part of a telecommunications network. His integrity is directly related to the integrity of the infrastructure. In other words, in terms of the operator, the execution of malicious code (like viruses, Trojans, spyware, malware, etc.) represents a direct risk to the infrastructure owner. Such an intrusion is a successful attack against the user who owns the terminal, which can then be used to attack users in the vicinity of the virtual network, and in some cases, even attack the infrastructure (flood, discovery). Such multi-stage attacks are often called blended attacks.

– Insufficient traceability: in order to detect and understand problems and to find the responsible person in case of investigation, a good accounting and a traceability are required. Infrastructure owners must be equipped with means to protect themselves against charges of third parties for attacks originating from their infrastructures. In some states, conservation of such data can be imposed by laws [EUDON]. Thus, measurements and statistics – metrology – in any infrastructure that can be potentially used by several actors are crucial not only for dimensioning and optimization reasons but also for security reasons. An important point today, which is mostly ignored, is that these observations should also provide evidence for evaluating the effectiveness of the deployed security measures. Such measurements constitute a basis for assessing the security assurance [CC 04].

Moreover, to improve reputation (external visibility), the owner searches to increase the reliability of the infrastructure for the users. To maintain the contracts, the owner must protect themselves against any risk making their infrastructure less available to its contractual partners. The owner must be concerned about the governance and knowledge of their infrastructure:

– The unavailability of the infrastructure: the availability of the infrastructure should be a major concern of the owner, as the owner commits to the contracts for its use. Especially regarding the Internet, denial of service (DoS) techniques have made enormous "progress". Beginning with very naïve attacks (direct flooding) during the early commercialization of the Internet, DoS becomes intelligent and focuses on the specific vulnerabilities of the involved systems (see "ping of death"). Later, DoS starts using indirect techniques (attacking a machine in order to fool it to attack other machines) usually called reflection attacks. At the beginning, DoS was purely destructive. Today, it is one of the main pillars of cybercrime. More appropriate forms in this context use different blended attacks (attacking a machine at some point of time and mount an attack from this machine later). Used in distributed forms (that take control of several machines and mount an attack from this cluster at any given point), often involving so-called botnets (see the recent case of Jeanson James Ancheta [JJAMSN]), DoS is an extraordinarily powerful attack method. Modern botnets are controlled from several control centers, which are responsible for further botnet development, maintenance and command. New capabilities are developed and steadily integrated into the existing botnets. New botnets are being deployed when necessary by using all kinds of blended attacks. Starting through an unsolicited email or drive-by downloads of malicious code from various websites, the local host's integrity is violated and the host control is taken over; the host then joins the botnet, invisibly and without annoying its rightful owner. Discretion has become an essential property of such intrusions. The infiltrated code listens to commands from the botnet's command center and downloads and prepares the actual attack code at request. The ability to attack and newly developed attack codes are rented and sold in a closed circle of cybercriminals. Finally, DoS is no longer restricted to traditional IT systems: in the interconnected world, it makes it possible to mount wider attacks based on the interdependence of infrastructures (attack infrastructure X to bring down infrastructure Z).

– Outsourcing: this has become one of the keywords of the modern business, since it makes it possible to focus and specialize on key business aspects, and therefore enables better control of the cost structure. It also allows cost cuts by using the best offer, for example through offshoring. However, unlimited outsourcing leads to an excessive loss of control of the own infrastructure and the process. It establishes much more complicated trust chains and makes the whole system more fragile. An example is the failure of Bouygues Telecom (BT) of 17 November 2005 [BTZDN]: an erroneous update applied to both HLRs (Home Location Registers) of BT, managed through outsourcing by an external contractor, caused a long unavailability of the whole network infrastructure. Besides, BT had to wait for the intervention of the external contractor to resolve the problem. This accident was independent of the actions of the actual owner.

For reputation reasons stated above, and often for legal reasons, infrastructure operators must address the protection of users and respect their wishes. Indirectly, they are also concerned by personal risks discussed in the next section.

2.3.7. *Personal risks*

This category brings together risks to a potential user of modern IT infrastructures, using the services available in or through these facilities, treating and transporting data by these facilities, etc.

The main security concerns from the user's point of view are data protection (programs, documents, pictures, etc.) and privacy protection. These considerations are often put at the forefront of security debates, attributing a secondary role to the used IT and telecommunications systems, and deriving security requirements from the primordial need to secure user data (see CIA). However, we recall here the subjectivity of security and emphasize the fact that the importance of a security consideration only depends on two criteria: on the value that the owner attaches to an asset and on the estimation of risks around this asset in a given environment. In this context, infrastructure protection cannot be seen as a logical extension of data protection.

Any user is affected by the following risks:

– Read access to private data: this risk includes unauthorized reading of the consumed, produced or transmitted content. An example is wire tapping or snooping, i.e. a passive reading of data during their transmission over a telecommunications network. Motivated by privacy protection, this category also includes read access to administrative data related to the access profile (such as personal identity, location, use statistics and billing).

– Modification of private data: an unnoticed change in the private data cited in the preceding section is a risk, because it can lead to taking control of the private data, to changes in usage statistics, accounting, etc. Note that depending on the used technology, modification does not necessarily imply read access. An example is the blind change of encrypted frames on a wireless link that uses the WEP encryption scheme according to the IEEE 802.11 standard.

– Rogue services: in the digital virtual world, the user runs the risk to connect to a rogue service. This may be due to technical faults: the examples include access to a rogue access point in a wireless network, redirecting to a fake Web server and impersonating the network. Today the impostors mostly benefit from combinations of technical flaws and socio-technological effects, e.g. when using techniques such as phishing. Phishing involves attracting users through the use of user interfaces mimicking known, authorized interfaces (e.g. Web portals) and false promises. The actual attraction to a rogue service typically uses human faults (ignorance, curiosity, greed, laziness). Only in a next phase they will try to exploit technical faults (loose, non-mutual two-phase authentication).

– Non-contractual access properties: the user runs the risk of not obtaining contractual access properties such as the reliability of access, the negotiated data rates, time and duration of connection, etc.

– The fragility of the execution platform is one of the biggest risks today, orthogonal to the risks discussed earlier. Along with the information about the source, destination and data protection discussed above, the user must care about the integrity of the platform and the used programs. If the used platform is not reliable, malicious access (through a virus, a Trojan and any type of “malware”) to the private data of the user, including the user’s identity and credentials is possible. This category also includes all the spyware used to (often illegally) spy on user's activity, to establish a commercial, medical, political user profile, etc.

– Identity usurpation is a major risk to authorized users, since any act committed under a spoofed identity can be falsely attributed to an authorized user. This in turn allows access to and the ability to modify private data, etc.

2.4. *From wireline vulnerabilities to vulnerabilities in wireless communications*

To illustrate the abstract concepts introduced throughout this chapter, in this section we give an example of increasing vulnerabilities in a telecommunications system using wireless technologies[2]. We observe that the same IS (corporate networks, access networks, local/personal area network, network operators, services, etc.) ported from wireline communications to wireless communications suffers from major security problems. The aggravation of the security situation can be discussed along three main axes discussed in the following sections.

We draw attention to the fact that we are talking about possible and not certain threats. Indeed, any technology can be used in a way such that a given vulnerability does not have any real impact. On the other hand, it is perilous to believe that we can impose constraints on the use of any technology in modern society.

2.4.1. *Changing the medium*

The wireless medium, in the sense of radio-based broadcast transmissions, is very vulnerable by nature, much more vulnerable than the wired medium. The overall role of the medium for IS has been discussed in section 2.3.5. The medium allows wireless access for any actor: reading, injection, deletion and modification of data are possible for any actor in most configurations. In addition, all communications are purely virtual: generally, we cannot delimit the perimeter of the network (because of physical properties of the medium: the signal attenuation is strong, but the multi-path propagation, reflections/refractions, etc. often produce surprising results), nor can we distinguish different connected terminals. In other words, the medium does not limit the circle of actors involved in the processing of exchanged data. It does not detect whether access to the medium or to the transmitted data took place during the transmission.

[2] We recall here that the wireless transmissions can also be optical (infrared, laser), directional, acoustic, etc. However, in practice the term is now mainly used for networks based on radio transmissions.

For an attacker, such a wireless medium is often more attractive because it does not require the physical presence of the attacker. Well equipped, an attacker is able to mount attacks against natural medium vulnerabilities while remaining outside of the attacked area (for example, a parking lot attack on an enterprise IS). In addition, attacks can be easily automated or at least semi-automated: the equipment can record all received frames for spying on the encountered wireless infrastructures (wardriving), or for an autonomous *a posteriori* treatment (dictionary attack, brute force attack), even without exploiting any particular flaws in the security measures usually implemented in such networks (mainly access control, confidentiality and integrity).

To overcome the transmission problems related to reliability and security of communications, management and control systems, finite state machines and protocol stacks used in these networks often exhibit elevated complexity. Reflected within the network interface card, within the drivers and in the dedicated applications, this complexity results in new vulnerabilities.

2.4.2. *Wireless terminals*

The terminals used in wireless networks are characterized by their portability. They are small, often equipped with a restrained human-machine interface (HMI), limited in terms of processing power and storage and powered by a battery.

These characteristics have a significant impact on the security of the terminal, and, by extension, on the IS using it. The limited user interface often poses problems in pairing and access control phases (how to enter a password in a pair of headphones, how to establish a unique identity of a USB key, etc). Limited computing (CPU) and storage (memory, disk) capabilities introduce constraints with regard to possible calculations. For example, it is arduous to base the security of an embedded device, for example, of a sensor or of a mobile phone without a dedicated module, on public key cryptography, since the necessary private key computations take too much time and energy.

Battery management implies several changes in the behavior (unexpected on/off, technically close to mobility) and requires additional management systems (standby management, mechanisms for paging, etc). In addition, the development of battery technology is constant but linear. It cannot follow the exponential development speed typical for microelectronics (Moore’s law). Often referred to as the wireless security gap, this problem can be overcome through a high quality circuit design, sophisticated standby management and protocols and complex adaptive power management, complicating the terminal and making it potentially more vulnerable.

2.4.3. *New services*

Beyond these aspects, networks based on radio transmissions add a degree of freedom to every transmission: the spatio-temporal context. With wireless communications, it is reasonable to talk about mobility, nomadicity and location of users connected through this medium. This new freedom justifies the implementation of new services for mobility or localization support (location-based services, etc). These new services are not reserved to wireless communications, but practice shows that with wireless technology they become truly interesting: mobility is not limited to wireless and wireless does not imply mobility, but in reality there is a considerable overlap.

Mobility represents a known problem for security considerations, not only because it introduces new mechanisms and subsystems and therefore results in a higher complexity, but before all through the presence of several potential authority domains. This complicates the chains of trust. It is often necessary to provide services to users from another authority domain, subject to a different security policy. Since security policies are not directly comparable, this often results in irresolvable requirements and cannot be realized. However, even the reception of mobile users from the same authority domain is complicated: the network has to verify that, after a period of absence, the configuration of the mobile user is still consistent with the security policy requirements of the domain. In practice, this often results in drastic measures regarding the access rights of mobile users, or mobile users have to pass through a quarantine period prior to regaining full user rights. As a result, mobile systems are normally more vulnerable, both from the point of view of users and of operators. It is difficult to fulfill all security requirement in the CIA sense, but it is even more difficult to implement correct non-repudiation properties, a sufficient traceability (e.g. for billing) not leading to new abuse (anonymity, respect for the privacy requirements), to verify the lack of viruses, malware, etc., on the mobile terminal.

The security of mobility must be treated with great caution. The problem is that the security mechanisms often become active simultaneously to the typical mobility mechanisms, like. handover treatment. These mechanisms interfere and extend handover delays; hence, they become critical to the performance of the resulting IS. Trade-offs are often required to achieve acceptable results for the provided service.

In addition, as already discussed in section 2.2.4, the provision of new services is dangerous. The addition of these services changes the functional part of the IS and must be followed by a new risk assessment, taking into account the previously provided services, the impacts of changes on these, and the desired security for new services. Very often, none of this is done in practice.

2.5. Conclusions

Security is a major problem in modern information technology systems. It will certainly become even more critical in future technologies (sensor networks, future Internet, autonomous networks, 4G, etc.). The democratization of information and communication technologies materialized through the interconnection of various systems (wireline, wireless, autonomous or others) makes data and infrastructure protection considerably more complex.

Despite the intrinsic security problems, wireless networks continue growing in several vertical markets such as telecommunications, industrial applications, M2M (machine-to-machine) and home automation. It is important to understand the difficulties related to the setup of such networks, the service provisioning quirks in this new interconnected and communicating world, but it also crucial to recognize new opportunities that they offer.

Unable to find by themselves a good trade-off between security and its cost for the required services, enterprises and citizens become more exigent regarding the security guarantees given by service providers. Security is therefore one of the major challenges for the marketing of services and products in IT and goes beyond purely technical dimensions. Today, security concerns all involved actors (network operators, service providers, systems integrators, users, the State). Legislation, industry, academics and users are called to work together to develop improved methodologies for the security process.

The first barrier has been taken: an increased sensitivity to security problems is reflected in today’s political debates, product marketing and customer requests. At the same time, a range of research projects is being conducted by academics, industry and through various consortia and standards bodies, aiming to make improvements and develop more robust solutions.

However, we must understand that there can be no standard security solution for everyone. This is due to the very different risk appreciation around a given asset in a given environment, but largely because of the increasing complexity of ISs. In addition, the development of new services and products brings new vulnerabilities, the severity of which cannot be measured in advance because it depends among other things on the scale of deployment. Security remains a process that must accompany the development of an information system. As the world becomes more connected and more communicative, security concerns are likely to worsen in the future. A critical issue in this context is the protection of privacy.

This chapter makes a statement on the vulnerabilities of semi-open systems such as modern information systems. In discussing several security issues, we have classified the risks involved in using these systems.

2.6. Bibliography

[BERCN] WORLD INTELLECTUAL PROPERTY ORGANISATION (WIPO), *Berne Convention for the Protection of Literary and Artistic Works*, September 9, 1886, http://www.wipo.int/treaties/en/ip/berne/trtdocs_wo001.html (December 2008).

[BTZDN] A computer failure at Bougues Telecom paralyzes the network (in French), http://www.zdnet.fr/actualites/internet/0,39020774,39183813,00.htm (December 2008).

[CC 04] CC (2004), *Common Criteria for Information Technology Security Evaluation*. Part 1: Introduction and general model, Part 2: Security functional requirement, Part 3: Security assurance requirements. Version 2.2, January 2004.

[COECRM] COUNCIL OF EUROPE, *Convention on Cybercrime*, November 23, 2001, http://conventions.coe.int/Treaty/en/Treaties/Html/185.htm (December 2008).

[DOD 03] US DEPARTMENT OF DEFENSE, *Dictionary of Military and Associated Words*, 2003.

[EBIOS] DCSSI (2004) *Expression des Besions et Identification des Objectifs de Sécurité : EBIOS*, Technical Report, Direction centrale de la sécurité des systèmes d'information (organisme gouvernemental français), 2004 http://www.ssi.gouv.fr/en/confidence/ebiospresentation.html (January 2007).

[EUDON] EUROPEAN PARLIAMENT AND COUNCIL OF EUROPE, *Proposal for a directive on the retention of data processed in connection with the provision of public electronic communication services*, December 14, 2005, http://europa.eu/bulletin/en/200512/p104029.htm (December 2008).

[HAL 05] HALLBERG, J., HUNSTAD, A. and PETERSON, M. (2005) "A Framework for System Security Assessment", *Proceedings of the 2005 IEEE Workshop on Information Assurance*, June 15-17 2005, West Point, New York.

[JJAMSN] The case of Jeanson James Ancheta, http://www.cybercrime.gov/ancheta Arrest.htm, http://reviews.cnet.com/4520-3513_7-6427016-1.html (December 2008).

[MEHARI] Club de la Sécurité des Système d'Information Français (CLUSIF) (2004) *MEHARI – Version 3.0*, Technical report, 2004, https://www.clusif.asso.fr /fr/production/mehari (December 2007).

[TVRA] TS 102 165-1, *Method and Proforma for Threat, Risk, Vulnerability Analysis*, ETSI, December 2006.

This page intentionally left blank

Chapter 3

Fundamental Security Mechanisms

3.1. Introduction

For a better understanding of security solutions described in what follows in this book, it is useful to first present the fundamental mechanisms of network security. This chapter introduces mostly security services notions, two cryptographic families, hash functions, electronic certificates and PKI, the SSL and IPsec security protocols, etc. The chapter also describes the VPN (Virtual Private Network) technologies supporting virtual private network implementation and several authentication techniques, and access control solutions like firewalls and intrusion detection systems.

3.2. Basics on security

3.2.1. *Security services*

Security services refer to security concepts contrary to security mechanisms which include the set of cryptographic tools useful for implementing security services. The X.800 standard [X800] defines the security services (except for replay detection) as follows:

– availability: "the property of being accessible and useable upon demand by an authorized entity";

Chapter written by Maryline LAURENT-MAKNAVICIUS, Hakima CHAOUCHI and Olivier PAUL.

– access control: “the prevention of unauthorized use of a resource, including the prevention of use of a resource in an unauthorized manner”;

– data integrity: “the property that data have not been altered or destroyed in an unauthorized manner”;

– data origin authentication: “the corroboration that the source of data received is as claimed”;

– peer entity authentication: “the corroboration that a peer entity in an association is the one claimed”. Note the clear distinction between “identification” and “authentication”. Identification refers to an entity (user, equipment) claiming its identity by providing an identifier (name, pseudonym, email address, IP address, domain name), or the procedure to find the Identity of a user among N users known by the systems under several features. Authentication consists of proving the claimed identity by providing one or several authentication elements;

– confidentiality: “the property that information is not made available or disclosed to unauthorized individuals, entities or processes”;

– replay detection (not defined in X.800): a replay detection consists of an entity to detect that received data are duplicated from a previous exchange. Some data might have been sent in a secure manner by a legitimate entity, but they can be copied and injected again to the same destination. Data are still authentic but they are already processed; thus, it is necessary to detect replay to avoid them being processed several times.

Encryption mechanisms enable the implementation of data confidentiality. Most of the time, the services of data integrity and data origin authentication are implemented by the same security mechanisms: hash function and MAC generation (see section 3.2.4). The replay detection mechanisms are based on injecting a sequence number in each of the transmitted information elements.

3.2.2. *Symmetric and asymmetric cryptography*

Since the 1970s, two cryptography families emerged [SCH 96]. In symmetric cryptography, the enciphering and deciphering systems know the same cryptographic key, while asymmetric cryptography (known as public key cryptography) is based on two complementary keys – the public and private keys – one of them for encrypting and the other one for decrypting. Both families are hereafter described with a few examples of algorithms that are commonly used today, their advantages and drawbacks, as well as their complementarities.

Note that older cryptographic algorithms were based on the secret of the algorithm itself. This means that as soon as the algorithm was cracked, the cryptographs needed to invent a new one. The novelty of symmetric and asymmetric algorithms was to make public the whole enciphering/deciphering processing and to externalize the secret into a secret parameter also called the “cryptographic key”.

3.2.2.1. *Symmetric cryptography*

Symmetric cryptography is based on the usage of the same key to encrypt and decrypt data. These keys are called symmetric keys (sometimes secret keys). In the context of exchanges over a network, a transmitter encrypts data with a key and the destination entity decrypts the data with the same key. If the symmetric algorithms are efficient, and make it possible to reach a high data rate when encrypting/decrypting, they raise the problem of establishing the same key between the transmitter and the receiver, however. Sharing a key with each possible communicating entity, even in a closed group of entities, is a very high constraint, and rapidly leads to a big number of keys to be managed. Thus, it is better automating the establishment of these keys (see section 3.2.2.3).

The most well known symmetric algorithms are, in the chronological order of their definition: DES (Data Encryption Standard), 3DES (pronounced “Triple DES”), and AES (Advanced Encryption Standard). DES was invented in 1977 by IBM as the public encryption algorithm with secret keys of 56 bits and input of 64-bits data blocks. DES is based on permutation mechanisms and exclusive OR gates. These fast operations make DES highly efficient, but brute force attacks are still able to crack the 56-bits keys by trying any combinations of keys. The 3DES algorithm was more robust and was successor of DES; 3DES applies DES three times, one after the other; the 3DES key is maximum 168 bits (3*56=168) and applies to the same input block size (64 bits). The 3DES algorithm is not always efficient from an encryption rate point of view, robustness to brute-force attacks, etc., so an international competition was launched in 1997 to elaborate a new algorithm to replace 3DES. After several selection steps, the Belgium Rinjdael algorithm was selected for its fast processing time, its portability on several platforms (hardware and software, 8 and 32 bits), several supported key lengths, etc. Thus, we talk indifferently about AES or Rinjdael. AES is the generic name of the algorithm winning the competition. It relies on inputs of 128-bit blocks and key lengths of 128, 192 or 256 bits.

Symmetric algorithms can work according to several modes. Usually we distinguish the ECB (Electronic Code Book) mode which consists of encrypting each of the blocks independently. Thus, the operation of encrypting a message consists of fragmenting a message into blocks of the expected size (dependent on the selected algorithm). Each of these blocks is then encrypted independently. The drawback of the ECB mode is that two similar blocks give similar encrypted blocks outputs. This makes ECB vulnerable, as a spy on the network can detect two similar encrypted blocks, attempt to guess their contents, and perform a brute-force attack by testing any combination of the keys until the assumed cleartext message is obtained. The CBC (Cipher Block Chaining) mode, also known as the “chaining mode”, consists of processing a block by injecting the last computed ciphered block into the processing, such that encrypting two similar blocks leads to two different blocks. Attacks are more difficult to implement.

According to the algorithms, the encryption mode used and the length of the message to be encrypted, it is sometimes necessary to align the message on a whole number of blocks. Aligning consists of appending to the message prior to encryption some padding information, that is, some meaningless information.

3.2.2.2. *Asymmetric cryptography or public key cryptography*

Asymmetric or public key cryptography relies on two encryption keys, called “asymmetric keys”. Both keys are generated at the same time and play a complementary role in that the encryption with one of the keys needs to be decrypted with the other key. Each key plays a specific role. The private key must be known by only one entity and can be used for authenticating itself for instance. The public key can be largely published and it is better that public keys are largely published in order that any other entity can perform authentication. Obviously, knowledge of the public key does not enable us to deduce the complementary private key.

To authenticate the origin of message in a communication over a network, the transmitter must use its own private key, for instance to generate an electronic signature (see section 3.2.4) that it will append to the message before transmission. The receiver who knows the public key will be able to verify the validity of the signature and will have guarantees regarding the origin of the message.

To ensure the confidentiality of a message, it is necessary to encrypt the transmitted message with the public key of the receiver. This public key is known by all entities and can be served to any entity to encrypt a message. However, the complementary private key is only known to the destination of the message; the receiver will be the only one able to decrypt the message. The property of confidentiality is thus obtained.

using these classical examples of usage of the public/private keys, we must understand that all the difficulty is related to the guarantee provided that a public key is truly associated with the unique identified entity. This association entity/public key is fundamental. With no such reliable guarantee, it is useless implementing security over a network. A first idea would be that each entity previously registers every public key of its correspondents, but this would be as bad as the symmetric cryptography with a number of keys to be managed. A second solution now that largely applies consists of defining a trusted third party, that is, an entity in which a large number of entities have trust. This trust is created by the knowledge of a public key associated with the trusted third party. This trusted third party can for instance take the form of a certification authority whose role is to issue electronic certificates (see section 3.2.5), i.e. some data structures binding a public key to an entity, and signed by the certification authority.

RSA (Rivest, Shamir, Adleman) is the most well known asymmetric algorithm. It is based on the theory of prime numbers and encryption keys with classic key sizes of 512 classic, 1,024, 2,048 or 4,096 bits. Today a good level of security refers to keys of at least 2,048 bits. If the robustness of cryptographic algorithms is dependent on the length of the keys used, it is meaningless comparing the robustness of symmetric and asymmetric algorithms with respect to the same length of keys. Indeed, cracking an asymmetric algorithm (in order to discover the used key) does not require testing all the possibilities for keys like the symmetric algorithms; breaking RSA for instance is like successfully factorizing a big number into two primary numbers; in terms of robustness, 1,024-bit RSA keys are equivalent to 80-bit symmetric keys or 2,048-bit RSA keys to 112-bit symmetric keys.

To accelerate the deployment of public key cryptography, developers organized into the commercial organization RSA Laboratories to define the syntax of different data structures related to public key cryptography like private keys but also the certification requests, etc. These de facto standards were included in a number of standards and are known as PKCS (Public-Key Cryptography Standard).

3.2.2.3. *Complementariness between the two cryptographic systems*

Security protocols use both security protocols, each having a specific usage:

– symmetric cryptography (or secret key cryptography) makes it possible to protect high bit rate data exchanged over a network; the processing speed of symmetric algorithms is used;

– asymmetric (or public key) cryptography is used to initialize a secure connection between two entities of the network by enabling those entities to authenticate each other and to establish a symmetric key in a confidential way.

3.2.3. *Hash functions*

Hash functions aim to give a representative result of the message's content over a limited number of bytes. They are pretty like a more sophisticated CRC (Cyclic Redundancy Check).

The awaited properties of these hash functions are as follows:

– a result on a limited number of bytes (usually 16 or 20 bytes);

– inability to recover the original message from the outcome of the function;

– two messages that differ by only 1 bit produce two results that differ by at least half a bit.

Several terms for hash functions like irreversible functions or one-way functions are used indifferently. Several other terms for the result of the hash functions are used like hash or fingerprint. Thereafter, the term "footprint" is used.

Several hash functions are today defined like a series of MD (Message Digest) functions MD2, MD4 and MD5 which give a 16 byte fingerprint, but also algorithm SHA-1 (Secure Hash Algorithm-1) which gives a 20 byte long result. Today MD5 and SHA-1 are the most frequently used. However, MD5 was recently proven vulnerable to attack by collisions [LLO 06]. Indeed, within hours, it is possible to find a message leading to the same footprint MD5.

A series of algorithms named SHA-256, SHA-224 and SHA-512 have been invented by the NSA (National Security Agency) since 2000. They all derive from SHA-1, but give results in a greater number of bits (256, 224 and 512 bits). SHA-256 is today the most popular and is considered as the successor of SHA-1.

3.2.4. *Electronic signatures and MAC*

The electronic signature or a MAC (Message Authentication Code) appended to a message has a twofold objective: to enable the recipient to authenticate the origin of this message and to prove its integrity. Implementation of the electronic signature and MAC uses hash functions and symmetric or asymmetric keys. In the case of symmetric cryptography usage, we can talk about "MAC" (or MIC – Message Integrity Code). In the case of asymmetric cryptography, we can talk about "MAC", but we prefer talking about the electronic signature.

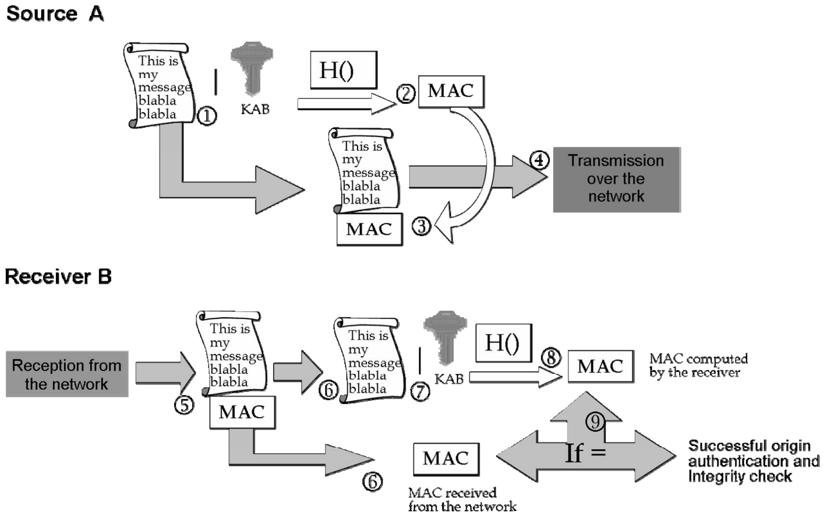

Source A
This is my message blabla blabla
①
KAB
H()
②
MAC
This is my message blabla blabla
MAC
③
④
Transmission over the network
Receiver B
Reception from the network
⑤
This is my message blabla blabla
MAC
⑥
This is my message blabla blabla
⑦
KAB
H()
⑧
MAC
MAC computed by the receiver
⑨
MAC
If =
⑥
MAC received from the network
Successful origin authentication and Integrity check

Figure 3.1. *Generation and verification of a MAC (symmetric cryptography)*

This section describes two ways to generate a MAC and to check the validity of a MAC depending on the cryptography in use. In the case of symmetric cryptography, as shown in Figure 3.1, the generation of the MAC requires steps 1 to 4 from the source (A), while verification by the receiver (B) requires steps 5 to 9 for its validity. The source first concatenates the message and the key K_{AB} in a specific order. The result of this first step is then hashed by the hash function H() and the MAC is then obtained (step 2) and appended to the message (step 3) before transmission over the network (step 4). Upon receipt of the message (step 5), the receiver separates the message from the MAC (step 6). In the message, a “local” MAC is then calculated locally following the same steps as the source (steps 7 and 8). The local MAC is then compared to the MAC received from the network. In the case of equality (step 9), the message can be considered authentic and unaltered. On the one hand, the calculation of a MAC requires that the source knows the correct shared key K_{AB}, so if only one entity is assumed to know this key, then the source is truly the claimed entity. On the other hand, alteration of the message or the MAC during transfer over the network will lead to the receiver calculating a MAC different than the source MAC, and there would be no equality.

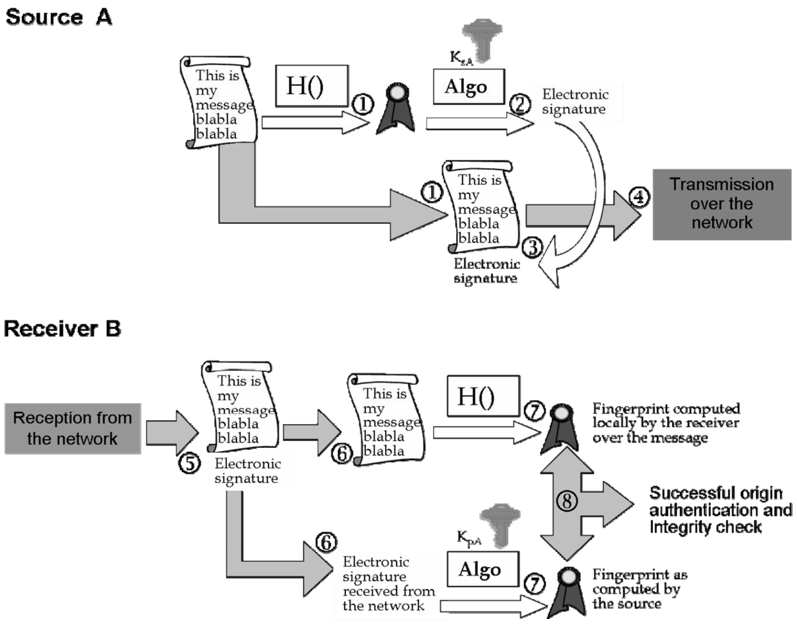

Source A
This is my message blabla blabla
H()
①
KSA
Algo
②
Electronic signature
①
This is my message blabla blabla
③
Electronic signature
④
Transmission over the network
Receiver B
Reception from the network
This is my message blabla blabla
⑤
Electronic signature
This is my message blabla blabla
⑥
H()
⑦
Fingerprint computed locally by the receiver over the message
⑧
Successful origin authentication and Integrity check
⑥
Electronic signature received from the network
KpA
Algo
⑦
Fingerprint as computed by the source

Figure 3.2. *Generation and verification of an electronic signature (asymmetric cryptography)*

In the case of asymmetric cryptography, as shown in Figure 3.2, source A starts generating a fingerprint with the hash function H() (step 1), then it encrypts the fingerprint with A's private key (step 2). This gives an electronic signature that is appended to the original message (step 3). As such, A is sending over the network: the message and the signature (step 4). Upon reception (step 5), receiver B separates the message from the signature (step 6). In step 7, on the one hand, B calculates a local fingerprint over the received message; on the other hand, it decrypts the received signature using the same algorithm and A's public key to obtain the fingerprint that was calculated by A. In the case of equality (step 8), the message is proved to be authentic and unaltered. First the source having generated the signature must possess the proper private key K_{SA} so it makes sure it is entity A. Moreover, if there was any alteration of the message transmitted over the network, it is clear that the fingerprint calculated locally by the receiver would have been different from that calculated by the source (obtained by the receiver after deciphering the signature).

Note that the MAC generated by the symmetric cryptography is not useful for non-repudiation, as it is not possible to prove retrospectively whether party A or B delivered the signed message. The electronic signature overcomes this problem by making it possible to allocate a signed message to a specific entity.

3.2.5. *Public Key Infrastructure (PKI) and electronic certificates*

On the Internet, many standards such as S/MIME, SSL/TLS, IPsec and SSH are mainly based on the use of public keys to secure exchange of electronic messages, electronic transactions and remote connections. They suffer from a fundamental security problem: how to trust an association that binds a public key to its owner. This binding is particularly critical to authenticate entities (users, Web servers, etc.), and to ensure the confidentiality and integrity of their exchanges. It is therefore essential to regulate the management of public keys thanks to PKIs.

A PKI [HOU 99] supports both organizational and technical aspects to perform the following functions: the generation of public/private keys, their distribution to their owners (when initializing a new entity in the PKI) and the publication, validation and revocation of public keys. Generally, PKIs are based on electronic certificates and certificate revocation lists (see sections 3.2.5.2 and 3.2.5.3), but sometimes, as outlined in section 3.2.5.1, the mere publication of a public key in a secure directory is enough to implement a PKI. Sometimes, the key pair is generated by the owner itself who only requests the issuing of a certificate from the PKI.

On the Internet today, many PKIs exist, and they come in the form of certification authorities that are organized hierarchically. The higher certification authority delegates management of a portion of certificates to the certification authorities below. More exactly, PKIs distinguish two roles of authorities:

– Certification Authority (CA): the CA is the only authority that holds the private key of the CA and thus is empowered to issue electronic certificates and certificate revocation lists.

– Registration Authority: one (or more) registration authority is associated with a CA and acts as an interface with users. It filters certification requests coming from users through a variably strict control over the identity of the requester. The registration authority is also in charge of publishing and validating electronic certificates generated by the certification authority. Finally, it checks the authenticity of any certificate revocation request and publishes lists of revoked certificates.

Today, PKI services are marketed by companies authorized to issue and manage electronic certificates. These companies are known as “Certification Service Providers” (CSP). The company Verisign is the oldest and best known CSP.

Within a PKI, several levels of certificates are possible depending on the usage of certificates, the expected security level, etc. It is clear that a certificate issued for a user to protect its e-mail messages does not need the same level of security than a certificate issued to a company wishing to sign contracts electronically. Thus, when enrolling a user in a PKI, the verification of identity performed by the registration authority can be very thorough with face to face control (identity card, official letter from the company authorizing the enrolment of the employee) or pretty non-existent, for instance, with a simple electronic message sent to the person for invitation to download his certificate over the Internet. Of course, according to the usage of certificates, the risks involved in financial terms are very different; in many cases, it is possible to take out insurance for damages from specialist insurers with an amount appropriate to the level of trust and security desired.

3.2.5.1. *DNS hierarchy servicing as a PKI*

It is possible to match a PKI hierarchy onto the DNS (Domain Name Server) thanks to the DNS security extension DNSSEC [RFC4034]. DNSSEC is designed to publish, in a DNS server, public keys and certificates associated with domain names in the form of DNS records (RR – Registration Record). The integrity and authenticity of these records are maintained using electronic signatures (RRSIG RR) also published in the DNS. These signatures are calculated by the secure DNSSEC zone on DNS records needing protection.

3.2.5.2. *Electronic certificates*

Certificates are designed to securely link a public key to an entity (user, server, etc.). These certificates correspond to a data structure whose most common format is provided by the X.509v3 standard [HOU 02] and includes: a serial number, a public key, the identifier of the public key's owner, the date of validity (start date and expiry date), the identifier of the CA issuing the certificate, and the signature of the certificate with the CA's private key. The signature by the CA guarantees the authenticity of the certificate. It is sufficient that an entity trusts the CA and knows the CA's public key for this entity to have confidence in all the certificates issued by this CA. If there are multiple levels of CAs, then each CA must have a public key certificate signed by the higher CA, including its role as a CA; this certificate proves that the CA is empowered by the higher CA to manage certificates. Only a root CA signs itself its own certificate; this certificate is called a "self-signed certificate". In a hierarchy of CAs, a chain of certification of a certificate is formed by all CAs from the root CA down to the issuer CA. This concept of certification chain is helpful when verifying a public key; the verifier needs to verify all the certificates of the chain.

In practice, before using a public key of a remote device to secure exchange, we must proceed with several verifications, as follows:

– the date of validity: when verification occurs, the certificate must be in its window of validity. This implies in particular that the system of the verifier is on time with the right date. Very often, in practice, the problems causing rejection of certificates are due to bad clock synchronization or a too fast generation of certificates giving a validity period of zero;

– confidence in the CA: the CA who signed the certificate must be recognized as trusted. Several cases might occur:

- the CA issuer is preregistered into the verifier device, in which case this verification is immediate,

- the CA issuer belongs to a hierarchy of CAs: theoretically, it is necessary to verify the entire certification chain. This requires getting all the certificates of the CAs of the chain, having confidence in the root CA, and then checking one by one the validity and authenticity of all the certificates, from the root CA down to the CA issuer, by following the same steps as described here. In general, the verification systems do not proceed to certificate chain verification; they only verify the presence of the certificate of the CA issuer in their certificate store,

- the CA issuer is not trusted because its certificate is not known by the verifier system as a trusted entity, or the root CA of the CA hierarchy to which it belongs is not recognized as trusted;

– the non-revoked state of the certificate: the proof must be established that the certificate has not been previously revoked. Several solutions are described in section 3.2.5.3;

– the validity of the CA's signature: the electronic signature in the certificate must be valid.

3.2.5.3. *Verification of the non-revoked state of a public key*

A public key can be revoked for instance after private key compromising or the inability of the CA to continue its certification service. The CA has several ways to disseminate this information. The most prevalent today is to publish a list of revoked certificates called the CRL (Certificate Revocation List). According to the X.509v2 standard [RFC2587] in effect, the CRL must contain: the list of serial numbers of the revoked certificates, the date of issue of the CRL, the date of the next publication of CRL and the signature of the CA. The obtained CRL must be processed periodically by the CA and made available to users by the registration authority.

Part of verification of a certificate consists of downloading the CRL and verifying that the certificate is not part of the CRL. For easier localization of the CRL to be downloaded, the certificates often include the URL (Uniform Resource Locator) where the CRL can be downloaded with a reference like HTTP or URI (Uniform Resource Identifier) if an LDAP download is practiced. The major drawback of CRL is its cost in bandwidth which can be very high if the CRL is big; some methods exist to reduce the size of a CRL. In addition, the lack of freshness of the revocation information in CRLs is critical as CRLs are generated with a high or low frequency, thus implying that certificates being revoked in one of these intervals of time are only known as revoked in the time interval after.

With the effort to overcome these disadvantages, some servers were defined to offer online verification of certificates. Two types of servers are currently emerging:

– the OCSP (Online Certificate Status Protocol) server [RFC2560] is under the supervision of the CA in charge of managing the certificate. Its role is to look into the CA's directories to find the status of the requested certificates. Thus, an OCSP client sends in its OCSP request the serial number of the certificate and the OCSP server sends back the status associated with the certificate. The OCSP server does not check the chain of certificates. The disadvantage of this method lies in the need for the CA to have an OCSP server and for the client to trust this OCSP server that might be under the supervision of any administrative organization;

– the SCVP (Server-based Certificate Validation Protocol) server [FRE 07] can check the entire chain of certification of a certificate. As such, the performed operation is more complex than the OCSP server's. It enables applications wishing to verify the status of a certificate to centralize the verification operations into a SCVP server. The SCVP architecture assumes that the server is located in the SCVP client's LAN and therefore belongs to the same area of administration. In this case, the relationship of trust between the SCVP client and server is easy to build.

3.2.5.4. *Today's problem of using electronic certificates*

It should be noted that certificates are critical in the implementation of secure communications. Indeed, it is useless to desire protecting exchanges (e.g., with encryption) with some other party if you are not sure to be in communication with the claimed entity. The management of certificates must be very careful on that point. Indeed, an entity must only accept a certificate as safe if the CA is known to be trustworthy.

A CA is considered "trustworthy" if it is preregistered in the certificate store of the device and identified as "trusted". This preregistration assumes that the public key of the CA is stored into the device, which is then able to check the validity of any certificate issued by this CA for signature verification. If the CA is not recognized by the device as "trusted", then the device asks the user himself to take a decision as to whether to accept this certificate for exchanges to continue. It is at this precise moment that the user takes the greatest risk. Indeed, he can be easily abused by an entity that can take any identity proving its identity by only showing a certificate generated from scratch. From a security point of view, it is unacceptable to let a user without any special knowledge on PKI take the decision to trust a certificate. It would be much more reasonable that the CAs define trust relationship among themselves, so the users can easily trust a CA that is presented by its own CA as trustworthy. The mechanisms in use could take the form of cross-certificates, where each CA could certify the root certificate of the other PKI.

Most of the devices using electronic certificates do not check their status of revocation, which weakens the security of electronic exchanges. Self-signed certificates are still frequently encountered on websites and are of no use to judge the authenticity of a server. Finally, the certificates are not intended for use in a mobile environment; authentication between the mobiles and the network is a prerequisite for connecting the mobile to the network; in the case of mutual authentication based on electronic certificates, the mobile is therefore unable to decide the validity of the access equipment representing the network because it does not have access to the network, and as such cannot verify the non-revoked status of the certificate.

3.2.6. *Management of cryptographic keys*

Cryptographic keys to secure connections can be of several types. The group (or multicast) key serves in the case of multicast applications to ensure that only group members can access the content of flows and that they are the only ones able to contribute to the generation of flows of the application. Today, research works are still improving techniques for managing group keys; such techniques vary greatly according to the number of members of the group and the ability to prioritize the management of these keys. Chapter 14 presents management solutions adapted to multicast keys for ad hoc networks.

Unicast keys are used in the case of communications between two parties and key management techniques are today well developed. All the difficulty is related to the generation and renewal of these keys. In recent years, some properties have been highlighted in these unicast keys. For instance, the PFS (Perfect Forward Secrecy) property requires that the keys used to protect a connection at different time intervals are independent of each other; this prevents the disclosure of one of them leading to the discovery of all others and therefore the loss of confidentiality of all exchanges on the connection.

In addition, the generation of cryptographic keys involves generating random numbers and it is essential that these generators are truly random, otherwise the key will be more easily broken. The random number generation functions as proposed in the operating systems do not fully satisfy this condition.

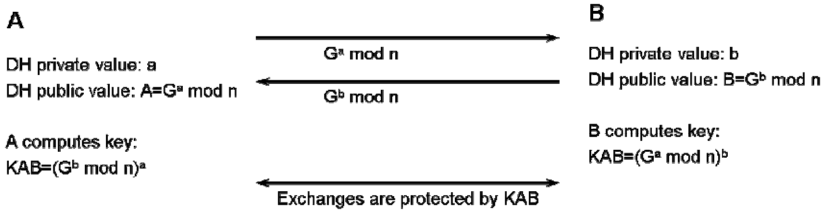


Figure 3.3. *Diffie-Hellman (DH) method to generate cryptographic keys*

The method for generating cryptographic keys can also be based on the Diffie-Hellman (DH) principle where the two communicating parties provide a DH to each other that makes it possible to build a common key. This method is illustrated in Figure 3.3. The entities A and B communicate their DH public value: Ga mod n and Gb mod n (where n and G are publicly known). From these two values, both entities calculate the key KAB = (Ga mod n)b = (Ga*b mod n) = (Gb mod n)a. This resulting key can only be known by A and B because the private values a and b are only known by A and B. It should be noted that the DH method is vulnerable to spoofing attacks because the provided DH public values are not authenticated, so an attacker C could contact B pretending to be A and submit its own public value DH Gc mod n. To counteract such attacks, it is necessary that the DH public values sent on to the network are authenticated. In general, the two entities A and B have a set of public/private keys and use their private key to prove the authenticity of their value. This is called authenticated DH.

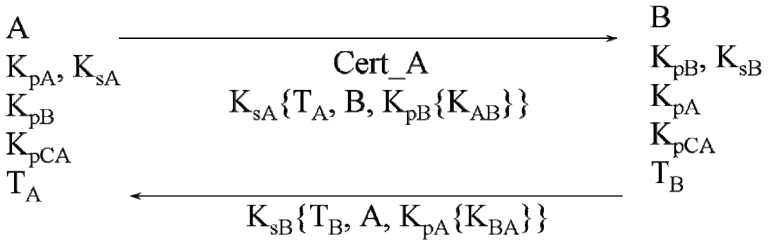


Figure 3.4. *Two-way X.509 protocol*

3.2.7. *Cryptographic protocols*

Cryptographic protocols define the rules and formats (semantics and syntax) to which the messages exchanged between entities must comply in order to meet certain security properties. Among the implemented security services, these protocols implement the peer entity authentication and also the establishment of common cryptographic keys.

For illustration, let us consider the case of the two-way X.509 protocol [X509] based on public key cryptography and let us analyze its properties. As shown in Figure 3.4, entity A (and B for the entity) has a set of keys: public key K_{pA} and private key K_{sA}. A sends in its first message its electronic certificate *Cert_A* and a token encrypted with the private of A that contains the time of A (T_A) when sending the message, the identity of the recipient B, and a key K_{AB} randomly generated by A and encrypted with B's public key. Upon receiving this message, B obtains A's public key K_{pA} from certificate *Cert_A* and it then decrypts the token using A's public key; this guarantees that the token was generated by A. Thanks to the content of the token, B checks it is the recipient of the message, that the request is fresh enough (by controlling the timestamp value), then it decrypts the key by using its private key K_{sB}; note that B is the only entity that can decrypt the key K_{AB} because it is the only one owning the complementary key K_{sB}. Likewise, B builds a message encrypted with its private key K_{sB} and sends it to A. The keys K_{AB} and K_{BA} being exchanged between the two entities aim to protect unidirectional communications between A and B.

The security properties made by the two-way X.509 protocol include:

– freshness of the exchange: the timestamps positioned by A and B during exchanges ensure that these messages are "fresh", i.e., they were not copied and pasted by spies;

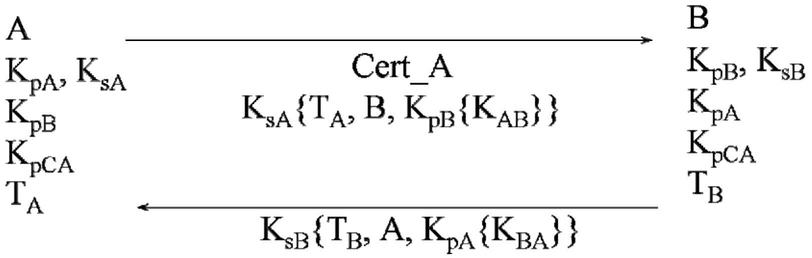


Figure 3.5. *Three-way X.509 protocol*

– confidentiality of exchanged symmetric keys: the content of a token can be easily deciphered by anyone with mere knowledge of the public key of the sender; however, the encryption of keys K_{AB} and K_{BA} with the recipient's public key guarantees the confidentiality of the key. Indeed, the complementary private key that makes it possible to decrypt this information must remain known by its owner only;

– data integrity: this service is indirectly implemented in the protocol; the accidental or malicious modification of the token will result in far-fetched values inside and the receiver could verify he is not the true receiver and/or the timestamp is not correct;

– origin authentication of X.509 messages: the messages do not include any electronic signatures; this service is ensured indirectly as the sender is encrypting the token with its private key.

This two-way protocol suffers from a major disadvantage: it requires good synchronization of clocks of both entities. Too big a difference between clocks leads to the rejection of the token and the inability to establish communication between entities. A three-way X.509 protocol can solve this problem. Instead of using timestamps as a proof of freshness, the messages include nonces N_A and N_B, as shown in Figure 3.5. A nonce is a random number that can either be called random, random number or challenge. Entity A generates N_A for this phase of authentication with B; as such, A is ensured to communicate with B because the token sent back by B contains the same value N_A and it is encrypted with the private key of B.

To verify the security properties offered by a cryptographic protocol, several formal validation tools exist, like AVISPA (Automated Validation of Internet Security Protocols and Applications) [AVISPA]. This can also test the robustness of these protocols to man-in-the-middle attacks which consist of having an attacker on the path of communication between two entities A and B. If the attacker succeeds in making A think it is B and B think it is A, then the cryptographic protocol is said not to be robust to man-in-the-middle attacks.

3.3. Secure communication protocols and VPN implementation

Several security protocols are defined to protect communications going through a network. Generally, these protocols are based on two successive phases, namely:

– the initialization phase where two entities mainly authenticate and negotiate services and security mechanisms in order to protect their data, and agree on one or more symmetric key(s). Peer authentication during this first phase uses a cryptographic protocol (see section 3.2.7);

– the data protection phase: the services and security mechanisms and symmetric keys that were previously agreed on during the initialization are activated to protect data exchanges.

In this section, two popular security protocols, IPsec and SSL, are presented. For each of them, the two phases of operation are presented with the associated protocols, the security services supported to protect the exchanges during the two phases and the processing done over data for their protection. A comparison of these two protocols and the possible usage of them in a VPN tunnel protection context is described.

3.3.1. *Secure Socket Layer (SSL) and Transport Layer Security (TLS)*

The original SSL [RES 01] was designed by Netscape Communications to protect e-commerce applications based on HTTP and became very popular because of its systematic integration in Microsoft browsers Internet Explorer and Netscape Navigator. Becoming the de facto standard, version 3.0 of this protocol was standardized in 1999 by the IETF [RFC2246] and was renamed TLS. In fact, the IETF has made minimal changes to SSL and it is also called version 1.0 of TLS so it is version 3.1 of SSL. Thereafter, the explanations focus on SSL and the few differences between SSL and TLS will be provided in section 3.3.1.5.

SSL is in the form of an additional protocol layer between the application and transport layers, as shown in Figure 3.6. Thanks to its position in the protocol stack, SSL can support protection for any TCP-based applications like ftp, telnet, smtp, pop3, etc. To enable SSL security, applications are needed to call a secure socket connection setup instead of a standard socket setup. To distinguish between a non-secure TCP application and a TCP application protected by SSL, a convention is to add an “s” at the end of the protocol and assign a new port number, as shown in Table 3.1. Thus, the HTTP application that classically listens on port number 80 becomes an HTTPS application and should listen on port number 443, according to the convention.

Note that SSL works as a client-server model and is therefore not completely symmetric between the two entities.

Applications	Port numbers
https	443
telnets	992
ftps	990
ftps-data	989

Table 3.1. *Port numbers conventionally associated with applications protected by SSL*

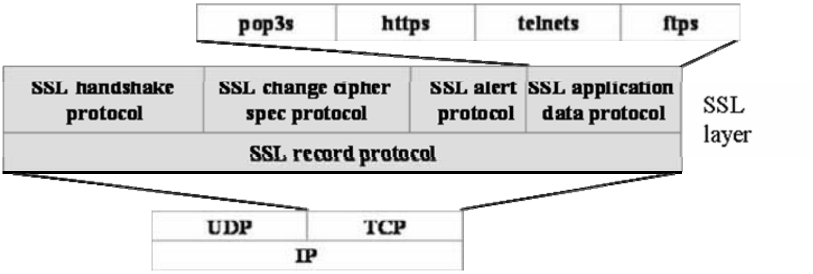


Figure 3.6. *Sublayer organization of SSL*

This section does not address all the details of SSL and TLS. Further information is available in [RES 01].

3.3.1.1. *Security services*

SSL supports a set of security services according to the protocol phase.

For the initialization phase, the offered security services are as follows:

– SSL server authentication to the SSL client. An SSL server must own an SSL certificate and private key; its electronic certificate serves to authenticate itself and must be sent to the SSL client.

– Optional authentication of the client to the server. An SSL client who can take the form of a browser is not required to own an electronic certificate to connect to an SSL server; in most cases on the Internet, the client is not requested to authenticate; according to the local security policy, the SSL server will decide whether an unauthenticated client is permitted to access to the server.

– Replay detection. We must avoid a malicious user having previously spied exchanges over the network to replay the exchanges of the initialization phase and make the client think it is the server (or the server think it is the client, if the authentication of the client is mandatory).

– Negotiation of the services and security mechanisms.

– Establishment of a symmetric key (master key).

– Protection of the initialization phase messages: integrity and authentication of data origin.

For the data protection phase, data produced by the TCP applications need the following protection (see Figure 3.6):

– privacy;

– data integrity;

– data origin authentication;

– replay detection. It is important that the client and the server can detect possible replays to be sure that data protected by SSL cannot be injected several times in an existing SSL session and be treated several times by the recipient.

The details of each of these phases are given in sections 3.3.1.3 and 3.3.1.4. It is important to note at this point that SSL security relies strongly on the authentication of the server. Indeed, it is useless for clients to protect data with the most sophisticated cryptographic algorithms if they are not sure of the server's identity. As explained in section 3.2.5, the authentication of the server is based on public key cryptography and raises the problems that the client must have confidence in the CA issuing the server's certificate. In particular, too weak a management of trust will result in making it easier to implement man-in-the-middle attacks.

3.3.1.2. *SSL organization into sublayers*

SSL is organized into two main sublayers, as shown in Figure 3.6:

– The *SSL record protocol* sublayer implements all the security services to protect the data coming from applications and also certain control messages. The details of this sublayer are presented in section 3.3.1.4.

– The upper-level sublayer is mainly used to establish and manage SSL sessions by implementing the initial authentication, negotiation of security parameters, processing of errors, etc. It comprises four modules:

- the *SSL handshake protocol* implements the exchanges of the initial phase between the client and server. This protocol module is presented in section 3.3.1.3;

- the *SSL alert protocol* sends and manages messages alerts, especially messages for closing an SSL session;

- the *SSL change cipher-spec protocol* occurs once the initialization phase is completed to enable the data protection phase. The client and the server should notify each other at what exact moment their transmitted messages are protected with the security context previously negotiated by the *SSL handshake protocol* module;

- the *SSL application data protocol* communicates data coming from the TCP applications to the sublayer *SSL record protocol* for protection.

3.3.1.3. *Initialization phase of SSL*

The initialization phase is implemented by the *SSL handshake protocol* module. The earliest exchanges between the client and server serve to agree on the SSL version number for next exchanges. Versions 3.0 and 3.1 are commonly used today on the Internet (version 2.0 more rarely).

The exchanges during this initialization phase depend on the key exchange method in use and whether the client authenticates or not. The exchanges described in Figure 3.7 correspond to a simple method for key exchange based on RSA encryption and no client authentication. Other methods are mainly based on DH.

Figure 3.7 gives the functional messages that are exchanged between the SSL client and server through their *SSL handshake protocol* module. Several types of messages are defined in the *SSL handshake protocol* for such exchanges like: ClientHello, ServerHello, Certificate, ServerHelloDone, ClientKeyExchange and Finished.

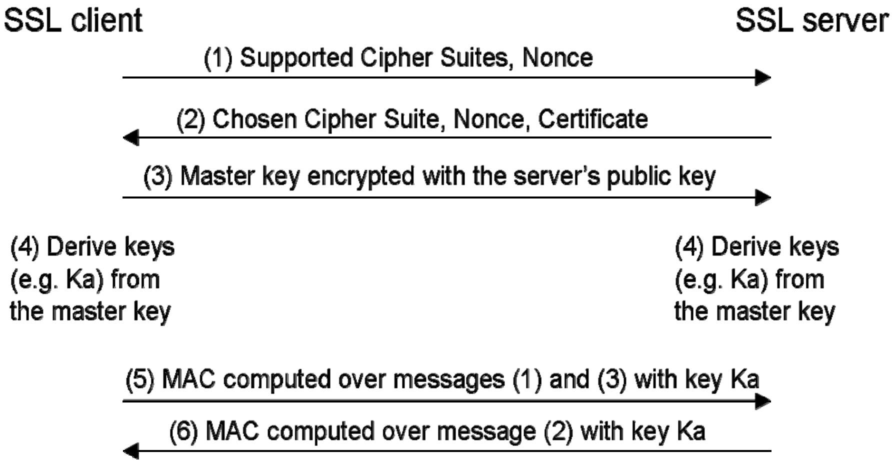

SSL client
SSL server
(1) Supported Cipher Suites, Nonce
(2) Chosen Cipher Suite, Nonce, Certificate
(3) Master key encrypted with the server's public key
(4) Derive keys (e.g. Ka) from the master key
(4) Derive keys (e.g. Ka) from the master key
(5) MAC computed over messages (1) and (3) with key Ka
(6) MAC computed over message (2) with key Ka

Figure 3.7. *SSL handshake protocol exchanges*

The first message (1) allows the client to initiate an SSL session by presenting the supported cipher suites, and a random number. A cipher suite defines the set of parameters that need to be negotiated between the client and server to ensure data protection, i.e. the compression method, the data encryption algorithm and the hash function used for generating a MAC.

The server then selects a cipher suite among those presented in message (1) and returns its selection to the client in message (2) along with its certificate, and the nonce that was previously received from the client. The client can then verify the authenticity of the server's public key. It then generates a master key that it encrypts with the public key of the server and it sends it in message (3). The server obtains the master key by decrypting the information with its private key. The client and server are then able to derive symmetric keys (i.e. Ka) from this master key on their own (step (4)).

We can easily notice that no security services have been implemented so far until step (4). Neither authentication of the server or integrity of the *SSL handshake protocol* messages (1), (2) and (3) have been made. Note that the entire security of this initialization phase is guaranteed by messages (5) and (6). The client sends message (5) which carries the MAC calculated over messages (1) and (3) with the key Ka. As such, the server can verify the integrity of messages (1) and (3). Likewise the server sends a MAC that serves to protect its message (2) and the client is then able to check several security properties. First, if the MAC is the one expected, the client has the assurance that message (2) remains as it was sent. Second, the client is able to authenticate the server if the following condition applies. If the received MAC is correct, it means for the client that Ka is known by the server, so the server was able to decipher the master key, and finally the server owns the private key, thus proving its identity.

The client is also protected against replays as its randomly generated nonce guarantees that the server's answer was generated directly to its request, and the nonce is protected by the MAC of message (6), thus proving its origin from the server.

After step (4), we can notice that the client and the server share the same security context, i.e. the cipher suite and some symmetric keys. They only have to activate that security context to protect subsequent exchanges. The SSL cipher-spec protocol activates the newly negotiated context by sending a control message. The client can activate the security environment at any time after sending message (3). The server can activate the context only after receipt of message (3). If the activation context is made by the client immediately after sending message (3), this means that any subsequent messages of the SSL handshake protocol module, especially message (5), will benefit from this protection. As such, the negotiated context ensures not only data protection, but also protection of control messages.

3.3.1.4. *Data protection phase*

Once the context is activated by the *SSL cipher-spec change protocol*, data protection can be ensured by the *SSL record protocol* sublayer. These data issued by the applications and forwarded by the *SSL application data protocol* sublayer are first fragmented by the sublayer *SSL record protocol*, as shown in Figure 3.8. Then they are compressed, integrity is protected by injecting a MAC and they are encrypted. The type of encryption and MAC depends on the prior negotiation conducted during the initialization. MAC generation may be of several kinds, based on MD5, SHA-1 or SHA-256. The encryption algorithms can include other RC4, 3DES, AES, etc.

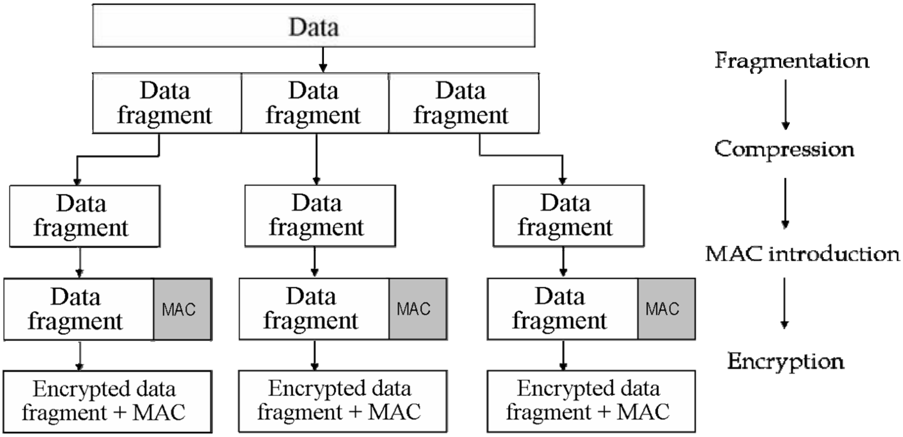


Figure 3.8. *Operations performed by the SSL record protocol sublayer*

3.3.1.5. *Differences between SSL and TLS*

At the time of standardizing the SSL protocol, the IETF wished to inject improvements gained from its expertise on security protocols acquired during IPsec standardization. As such, TLS retains the same organization as SSL software with the same sublayers and modules. The main differences between SSL and TLS are the mechanisms for key exchange and the construction of the MAC which the IETF changed into HMAC which was developed for IPsec (see section 3.3.2.2).

3.3.2. *IPsec protocol suite*

The IPsec protocol suite (IP security) was standardized by the IETF, resulting in several RFC (Request For Comment) series. In 1995, there were some IPsec products but they were not bought by industrials at that time. The IPsec market became increasingly larger after that time (after 1998). In 1998, the next series of RFC were strongly improved and were widely implemented in firewalls and other VPN equipments/software. The latest series of 2005 introduced only a few improvements, thus avoiding the problems of compatibility between generations of IPsec equipments/software.

As shown in Figure 3.9, the IPsec protocol suite includes several protocols operating at various levels of the protocol stack:

– Protocols protecting IP packets. The two sub-protocols AH (Authentication Header) and ESP (Encapsulating Security Payload) make it possible to encrypt the contents of packets and/or to append a MAC. These sub-protocols are part of the IP layer and are presented in section 3.3.2.2.

– Protocol implementing the initialization phase. The IKE (Internet Key Exchange) protocol provides this role and takes the form of an application level module running over UDP port 500. It also negotiates all the security settings appropriate for the protection of data flows; this set of parameters is known as the “security association”. This protocol is described in section 3.3.2.3.

Two types of protection are possible with IPsec. The tunnel mode makes it possible to define and manage an IPsec tunnel, while the transport mode provides direct protection of IP packets without additional encapsulation. The tunnel mode is mainly used in a VPN connecting two remote private networks (see section 3.3.4.1). The transport mode is used to protect a simple connection, i.e. all the exchanges realized between the two end devices of a connection. The transport mode is used in the case of IP mobility (see Chapter 12) and in the case of mobile VPN (L2TP/IPsec solution, see section 3.3.4.1).

IPsec works today in a point-to-point mode, but according to RFCs, it could be used in a point-to-multipoint mode. However, as a large number of issues are not yet resolved with the point to multipoint mode, like the group key management, the IPsec suite is only used today as a point-to-point mode and only this latter mode is described in this section.

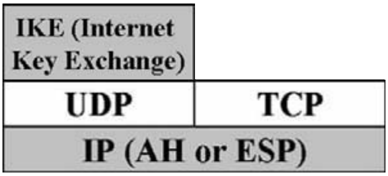


Figure 3.9. *Organization of the IPsec suite into layers*

3.3.2.1. *Security services*

For the initialization phase, the IKE protocol is responsible for implementing the following security services and functions:

– Mandatory mutual authentication between two IPsec entities. Note that no IPsec system is predominant over the other one in an IPsec exchange. Therefore, there is a certain symmetry between the two entities, and both are required to authenticate each other. Several methods for authentication are available, like electronic certificates or pre-shared keys.

– Integrity and origin authentication of the messages within the initialization phase.

– Replay detection.

– Negotiation of security services and mechanisms (security associations).

– Symmetric key establishment.

For data protection provided by sub-AH or ESP protocols, the following security services are supported:

– data confidentiality;

– data integrity;

– data origin authentication;

– replay detection.

3.3.2.2. *AH and ESP sub-protocols*

The AH (Authentication Header) and ESP (Encapsulating Security Payload) sub-protocols support data protection, each providing the security services listed in Table 3.2.

Sub-protocol	Security services and coverage of the protection
AH	Integrity and data origin authentication and optional replay detection Protection covering the content of the packet and part of the IP header
ESP	Data confidentiality (optional) Integrity, data origin authentication and replay detection (optional) Protection over the content of the packet only At least one security service activated

Table 3.2. *List of security services provided by AH and ESP with their features*

It may be noted that for ESP, all the security services appear as optional, but at least one of them must be activated. Also note that the services implemented by AH (integrity/origin authentication) are also implemented by ESP with the same MAC mechanism. As such, it is legitimate to wonder whether AH has a useful role in IPsec, but we should see that the protection coverage of AH and ESP is different. For ESP, protection coverage is limited to the packet content, while AH protection is done over the packet content and part of the IP header.

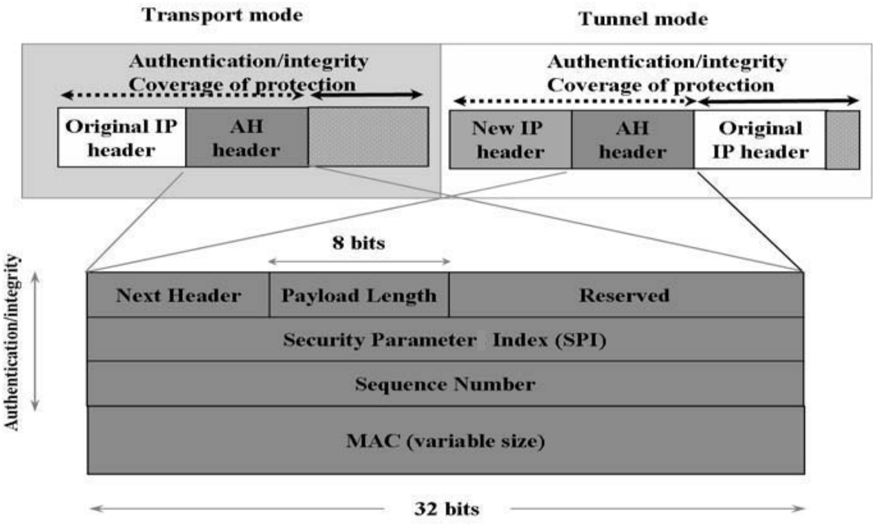


Figure 3.10. *Format of AH header*

The formats of the headers appended by AH or ESP are given in Figures 3.10 and 3.11. They show the coverage of protection over the packet for each tunnel and transport mode. Each line of the headers is 4 bytes long. The AH header [RFC2402] is mainly used to transport MAC covering the full content of packet and part of the IP header and the AH header. Partial protection means that all fields in the IP header except those being modified during transfer over the network (e.g. TTL fields) are under the protection of the MAC. The AH header also serves to transport a 4-byte sequence number for detecting packet replays, a 4-byte SPI to help identify the security association used for protection, and a *Next header* field used to identify the protocol that follows the AH header (i.e. "TCP" or "UDP" for the transport mode, or "IPv4"/"IPv6" for the tunnel mode). The replay detection done at the IPsec receiver side consists of verifying that the sequence numbers included in each received packet are correctly incremented from one packet to another and also that the MAC that covers the sequence number is correct.

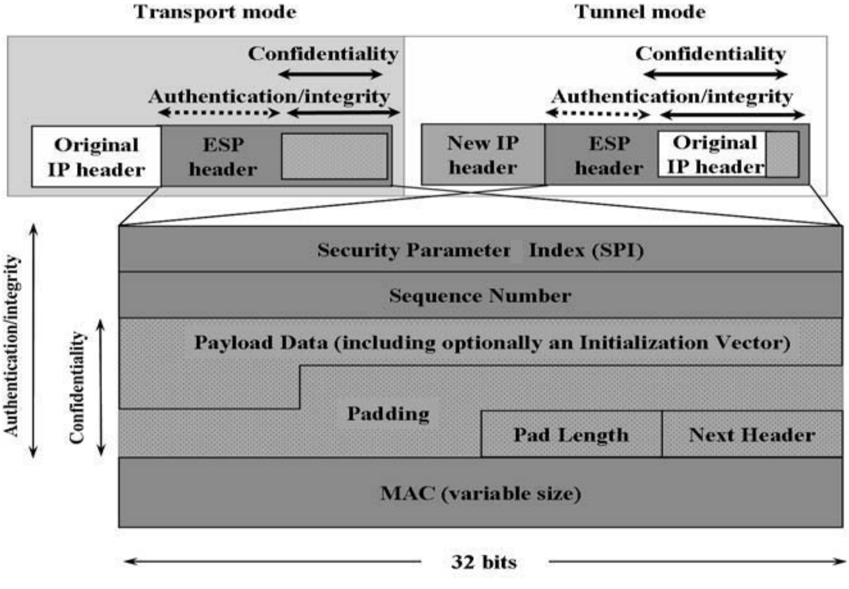


Figure 3.11. *Format of ESP header*

Like the AH, the ESP header [RFC2406] includes, for the same reasons, the SPI fields, a sequence number and MAC, as shown in Figure 3.11. In addition, ESP can encrypt the contents of an IP packet, the payload of the IP packet in the case of transport mode, or the packet encapsulated in the case of tunnel mode. The shaded part in Figure 3.11 represents all the data encrypted within ESP. It contains padding that helps to align data before encryption by injecting no meaningful information.

The alignment depends on the encryption algorithm (see section 3.2.2.1). The padding length enables the receiving IPsec system to distinguish useless data (padding) from useful data (payload data). To enable the receiver to interpret these encrypted data correctly, the next header field specifies the type of the first data being encrypted; traditionally it is called TCP or UDP in the case of transport mode, or IPv4 and IPv6 in the case of tunnel mode.

The most recent changes to IPsec cover the possibility of increasing the size of the sequence number without modifying the format of the ESP and AH headers [RFC4302, RFC4303]. The idea is to remember from both sides the 8-byte sequence number and to communicate in each packet the 4 low-weight bytes of the sequence number. Having a larger sequence number makes it possible to renew the security associations less often since a security association is supposed to be renewed each time the sequence number is performing a cycle. Another important change is to separate the RFCs defining the AH and ESP header format from those specifying algorithms/functions relevant to their construction. This separation is important because the formats are not changed frequently, while the list of algorithms/mandatory functions can change relatively frequently at the discretion of vulnerabilities discovered on some of them or the definition of new algorithms. The last RFC giving the algorithms/functions for ESP and AH is RFC 4305 [RFC4305]. A summary of current mechanisms is given in Table 3.3. Each of these algorithms/functions is described in a specific RFC. Note that the NULL algorithm matches the lack of encryption and disables encryption in ESP. The HMAC-SHA-1-96 function serves to generate a MAC using SHA-1; the MAC is computed not only over the IP packet, but over the packet concatenated with a symmetric key and only the first 96 bits are put into the MAC of the header. More recently, the IETF identified some cipher suites VPN-A and VPN-B [RFC4308].

Encryption algorithms (for ESP)	NULL, Triple DES-CBC [RFC2451] Non-mandatory algorithms: AES-CBC-128
MAC generation functions (for ESP and AH)	Mandatory functions: HMAC-SHA-1-96 [RFC2404] Optional functions: HMAC-MD5-96 [RFC2403]

Table 3.3. *Mechanisms selection for AH and ESP*

3.3.2.3. *IKE protocol*

The IKE protocol controls and establishes the security association for AH or ESP. This application level protocol operates on UDP port number 500. Two versions of the IKE protocol have been successively defined. The first one, known as IKEv1, has the objective of being generic, i.e. to produce a security association that is applicable to any security protocol. As a result, IKEv1 is described with four large RFCs (the main one is [RFC2409]), with occasional inconsistencies between RFCs.

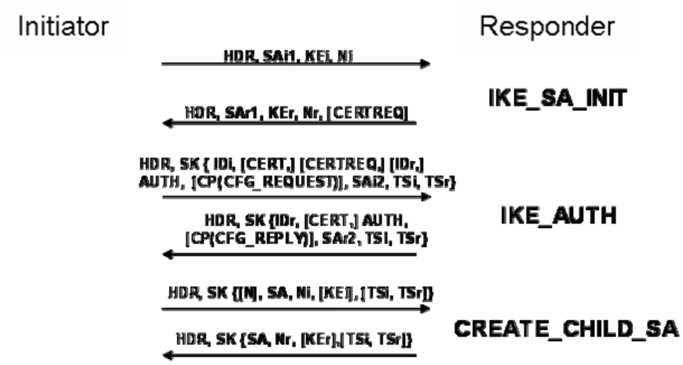

Initiator
Responder
HDR, SAi1, KEi, Ni
IKE_SA_INIT
HDR, SAr1, KEr, Nr, [CERTREQ]
HDR, SK { IDi, [CERT,] [CERTREQ,] [IDr,]
AUTH, [CP(CFG_REQUEST)], SAi2, TSi, TSr}
IKE_AUTH
HDR, SK {IDr, [CERT,] AUTH,
[CP(CFG_REPLY)], SAr2, TSI, TSr}
HDR, SK {[N], SA, Ni, [KEi],[TSi, TSr]}
CREATE_CHILD_SA
HDR, SK {SA, Nr, [KEr],[TSi, TSr]}

Figure 3.12. *IKEv2 exchanges*

To remove this high level of complexity, the second version of IKE deleted the generic nature of IKEv1 and requires a single RFC [RFC4306] only. The result is greater clarity and simplicity. IKEv2 defines six messages instead of eight. IKE protocols distinguish two levels of security associations, one to protect IKEv2 messages (IKE_SA) and another one named CHILD_SA (or SA_IPsec) that ensures IP level protection by the AH or ESP headers.

The six IKEv2 messages are shown in Figure 3.12 and distinguish the roles of initiator and responder. The initiator is the IPsec system that initiates the IKEv2 exchange. The first four messages IKE_SA_INIT and IKE_AUTH serve to initiate a secure connection and the last two CREATE_CHILD_SA to renew or establish other CHILD_SA security associations:

– IKE_SA_INIT: the first two messages make it possible to negotiate a IKEv2 security association IKE_SA (SAi1 and SAr1 information), exchange random numbers (Ni and Nr) and perform a DH exchange (KEi and KEr) to agree on a common symmetric key. In the first message, the initiator proposes a set of cryptographic algorithms (SAi1) and the responder returns its choice to the initiator (SAr1). Many IPsec keys are derived from the shared key obtained from DH, and especially the keys useful to protect IKEv2 messages. The obtained IKE_SA security association helps to protect all the following IKEv2 exchanges, especially the IKE_AUTH messages. Note that these two first messages are not protected and do not allow both parties to identify and authenticate.

– IKE_AUTH: these two messages allow the originator and the responder to identify themselves (IDi and IDR), authenticate each other and to prove the origin and integrity of IKE_SA_INIT messages. To do this, the AUTH element is calculated on IKE_SA_INIT messages with the private key of entities (if certificates are exchanged between initiator and responder) or an encryption key derived from the IKE_SA_INIT exchanges. The detection of replays of messages IKE_SA_INIT is also guaranteed as the AUTH element is also computed over the random numbers Ni and Nr. All the IKE_AUTH messages are encrypted using the key derived from IKE_SA_INIT, which guarantees the confidentiality of the communicating parties. IKE_AUTH exchanges enable establishing a security association by applying the same treatment as in IKE_SA_INIT (SAi2 and SAr2). The TSi and TSr elements identify IP flows that will benefit from this CHILD_SA protection, but all IP flows are not systematically protected by the CHILD_SA association.

– CREATE_CHILD_SA: these two messages are used to update an existing CHILD_SA security association or to establish a new CHILD_SA association for protecting another IP flow type (TSi and TSr). An update may be carried out at regular time intervals or after a certain amount of IP packets have been protected with the same security association. The establishment of a new CHILD_SA between the same two parties benefits from the previously established IKE_SA association.

IKEv2 in its basic version does not include support of client's mobility and the possibility that the mobile client can change its IP address. Indeed, the addresses of both parties identify an IKE_SA and/or CHILD_SA security association and in the event of an address change, the security association is lost. The MOBIKE protocol [RFC4555] was defined as an extension of IKEv2 to attach a security association to some new IP addresses. This change of address may occur when the initiator is moving, or when the poor connectivity between the two parties makes one party decide to connect to the other (multi-homed) party through one of its secondary addresses.

3.3.3. *Comparison between SSL and IPsec security protocols*

Due to their positioning within the protocol stack, SSL and IPsec address different usages. IPsec with AH and ESP sub-protocol is well designed to protect communications over sections of the network between two pieces of network equipment. By their nature, pieces of network equipment which are already operating processing IP packets like IP routing can also support IPsec functions. Administration of IPsec is pretty complex because of the distinction made between the IKE module for SA management and the IP flow protection (AH/ESP) module and the greater complexity of setting IKE configuration (compared to SSL). Therefore, it is better to avoid configuring IPsec at each terminal of a private network. It is preferable to centralize the IPsec functions in only one piece of equipment so the terminals of the private network will benefit from the protection ensured by this equipment at the front (see section 3.3.4.1).

The traditional network equipment (e.g. routers) does not support the socket level in data processing. It is therefore inappropriate and inefficient to secure sections of network using SSL. Thus, SSL is useful to implement end-to-end security (e.g. from one terminal to another), especially as SSL is easier to maintain and configure than IPsec.

3.3.4. *IPsec VPN and SSL VPN*

VPNs [GUP 02] were originally designed to facilitate communications of companies, in particular to allow remote sites within one company to communicate as if they were part of the same private network with access to the same services. This notion of VPN was then extended to nomads to allow nomads to connect remotely to a private network and receive the same services and resources as local terminals. VPNs are easily implemented by tunneling techniques, i.e. encapsulation of traffic. This encapsulation technique is illustrated in Figure 3.13; the terminal sends an unprotected packet addressed to terminal b that is located on a remote private network; gateway A implementing the tunneling function encapsulates the packet into a tunnel being defined by its end points A and B; gateway B located in the remote network decapsulates the packet and forwards the packet to destination b. Note that the VPN notion is not necessarily secure.

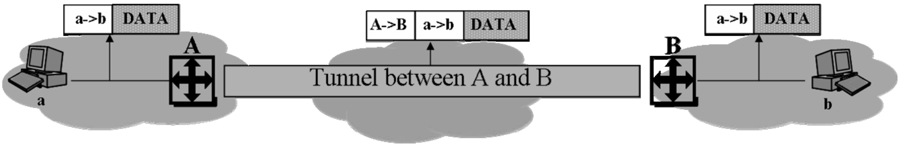


Figure 3.13. *Simple tunneling between two pieces of equipment A and B (ends of tunnel)*

There are two categories of VPN to interconnect remote private networks as the setup requires an operator or not. A company can decide to set up its VPN on its own. The company first obtains IPsec equipment. This equipment is usually known as IPsec gateways, and may take the form of firewalls. The company needs to configure this equipment, to subscribe to an Internet access offered by any ISP, and the VPN is then operational, but no QoS between remote interconnected networks is guaranteed. In case of bursts on the public network, the traffic is likely to suffer from disruptions. The second solution is to offer the VPN services of an operator. The operator will then configure the equipments managing tunnels and guarantee a certain QoS in the interconnection: tunnels put in place by the operator are usually MPLS (Multi Protocol Label Switching), ATM (Asynchronous Transfer Mode) or FR (Frame Relay) and are rarely secure with cryptography tools. However, in their commercial announcements, the operators are claiming that such tunnels are secure as the flows are said to remain separated during transfer, so there are no risks that flows arrive at the wrong destination. Let us note that it is very difficult but still possible to spy on these communications.

The rest of this section presents IPsec VPN and SSL VPN in two scenarios: interconnection of two remote sites and remote access of a nomad to its corporate network.

3.3.4.1. *IPsec VPN*

IPsec VPNs are the first secure VPNs that have been widely used by companies to securely interconnect their remote networks. IPsec tunnels were configured with security associations of IPsec in tunnel mode. Afterwards, IPsec adapted to a nomadic scenario, but at a very high cost because each manufacturer was initially interested in providing its proprietary IPsec solution requiring a license for each nomad. This IPsec solution for the nomads came to compete with Microsoft solution PPTP (Point-to-Point Tunneling Protocol) which allows secure remote access but with less security robustness.

In 2001, the standardized solution L2TP/IPsec (L2TP over IPsec) [RFC3193] for secure remote access has emerged through the combined efforts of Microsoft and Cisco and has supplanted proprietary solutions based on IPsec and PPTP. The solution is to protect a L2TP (Layer 2 Tunneling Protocol) tunnel by the IPsec protocol suite. More precisely, a security association is first established in the transport mode between the nomad and IPsec gateway. During the establishment of this association, the nomad (hardware equipment) must authenticate itself. The resulted security association protects the exchange of traffic between the nomad and the gateway. Then, an L2TP tunnel is built to enable the nomad to communicate with terminals located behind the gateway. The establishment of the L2TP tunnel fulfills two interesting functions: it serves to authenticate the user and to assign a private address to the nomad which is then able to connect to its remote network and gain access to the services as if it was connected locally. However, L2TP does not support any security services, so all the security of the connection is ensured by IPsec. The protocol layers of the L2TP/IPsec solution are defined in Figure 3.14. The PPP layer is encapsulated in L2TP and allows the user to authenticate itself using the usual PPP authentication methods.

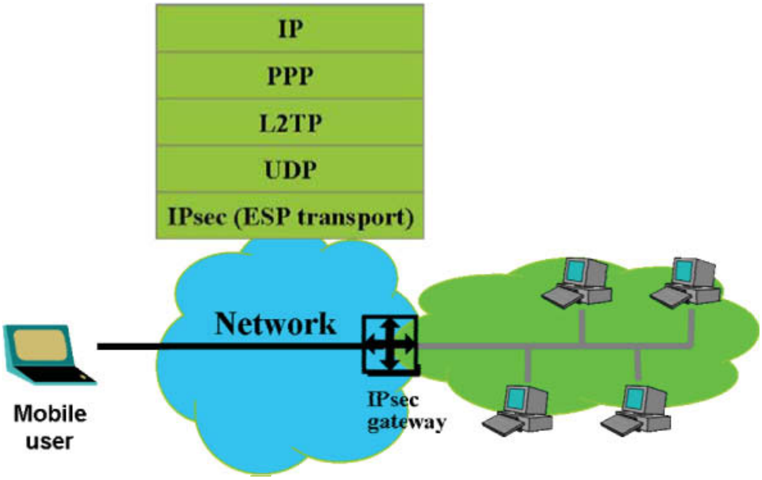


Figure 3.14. *Protocol stack for L2TP/IPsec*

3.3.4.2. *SSL VPN*

SSL VPN solutions were first widely marketed in the USA in the late 1990s. Around 2003, they arrived in Europe and successfully became the number one VPN product on the remote access market, far ahead of IPsec VPN solutions. SSL VPNs offer attractive costs either for purchase or maintenance. They make it necessary to install an SSL VPN gateway that will secure remote access and which is usually located in the DMZ of the company, as shown in Figure 3.15. The solutions fall into two categories [FRA 08]:

– Clientless SSL VPN: this solution only requires a simple browser on the client side. No license is needed. Most of clientless solutions assume that the browser communicates with the classical SSL VPN gateway using HTTP language (with SSL protection enabled). The gateway suggests a menu browser making it possible for the user to gain access to a greater or lesser number of applications. Then, the gateway has the responsibility to ask servers of the private network (SMTP, FTP, NFS, etc.) about the requested information, and then has to translate the answer into HTTP format in order for the client to visualize the answer on his browser. Applications that can be accessible are applications that can be easily “webized” (i.e. whose flows are in HTTP format).

– Non-clientless SSL VPN: this solution may take the form of a light client, for example, a Java applet, or a heavy client requiring a specific client software (with license). The goal of non-clientless SSL VPN is to realize the encapsulation of all exchanges into an SSL tunnel between the client and the SSL VPN gateway.

Therefore, this solution is pretty close to the L2TP/IPsec solution since a tunnel is built and it is therefore possible to gain access to any piece of equipment of the private network. Note that, unlike clientless SSL VPN, the client here has a private address that makes him visible in the private network.

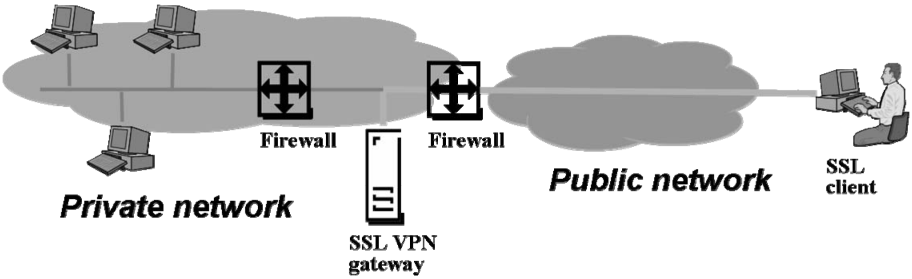


Figure 3.15. *SSL VPN architecture*

Clientless SSL VPNs are suitable for companies that limit remote access to a few applications like simple e-mail. Non-clientless SSL VPNs are more appropriate to companies that want more flexibility and for instance want to allow a development team to launch a demonstration remotely or to perform remote testing or developments.

3.4. Authentication

With the fundamental nature of the authentication service on networks, authentication techniques are well developed. The definition of the EAP (Extensible Authentication Protocol) described in this section greatly helped to diversify authentication methods. The strong need for security added to a much simple management of the security led to the definition of centralized authentication in certain pieces of equipment such as servers RADIUS, LDAP, etc. AAA protocols emerged for these reasons with the possibility of extending authentication to inter-domain authentication, i.e. to enable subscribers to successfully authenticate to an administrative domain different from their own domain of subscription. This section mainly focuses on describing various protocols and methods for authentication in order to control access to a network.

3.4.1. *Authentication mechanisms*

Authentication mechanisms are becoming more and more sophisticated on the information system security market. They are designed to provide users and administrators with a certain ease of use (rapidity of authentication, simplicity of use), a minimal administration, great robustness (against possible intrusions), high reliability (to avoid authentication errors) and ubiquity of its usage.

These mechanisms, described in more detail in the following sections, can be divided into several categories according to:

– what the entity knows, a password for example;

– what the entity owns, such as a smart card, a private key or a Kerberos ticket;

– what the entity is: this category covers the authentication techniques based on a user biometric features (fingerprint, iris, facial form, shape of hands, etc.);

– what the entity is known to do: by demonstrating its ability to reproduce the same action like a written signature, an entity can authenticate.

Usually, the technical authentication solutions distinguish between a weak authentication and a strong authentication. For a weak authentication, an entity is authenticated with only one piece of authentication (e.g. password). Strong authentication plans to combine at least two elements of authentication, typically a password and a smart card.

In order to diversify the authentication methods, the IETF has standardized a generic authentication protocol called EAP for [RFC3748]. This protocol is generic, in that it is independent of the authentication method. As shown in Figure 3.16, its role is limited to the transportation of authentication data between a client and a server. The content of these exchanges is not interpreted by the software layer EAP, but by the selected EAP method. As such, it brings the advantage that an EAP method suddenly detected as vulnerable can easily be changed to another more robust method while keeping the same EAP protocol. This makes the security equipment more flexible and able to evolve at low cost.

The EAP protocol is mainly operated in PPP or 802.11 (wireless) environments. Because of its limited role in encapsulation of authentication data, it is extremely simple and includes only four types of messages request, response, success and failure. Today, there are more than 40 EAP methods, but few of them are standardized like EAP-TLS, EAP-MD5 or EAP-SIM.

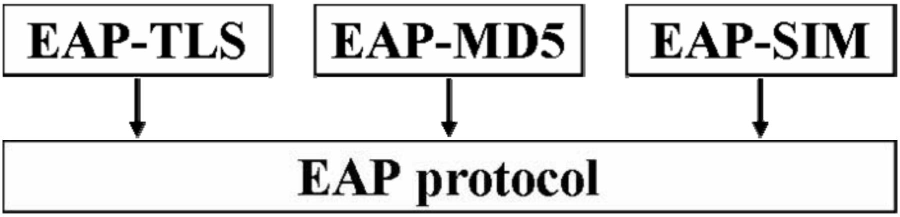


Figure 3.16. *Distinction between the EAP protocol and EAP methods*

3.4.1.1. *Password-based authentication*

Passwords might be static or dynamic. Static passwords obey security policy in a company that can define a minimum number of characters and a lifetime (expressed in days or number of connections). These passwords can be cracked (i.e. discovered by a malicious person) or spied on (on a phone link, data network, etc.) and can lead to the disclosure of confidential information by an intrusive access to a computer account, for example.

To overcome these drawbacks, dynamic passwords, also called OTP (One-Time-Password), have been defined. At each new session, a different value of password must be provided. OTP techniques obviously require a perfect synchronization between the client wishing to authenticate and the authentication server. This synchronization of dynamic passwords can be based on a clock, a series of numbers, a sequence number, etc.

The PAP (Password Authentication Protocol) and CHAP (Challenge-Handshake Authentication Protocol) [RFC1994] were originally based on static passwords and supported authentication of remote users connected on the telephone network through the PPP (Point-to-Point Protocol) and a modem. The PAP requires the PPP client to send a login and password in clear text over the network. The CHAP is based on a random number provided by the network and the client has to send back a hash calculated over this random number and password.

The original PAP and CHAP used static passwords and were then improved by the use of dynamic passwords. This dynamic password is usually generated by a token owned by the user, i.e. a sort of calculator or electronic badge. Upon entering the PIN code, the token provides a dynamic password as a string. Of course, it is necessary that the token and the authentication server are fully synchronized for the server to successfully check the password. If the synchronization is based on a clock, then the risk is high that temporal drift occurs between clocks of the token and the server. Current techniques therefore cope with this possible temporal drift by making the authentication server adapt to the temporal drift according to the password returned by the token. As such, if the client regularly connects to the server, the password will always be accepted as ranging within the windows of acceptance.

The EAP-MD5 method is the first EAP method to be standardized [RFC3748]. This is an adaptation of PPP CHAP protocol. This method is known to be unidirectional, i.e. to enable authentication of an EAP client to a server, but not the reverse. When static passwords are in use, EAP-MD5 is vulnerable to dictionary and brute force attacks, i.e. attempts to cracking passwords by testing commonly used passwords or all the combinations of passwords one by one. More recently, collision attacks were successful attempted, thus proving that the fundamental irreversibility property of MD5 was not satisfied, i.e. it was possible to find a message for a certain hash value. Thus, this means that in a few hours, it is possible to crack an EAP-MD5 password.

3.4.1.2. *Certificate-based authentication or PKI*

This type of authentication is based on asymmetric cryptography and makes it necessary to manage private keys of the entities and usually managing electronic certificates through PKIs (see section 3.1.5). This type of authentication is largely used in the TLS protocol in the context of e-commerce, but also more recently as part of SSL VPN (see section 3.3.4.2).

Several EAP methods have been derived from the TLS protocol. The EAP-TLS method [RFC2716] standardized by the IETF supports the mutual authentication of client and server by using their private keys and certificates. The difficulty of EAP-TLS is in providing each user with private/public keys and ensuring good maintenance of them, in particular by ensuring that the private key of the client remains confidential.

The TTLS (Tunneled TLS) method [FUN 08] is an extension of EAP-TLS in the sense that first a TLS connection is set up between the client and server with a unidirectional (or bidirectional) TLS authentication of the server. Once the TLS session is set up, TTLS benefits from this secure connection between client and server to authenticate the client using classical authentication mechanisms like PAP, CHAP or the Microsoft version of CHAP (MS-CHAP, MS-CHAPv2) or EAP. Thus, these methods, which are usually sensitive to dictionary attacks, are protected by the TLS encrypted channel (with the inability to spy on the password or hash value).

TTLS makes it possible to use former authentication mechanisms deemed to be vulnerable by ensuring their protection with TLS.

As for TTLS, the PEAP (Protected EAP) method is based on first setting up a TLS tunnel; however, it only allows EAP data to be exchanged into the tunnel. This prohibits using former mechanisms like PAP/CHAP for client authentication.

3.4.1.3. *Kerberos ticket-based authentication*

The Kerberos key management system was defined by MIT in 1983 in order to support users' authentication to an application and establish a secure channel with this application. Kerberosis built on two ticket-based servers, as shown in Figure 3.17:

– the KDS (Key Distribution Server) is responsible for issuing a ticket to the client. The ticket enables the client to contact the TGS server securely by certifying the request is authentic and by establishing a session key $K_{C,TGS}$ between the client and the TGS. To do so, the KDS sends a randomly generated key $K_{C,TGS}$ encrypted with the client's public key and communicates the same key to the TGS using the ticket that it is encrypted with the public TGS key. The ticket, which is not readable by the client, is forwarded unchanged by the client to the TGS. The ticket also contains the identifiers of the client and the TGS;

– the TGS (Ticket Granting Server) issues another ticket to the client. The principle of the ticket remains the same as before, i.e. the TGS receives from the client the ticket generated by the KDS and the ID encrypted with the key $K_{C,TGS}$. The TGS decrypts the ticket, deduces the key $K_{C,TGS}$, then decrypts the message built by the client and verifies the consistency between identifiers specified by the client and the KDS in the ticket. In the case of consistency, this means that the request issued by the client is authentic because the client knows the key $K_{C,TGS}$, and to decipher $K_{C,TGS}$ it was necessary to know its private key. The TGS server then proceeds similarly to the KDS generating a key $K_{C,App}$. Likewise, TGS communicates this key to the client and application by sending the client the key $K_{C,App}$ encrypted with $K_{C,TGS}$, and issuing a ticket for the application encrypted with the public key of application.

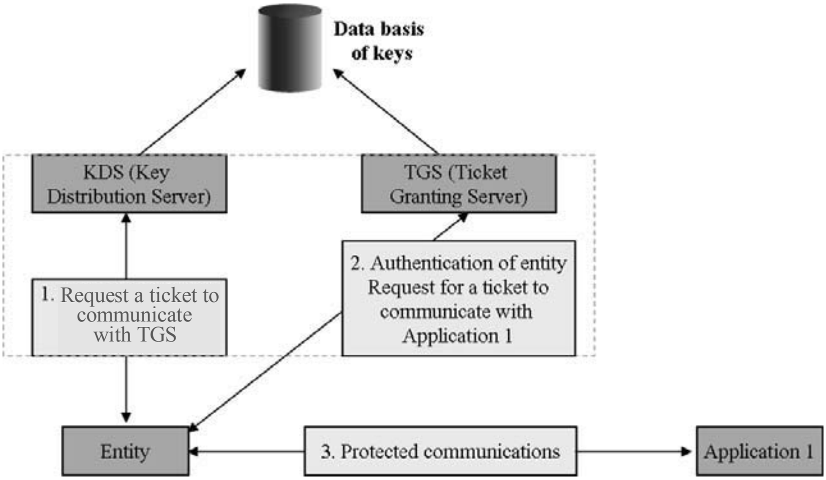

Data basis of keys
KDS (Key Distribution Server)
TGS (Ticket Granting Server)
1. Request a ticket to communicate with TGS
2. Authentication of entity Request for a ticket to communicate with Application 1
Entity
3. Protected communications
Application 1

Figure 3.17. *Kerberos architecture*

This Kerberos architecture, which is useful in establishing a secure channel between a client and an application, is very heavy: it requires implementing two Kerberos servers. For the first access by Kerberos, five exchanges of messages are needed with several encryption/decryptions. When accessing a second application, the procedure is lighter with only three messages between the client and the TGS, and the client and the application.

Some applications suggest controlling the access with a Kerberos ticket. This requires a two Kerberos server architecture to be operational and that these servers are first contacted by the client.

3.4.1.4. *Smart card-based authentication*

This type of authentication is a direct result of authentication performed in the second-generation GSM (Global System for Mobile Communications) cellular network for which subscribers have a smart card in their mobile equipment provided by an operator. The EAP-SIM (Subscriber Identity Module) method [RFC4186] is derived from the GSM authentication with a few improvements like mutual authentication between the network and the user and the establishment of a more robust session key between these two entities.

Another method, EAP-AKA (Authentication and Key Agreement), was defined by 3GPP for the 3G networks UMTS (Universal Mobile Telecommunication

System) and CDMA2000 to authenticate and distribute a session key. It was approved by the IETF as a standard [RFC4187] in 2006. EAP-AKA is based on symmetric keys and typically works within a SIM card.

3.4.1.5. *Biometry authentication*

The authentication of a user can rely on one or more of his biometric features called "biometric modalities". The most common modalities are the fingerprint, iris, face, voice or handwriting signature. More and more commercial products using biometrics mainly to authenticate a user, for instance in order to limit the use of certain equipment (fingerprint readers to unlock a laptop), to control access to sensitive buildings, certain areas of an airport, etc.

To initialize a biometric system with biometric data, first some people need to be enrolled, i.e. sensors digitize their biometric data; then, an algorithmic treatment is made of them and a "template" is stored serving as a reference for future authentication of these persons. This template can be stored in a centralized server or into a smartcard, depending on the use case.

During the authentication procedure, a sensor is again used to digitize a biometric modality. A comparison between these data and the template is made and the system can deduce whether it is the same person or not. Depending on the selected modality, but also the quality of the sensor, results may differ considerably. In particular, errors can occur: a criminal can be accepted mistakenly and then impersonate a legitimate user, or a legitimate user can be denied. These two types of possible errors lead to define two rates for evaluating the reliability of a biometric system: the False Rejection Rate (FRR) (i.e. the rate of rejecting a legitimate user) and False Acceptance Rate (FAR) (i.e. the rate of accepting an impostor). The iris is the most reliable biometric method, but is applicable only to certain very strict applications because of the intrusive feeling that users experience with this method. The fingerprint is unreliable, but is more naturally accepted by users.

To ensure more reliable biometric systems, research is underway on a combination of biometric procedures (e.g. iris and fingerprint). Further work is also underway on the robustness of biometric sensors against impostures like the presentation of an artificial finger, a two-dimensional image of an iris, etc.

Today biometrics are very commonly used to control access to buildings. The user is provided with a smart card where a template of his fingerprint is registered. At the entrance of the building, a smart card reader and fingerprint reader enable the user to enter his smart card and to press his finger on the reader. The system then verifies that the given fingerprint is sufficiently close to the template to unlock the door.

Let us note that biometric systems are not only limited to the authentication procedure. They can be used to identify a person among N entities. This function is useful for identifying a criminal from a list of known criminals in an airport or a football stadium.

3.4.2. *AAA protocols to control access to a private network or an operator's network*

Under deployment of charged network services, the network operator puts in place an architecture known as AAA (Authentication, Authorization, Accounting). Authentication identifies the user requesting access to network services. Authorization limits the user's access to permitted services only. Finally, accounting serves to count the network resources that are consumed by the user.

The AAA architecture makes interactions between three entities as shown in Figure 3.18: the user terminal, the AAA client generally installed at the access router of the operator and the AAA server installed in the operator's network.

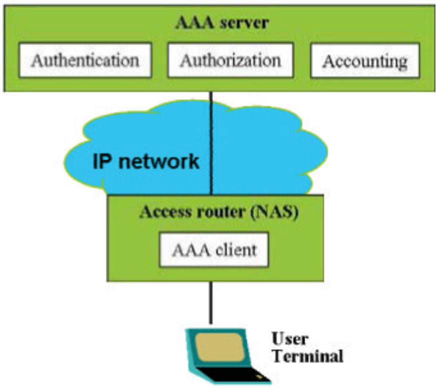


Figure 3.18. *AAA architecture*

The terminal interacts with the access router. In the case where a terminal connects from a switched network (PSTN, ISDN, GSM), the access router becomes a NAS (Network Access Server) gateway that ensures the connectivity between the switched network and IP network. Once it is physically connected to the network, the user terminal is authenticated. At the beginning of a communication between the terminal and the network, only packets belonging to the authentication protocol and addressed to the AAA server are authorized and relayed by the NAS. Upon a successful authentication, the NAS authorizes other packets coming from the user terminal to go through. This is made possible by the configuration of two ports at the NAS: a controlled port and an uncontrolled port. During the authentication phase, the traffic is going through the controlled port which recognizes the authentication traffic and lets it go through. After user authenticates, the traffic goes through the second port.

From the operator's point of view, the AAA client located on the NAS captures the authentication messages (e.g. EAP) coming from the terminal, encapsulates them into AAA messages, and sends AAA messages to the AAA server. The AAA server accesses a database that stores all the information relative to the users and necessary for authentication. In general, the AAA server and the terminal share a secret that allows the AAA server to authenticate the user.

In the context of roaming, the AAA architecture defines domains of administration. Each domain has its AAA server. A mobile user is registered with his home AAA (AAAH) server of origin and can be authenticated by any visited network or domain through an inter-domain AAA protocol. This inter-domain authentication is conducted by an AAA broker, as presented in Figure 3.19.

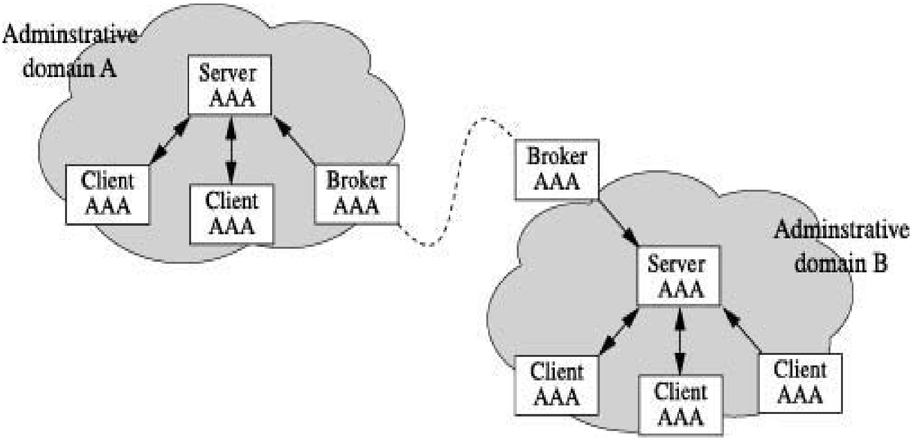


Figure 3.19. *Inter-domain AAA architecture*

The IETF has standardized protocols to implement AAA functions for:

– the terminal-NAS interface: two protocols are now envisaged for the transport of EAP messages, namely 802.1X and PANA (Protocol for carrying Authentication for Network Access) (see section 3.4.2.1);

– the interface between the NAS and the AAA server for intra-domain which is provided by the RADIUS protocol (see section 3.4.2.2);

– the interface between AAA servers for inter-domain which is implemented by the Diameter protocol (see section 3.4.2.2).

3.4.2.1. *EAP and PANA*

The AAA service usually requires a link layer protocol between the terminal and the access network. The EAP is one of the most used link layer authentication protocols. It is used to authenticate the terminal before it obtains an IP address. In Wi-Fi networks, IEEE 802.1X is the authentication scheme standard, and it uses the EAP. An alternative to the link layer protocol is PANA. It works over UDP and needs an IP address before proceeding with the authentication of the terminal. The PANA protocol encapsulates the EAP protocol, like 802.1X, but unlike 802.1X, PANA is applicable to any type of network access (Wi-Fi, WiMAX, etc.) when an IP connection can be mounted. However, it is necessary to ensure that the access network accepts only PANA messages in the beginning of the connection until the terminal is successfully authenticated. This is not straightforward because unlike EAP, which proceeds with the authentication before obtaining the IP address, PANA is running over IP and therefore cannot block other application messages at the entrance of the network unless a special filter is installed to allow PANA packets during the authentication and block any other packet, then allow all the packets when the user is authenticated. In EAP, no IP address is allocated during the authentication phase, so no application packets can go through the network until the authentication is finished.

As shown in Figure 3.20, the EAP architecture [RFC3748] involves an authenticator at the NAS that communicates with the supplicant entity in the terminal using the EAP. The server sends an authentication request to the terminal. The request depends on the authentication method. The identity of the user is known as NAI (Network Access Identifier) and, based on this NAI, the AAA server can choose the authentication method. In this architecture, the authentication server or NAS acts as a bridge between the terminal and the AAA server during the phase of user authentication. This is mainly to avoid direct communications between terminals and the AAA server for security reasons. Once the authentication is successfully completed, the terminal obtains an IP address and is authorized to issue traffic to the network.

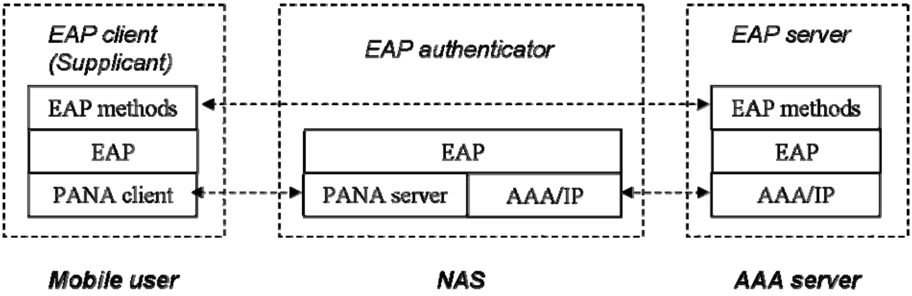


Figure 3.20. *EAP architecture with PANA at network access*

3.4.2.2. *RADIUS and Diameter*

RADIUS (Remote Authentication Dial In User Service) has been designed for intra-domain AAA service [RFC2865]. It uses IPsec between its various entities: the RADIUS client and RADIUS server, as shown in Figure 3.21. The RADIUS client in the NAS receives the request to connect to the network, initiates the process of authentication and transfers authentication messages between the terminal and the RADIUS server. The RADIUS server stores the information needed to authenticate the user. Different authentication algorithms can be used.

The messages exchanged between the RADIUS client and the RADIUS server are shown in Figure 3.22.

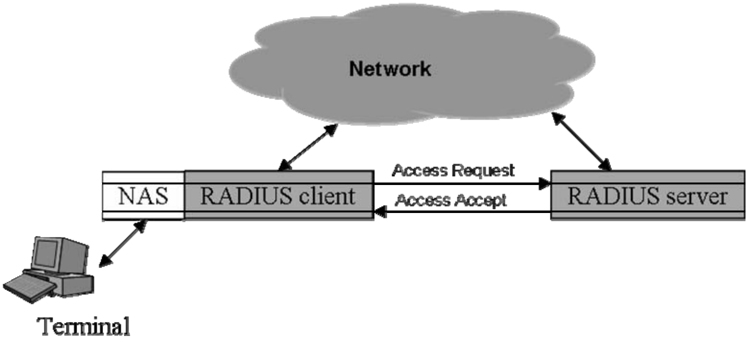


Figure 3.21. *RADIUS scheme*

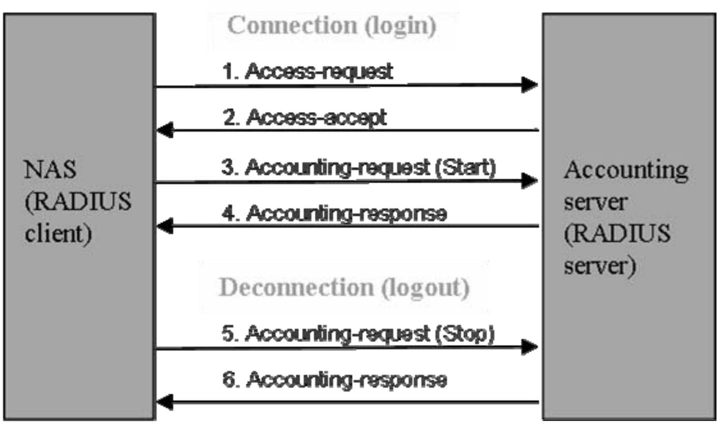


Figure 3.22. *RADIUS protocol*

With the mobility of users, the Diameter protocol was developed by the IETF to handle the AAA inter-domain authentication scheme (Figure 3.23). The RADIUS protocol is technically limited to the intra-domain authentication, and Diameter can be seen as an enhanced and scalable version of RADIUS [RFC3588]. Inter-domain mobility support, support for QoS and the extensions for accounting are some of the extensions implemented in Diameter.

The messages exchanged between the Diameter client and Diameter server are shown in Figure 3.24.

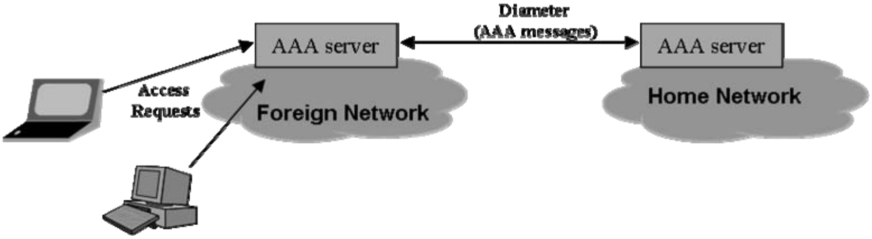


Figure 3.23. *Diameter architecture*

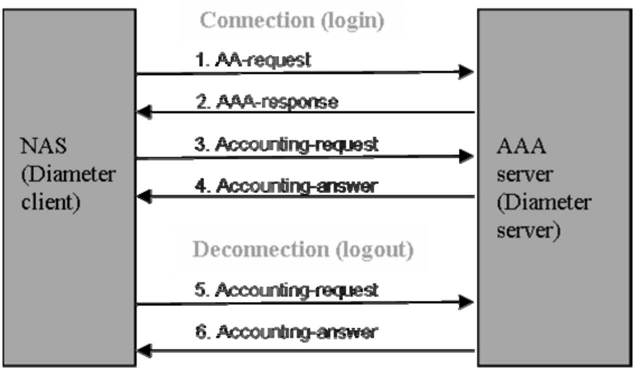


Figure 3.24. *Diameter protocol*

Note that the RADIUS and Diameter protocols are authentication and authorization protocols but they are limited somehow in the support of authorization compared to other protocols such as the COPS (Common Open Policy Service) protocol. [RFC3127] provides an AAA protocol evaluation between RADIUS, Diameter and COPS.

3.4.2.3. *Centralized authentication of users*

Many facilities require authentication of users or equipment. A first solution would be to duplicate the authentication database into each of these facilities, but this would highly increase the risk of disclosure of this database. A preferred solution which is easier and more secure to manage is to centralize this database in an LDAP server or RADIUS (see section 3.4.2.2).

Unlike RADIUS, the primary goal of an LDAP (Lightweight Directory Access Protocol) server is not to authenticate users, but make an LDAP database remotely accessible using an LDAP protocol. The LDAP database is widely used in companies to manage employees and identify their addresses, office phone, office number, etc. To restrict access to this database, LDAP identifies several roles: the administrator who controls the full database without any restriction, the user (after being authenticated) who has limited rights on the database (traditionally, writing and reading rights for his own attributes, reading rights on part of the attributes of other users) and the anonymous mode that does not require any authentication and thus leads to a very limited access to the database. The most widespread authentication is based on a login and a password which is registered to a specific attribute of the user's LDAP entry.

Any equipment that needs to authenticate a user can ask the LDAP server to do so. The procedure is as follows: first this equipment issues an LDAP request asking the LDAP server for access with the login and password provided by the user, as if the user was itself asking for this LDAP access. This forged request is rather like an identity theft with the equipment spoofing as the user. However, it enables the equipment to get an answer about the success or failure of the authentication. In the event of success, the LDAP is then closed by the equipment which grants access to some resources according to the privileges of the user.

In contrast to RADIUS, the LDAP is not designed to implement the functions of authorization and data collection. Of course, with the LDAP server can learn about the privileges associated to a user, but its role does not include serving as an AAA server, like RADIUS. Likewise, RADIUS is unable to store the attributes associated with a user.

3.5. Access control

Previous sections presented mechanisms aimed at providing confidentiality authentication and integrity services for communications. However, apart from targeting communications, attacks can also aim at other goals. Attacks against end systems can provide attackers with access to unauthorized resources. This can occur by taking advantage of weaknesses in authentication systems deciding whether communications should be established or not. This can also occur by exploiting software or hardware vulnerabilities in communication systems in order to bypass access control systems of existing resources. Finally, it can also be performed by taking advantage of the lack of proper separation between resources used by different users to monopolize resources and deny access to legitimate users or slow down their operations. In this section we consider network-based mechanisms aimed at protecting against these threats.

3.5.1. *Firewalls*

Firewalls appeared at the end of the 1980s. Their initial goal was to separate networks in order to protect insecure computers from attackers. Separation of networks is based on the amount of trust the security administrator can put in devices constituting them. The main task of the firewall is to control communications between networks with different trust levels in order prevent attacks from occurring. The notion of trust is often based on the level of control the security administrator has on operations performed by users and devices within a network. For instance, devices connected to the Internet through a network different from the protected network are often considered as unreliable since the security administrator has no way to limit operations executed by them. Within a network, devices and users are expected to be subject to the same security policy. We thus consider that they share the same level of trust which explains why communications within a network are not controlled. The frontier between two networks of different trust levels is called the security perimeter. Two general characteristics are usually expected from a firewall:

– It must be incorruptible. An attacker must not be able to change the behavior of a firewall. It must moreover have a failsafe behavior, meaning that in the event of a failure, it must limit the ability of attackers to take advantage of it.

– It must control all communications. There must be no way for devices located across the security perimeter to communicate without their communications being controlled.

The control of communications is performed by analyzing the content of exchanges and comparing this content with a policy describing authorized and forbidden content. When a communication is authorized, the firewall allows data units to cross the security perimeter. The filtering policy is not always fully defined by the security administrator. For instance, some filtering decisions are made by default.

More precisely, firewalls are normally used to provide several types of services:

– reduce the ability of attackers to attack devices within the security perimeter by limiting the resources accessible to them by filtering the types of data units that can cross the security perimeter. This strategy is usually referred to as “attack surface reduction”;

– prevent vulnerabilities in internal systems from being exploited by blocking or reformatting data units appearing as malicious;

– prevent obfuscation techniques from being used in order to prevent previously mentioned services from being correctly implemented. Obfuscation techniques are tricks used by attackers to hide their operations or to create a different understanding of communications between the firewall and the systems communicating through it.

3.5.1.1. *A taxonomy for firewalls*

Firewalls are usually classified according to several criteria [CHE 03].

The protocol level of the analysis performed by the firewall:

– At the network level, the content of the network level protocol (e.g. IP, ICMP) and transport level protocol (UDP, TCP) headers are used to decide whether packets should be accepted or not. This can for instance be used to limit the devices or services accessible to attackers.

– At the circuit level, firewalls take into account the notion of transport level connection. This allows them to check the link between packets belonging to the same communication. For instance, with TCP, when setting up a connection, a TCP connection setup segment from the source to the destination should be followed by a setup acknowledgement segment in the opposite direction.

– At the application level, firewalls require a filtering policy specific to the considered application. For instance, for the HTTP protocol, a firewall will usually provide the ability to decide which methods can be used on which objects for a given server. It will also usually reformat requests sent to a server in order to avoid ambiguous understanding of the requests between the server and the firewall.

The location. Two main classes of locations exist today:

– Network firewalls are located within a network. Therefore, they can protect devices belonging to a complete network. Their main advantage is their incorruptibility since they usually rely on specific hardware and/or software systems specifically tuned to increase their resilience to attacks. This resilience is further improved when the device is managed by a security professional.

– Personal firewalls are collocated with end systems. These tools have the ability to interact with the operating system as well as with the user in order to prevent attacks such as application hijacking that are more difficult to prevent using network firewalls. Moreover, these tools allow a security administrator to control every communication received or generated by the protected device. This type of tools tends to integrate with other host based security tools (anti-viruses, anti-spyware, etc.). Their main weakness is however their reliance on the security of the device they are expected to protect.

The transparency. Since their origin, firewalls have enforced two types of policies:

– A policy in which only data units considered as safe are authorized to cross the security perimeter. Other data units are dropped by default. This type of policy has the advantage of being able to block unknown attacks. However, defining which data units are safe for a given protocol is not always easy or even possible. This type of policy is usually called “deny by default”.

– A policy in which only data units that are considered as malicious are blocked. Other data units are forwarded by default. This policy has the drawback to require an extensive knowledge of existing attacks which is not always easy to acquire. This type of policy is usually called “permit by default”.

In practice, firewalls usually include complementary tools. Each tool can use a different combination of these criteria. This makes it difficult to define precisely and concisely what a firewall is as a whole.

3.5.1.2. *Firewall architectures*

As mentioned earlier, a firewall is usually a combination of tools often treating complementary protocols and various protocol stack levels. Such combinations cannot only increase the functionality of the firewall but can also be useful in order to increase the resilience of the filtering architecture by protecting more complex, weaker filtering tools with less complex, stronger ones. Filtering architectures also often exhibit redundancy in order to prevent a single failure from damaging the security of the whole architecture. For instance, the filtering architecture presented in Figure 3.25 shows two filtering routers. Each of them can partially supply to the failure of the other.

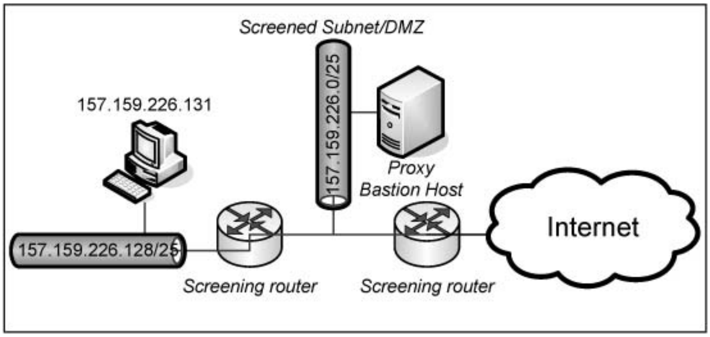


Figure 3.25. *Filter combination*

A common architecture consists of protecting application level filters through network or circuit level filters, as represented in Figure 3.25. Application filtering is implemented here through an application level filtering proxy, a program used as a mediator between internal clients or servers and external hosts. This proxy runs on a device connected to a network (157.159.226.0/25) isolated from the internal network and from the Internet through two filtering or screening routers. These routers prevent protocols other than those understood by the proxy from entering the

157.159.226.0/25 network. They also prevent direct communications between the internal network and the Internet.

The intermediate network is called the DMZ (De-Militarized Zone) since devices in this network cannot establish connections to the internal or external network by themselves. This network is also usually the place where servers accessible from the Internet should be located.

3.5.1.3. *Combination with other services*

Some tools used by firewalls can function autonomously. However, some other are usefully combined with other security tools:

– Tools used to limit access to resources are often combined in practice with authentication in order to be able to associate a user or a device with a communication at various protocol levels.

– Network and circuit level filtering tools can be associated with an intrusion detection system (IDS) in order to block or reset communications considered as harmful by the IDS. This combination is often referred to as intrusion prevention system (IPS).

– The application level filtering tools can be completed with helper systems specializing in the analysis of specific types of data. For instance, some application level filters for protocols such as FTP, HTTP or SMTP use the services of an external antivirus tool to check whether transported files can be considered as harmful.

3.5.2. *Intrusion detection*

Firewalls are not always sufficient to prevent all kinds of attacks against end systems. For example, if we consider firewalls trying to block malicious traffic, when only considering attacks that can be detected, firewalls will usually focus on blocking a small portion of these attacks. This can be explained by several reasons:

– In order to block a communications, a firewall must have strong evidence that the communication constitutes an attack. In practice, attack detection tools can usually make mistakes and consider that a benign communication is an attack (this is usually referred to as “false positive”). It is usually impossible to block communications automatically for a given attack when the level of “false positive” detection events for this particular attack is too high since it would affect the traffic of legitimate users negatively.

– Most filtering systems operations can be detected by parties communicating through a firewall. Potential attackers can therefore try to adapt their behavior to the firewall in order to hide their operations. They can also try to attack a firewall to make it fail. It is therefore often interesting to have additional means to control communications in a more furtive way.

– In terms of performance, the complexity of attack detection operations makes it difficult to introduce them in on-line devices like firewalls when these devices are used in high performance environments. In such environments, filtering operations might introduce intolerable delays for users.

Historically, initial intrusion detection approaches appeared at the beginning of the 1980s ([ME 06], Chapter 3), roughly 10 years before the first firewalls were designed. Earlier tools focused on detecting malicious usage of end systems by analyzing the occurrence of particular events. Today many firewalls include the ability to detect intrusions and many intrusion detection systems have the ability to interact with the communications they control. Therefore, the distinction between these two types of technology tends to blur.

3.5.2.1. *A taxonomy for intrusion detection tools*

Defining criteria in order to classify existing intrusion detection systems has long been an active research field. This section is mainly based on the criterion defined in [DEB 00].

The source of information. As with firewalls, intrusion detection systems can be located within a network and capture information about communications originating or targeted at devices located within this network. They can also be collocated with end systems and collect information such as applications or operating system logs.

The detection method. Two classes or detection methods are used today:

– Knowledge-based methods use the assumption that attacks can be identified by analyzing evidence of their occurrences. Defining and identifying evidence assumes a good understanding of the corresponding attack. A lack of understanding of attacks or the usage of an inappropriate scheme to define evidence can lead to false or weak evidence. This can lead the detection method to wrongly identify benign traffic as an attack or to miss attacks. Another problem with these methods is the number of pieces of evidence a tool has to manage. Since the number of pieces of evidence is linked to the number of known attacks, this number is constantly growing. For instance, the number of evidence used by a tool such as Snort [SNO 06] has grown from 5,000 to 15,000 between 2005 and 2008. Another problem faced by these methods and similar to the problem we encountered when describing firewalls using the “permit by default” policy is that they assume an up-to-date knowledge of attacks. This prevents unknown attacks or attacks for which evidence has not yet been defined from being detected.

– Behavior-based methods use the assumption that attacks can be detected by observing a set of parameters describing the operation of the protected system. These parameters are chosen so that they exhibit a form of stability when the system works normally and that they exhibit a form of change when the system is under attack. The reason why we use the term "form of stability" here can be explained by the fact that any parameter describing a real-life system exhibits some instability. The difficulty when building such methods is to find techniques allowing natural, legitimate instability to be extracted from the instability created by attacks. In practice, it is often impossible to extract every possible source of legitimate instability leading to imperfect models. On the other hand, such methods have the ability to detect unknown attacks as long as these attacks generate a change in the set of observed parameters.

The detection paradigm:

– Some intrusion detection systems attempt to detect whether a system reaches a particular state characterizing the completion of an attack. This can for example be the detection of whether a malicious code was added to an executable file on a computer. This first mode of operation is called "state-based" since it attempts to discover a particular state of the system.

– Some other systems focus on the detection of the set of actions that are performed by the attacker to reach this particular state. For instance, in order to introduce his malicious code, the attacker might have needed to execute the following actions: capture a password, fake the identity of the owner of the file, add executable code to the file. This second mode of operation is called "transition-based" since it attempts to recognize the various steps executed by an attacker to reach his goal.

The usage frequency. Detection operations can either be performed in real time or on a periodical basis. Real time detection is not always possible since some detection operations are too computationally expensive to provide real time results or would slow down the system monitored so much that it would become unusable. The main drawback of a non-real time approach is to be only able to detect attacks *a posteriori*.

The behavior after detection. As mentioned in section 3.5.1, detection operations can be followed by a reaction in order to block the attacker's actions or limit the damages generated by the attack. These systems, called "active intrusion detection systems", are opposed to "passive detection systems" that do not automatically perform reaction operations on their own. As mentioned earlier, the benefit of reactive systems is limited to attacks that can be detected with a low ratio of "false positive" detection events.

3.5.2.2. *Characterizing attacks*

Firewalls and intrusion detection systems sometimes require a good understanding of attacks in order to be able to recognize them. This is for example the case for firewalls and intrusion detection systems using knowledge-based detection methods. A rather complex problem is how to obtain this knowledge. Indeed, attackers often do their best to hide their discoveries in order prevent potential victims from protecting themselves. On the other hand, victims of attacks are often reluctant to share information about the security of their information system and the way it might have been compromised. In order to solve this problem, several types of solutions have been suggested.

Honeypots. Honeypots have become very popular during the last 10 years; however, the notion of honeypot was introduced a long time ago. In 1990, Bill Cheswick [CHE 90] described how he succeeded in studying the behavior of an attacker by building a simulated environment in which operations executed by the attacker were logged and analyzed. By the end of the 1990s, a set of tools had been produced in order to automate most of the actions Bill Cheswick used to execute by hand. A new additional idea is to use dedicated devices to study attackers' actions. The main advantage of using dedicated devices is that since these devices are not expected to provide any true service to legitimate users, any interaction between these devices and their potential users can be considered as malicious. Current honeypots are mostly classified depending on the level of interactivity they support. This level represents the level of realism of the simulated environment provided to potential attackers. Tools with a low interactivity level usually only simulate a part of the behavior of a real system. For instance, they can simulate a TCP/IP protocol stack or a particular service. On the other hand, high level tools aim at simulating a full system for example by simulating the execution of a complete operating system through an emulator or virtual machine. Both types of tool must provide to the honeypot owner the ability to control and record any operation executed by the attacker. The ability to control attackers operations is necessary in order to limit their ability to attack other devices from the honeypot.

Vulnerabilities market. Another strategy in order to obtain information about attacks has been to start a market for them [MIL 07]. Some companies are now offering monetary rewards for security researchers providing them with new vulnerabilities. The analysis of such vulnerabilities is then used in order to generate detection and protection systems for the companies' customers. It remains to be seen if this strategy succeeds in curbing the number of attacks in the long term.

The characterization of attacks has received a lot of attention from the research community in the recent past. Honeypots provide the ability to understand attacker operations. However, transforming this knowledge into models that can be used by filtering or attack detection tools is often a task that requires considerable effort. Today, this operation is mostly performed by hand, which limits the efficiency of these tools against new attacks. Automating the generation of these models is an ongoing research work [KIM 04, POR 06]. Another limitation to honeypots is that they provide a microscopic view about attackers' actions. This view does not always provide the ability to understand distributed attacks when hundreds or thousands of entities take part in an attack. Another active field of research is therefore the development of techniques allowing such distributed attacks to be observed [MOO 01]. A field where such techniques have proven useful is the case of denial-of-service attacks. Finally, most attacks require the execution of several operations by the attacker. Understanding the relations between these operations can help in gaining a better idea of the goal of the attacker which can improve the ability to detect the attack itself. This problem of correlating detection events produced by various intrusion detection tools has also received a lot of attention in recent times [KRU 05].

3.6. Conclusions

A number of vulnerabilities weaken the networks and make users feel very suspicious with regards to network security. So far, several security mechanisms have been developed to meet the needs of businesses and individuals. However, these mechanisms are vulnerable and they do not protect against all network attacks like denials of service. Actually, they provide a first level of security that can thwart most logical attacks, i.e. those made using ordinary means. The mechanisms originally developed for wired networks have subsequently been adapted to wireless networks. In the rest of this book, wireless mechanisms are presented.

3.7. Bibliography

[AVISPA] AVISPA project, http://www.avispa-project.org, 2006.

[CHE 90] CHESWICK B., "An evening with Berferd in which a cracker is lured, endured, and studied", *Proceedings of the USENIX Conference*, Jan 20, 1990.

[CHE 03] CHESWICK W., BELLOVIN S., RUBIN A., *Firewalls and Internet Security, Repelling the Wily Hacker*, Second Edition, Addison-Wesley Professional, 2003.

[DEB 00] DEBAR H., DACIER M., WESPI A., "A revised taxonomy for intrusion detection systems", *Annales des Télécommunications*, vol. 55, p 361-378, 2000.

[FRA 08] FRAHIM J., HUANG Q., *SSL Remote Access VPNs (Network Security)*, Cisco Press, June 2008.

[FRE 07] FREEMAN T., HOUSLEY R., MALPANI A., COOPER D., POLK T., "Server-based Certificate Validation Protocol (SCVP)", draft-ietf-pkix-scvp-33, December 2007.

[FUN 08] FUNK P., BLAKE-WILSON S., "EAP Tunneled TLS Authentication Protocol Version 1 (EAP-TTLSv1)", draft-funk-eap-ttls-v0-05, April 2008.

[GUP 02] GUPTA M., *Building a Virtual Private Network*, Premier Press, 2002.

[HOU 99] HOUSLEY R., FORD W., POLK W., SOLO D., *Internet X.509 Public Key Infrastructure: Certificate and CRL Profile*, RFC 2459.

[HOU 02] HOUSLEY R., POLK W., FORD W., SOLO D., *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, RFC 3280.

[JAY 07] JAYARAM P. *et al.*, "Protocol for Carrying Authentication for Network Access (PANA) Framework", draft-ietf-pana-framework-10, September 2007.

[KIM 04] KIM H., KARP B., "Autograph: Toward Automated, Distributed Worm Signature Detection", *Proceedings of the 10th Usenix Security Symposium*, San Diego, CA, 2004.

[KRU 05] KRUEGEL C. *et al.*, *Intrusion Detection and Correlation, Challenges and Solutions*, Springer, 2005.

[ME 06] ME L., DESWARTES Y., *Sécurité des systèmes d'information*, Hermes, 2006.

[MIL 07] MILLER C., "The legitimate vulnerability market inside the secretive world of 0-day exploit sales", *Workshop on the Economics of Information Security 2007*, June 2007.

[MOO 01] MOORE D. *et al.*, "Inferring Internet Denial-of-Service Activity", *Proceedings of the 10th USENIX Security Symposium*, Washington DC, 2001.

[POR 06] PORTOKALIDIS G., SLOWINSKA A. AND BOS H., "Argos: an emulator for fingerprinting zero-day attacks", *Proceedings of the ACM EuroSys 2006*, 2006.

[RES 01] RESCORLA E., *SSL and TLS: Designing and Building Secure Systems*, Addison-Wesley, 2nd Edition, March 2001.

[RFC1994] SIMPSON W., *PPP Challenge Handshake Authentication Protocol (CHAP)*, RFC 1994, August 1996.

[RFC2246] DIERKS T., ALLEN C., *The TLS Protocol Version 1.0*, RFC 2246, January 1999.

[RFC2402] KENT S., ATKINSON R., *IP Authentication Header,* RFC2402, November 1998.

[RFC2403] MADSON C., GLENN R., *The Use of HMAC-MD5-96 within ESP and AH,* RFC 2403, November 1998.

[RFC2406] KENT S., ATKINSON R., *IP Encapsulating Security Payload (ESP),* RFC 2406, November 1998.

[RFC2404] MADSON C., GLENN R., *The Use of HMAC-SHA-1-96 within ESP and AH,* RFC 2404, November 1998.

[RFC2409] HARKINS D., CARREL D., *The Internet Key Exchange (IKE)*, RFC 2409, November 1998.

[RFC2451] PEREIRA R., ADAMS R., *The ESP CBC-Mode Cipher Algorithms,* RFC 2451, November 1998.

[RFC2560] MYERS M., ANKNEY R., MALPANI A., GALPERIN S., ADAMS C., *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*, RFC 2560, June 1999.

[RFC2587] BOEYEN S., HOWEST T., RICHARD P., *Internet X.509 Public Key Infrastructure LDAPv2 Schema,* RFC 2587, June 1999.

[RFC2716] ABOBA B., SIMON D., *PPP EAP TLS Authentication Protocol,* RFC 2716, October 1999.

[RFC2865] RIGNEY C. *et al.*, *Remote Authentication Dial in User Service,* RFC 2865, June 2000.

[RFC3127] MITTON D. *et al.*, *Authentication, Authorisation and Accounting: Protocol Evaluation,* RFC 3127, June 2001.

[RFC3193] PATEL B., ABOBA B., DIXON W., ZORN G., BOOTH S., *Securing L2TP Using IPsec,* RFC 3193, November 2001.

[RFC3588] CALHOUN P. *et al.*, *Diameter Base Protocol,* RFC 3588, September 2003.

[RFC3748] ABOBA B., BLUNK L., VOLLBRECHT J., CARLSON J., LEVKOWETZ H., *Extensible Authentication Protocol (EAP)*, RFC 3748, June 2004.

[RFC4034] ARENDS R., AUSTEIN R., LARSON M., MASSEY D., ROSE S., *Resource Records for the DNS Security Extensions*, RFC4034, March 2005.

[RFC4186] HAVERINEN H., SALOWEY J., *Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)*, RFC 4186, January 2006.

[RFC4187] ARKKO J., HAVERINEN H., *Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)*, RFC 4187, January 2006.

[RFC4302] KENT S., *IP Authentication Header*, RFC 4302, December 2005.

[RFC4303] KENT S., *IP Encapsulating Security Payload (ESP),* RFC 4303, December 2005.

[RFC4305] EASTLAKE D., *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*, RFC4305, December 2005.

[RFC4306] KAUFMAN C., *Internet Key Exchange (IKEv2) Protocol*, RFC 4306, December 2005.

[RFC4308] HOFFMAN P., *Cryptographic Suites for IPsec*, RFC 4308, December 2005.

[RFC4555] ERONEN P., *IKEv2 Mobility and Multihoming Protocol (MOBIKE)*, RFC 4555, June 2006.

[SCH 96] SCHNEIER B., *Applied Cryptography: Protocols, Algorithms and Source Code in C*, Second Edition, John Wiley & Sons, 1996.

[SNO 06] Snort, www.snort.org.

[X509] UIT-T X.509, *Information Technology – Open Systems Interconnection – The Directory: Authentication Framework*, November 1993.

[X800] UIT-T X.800, *Data Communication Networks: Open Systems Interconnection (OSI); Security, Structure and Applications. Security Architecture for Open Systems Interconnections for CCITT Applications*, 1991.

This page intentionally left blank

Chapter 4

Wi-Fi Security Dedicated Architectures

4.1. Introduction

Previous chapters focused on standardized security mechanisms, whereas this chapter will emphasize architectures that were designed due to the new business generated by wireless infrastructures. First, authentication issues of wireless users in "hot spot" – also called "captive portal" – architectures will be discussed. Next, recent architectures such as WIDS/WIPS (Wireless Intrusion Detection Systems/Wireless Intrusion Prevention Systems) aiming at detecting any malicious wireless activity will be detailed. Lastly, we will focus on an architecture designed for research activities such as wireless honeypots that aim to discover and understand the attacks and associated techniques on wireless media such as Wi-Fi.

4.2. Hot spot architecture: captive portals

4.2.1. *Overview*

Hot spots are dedicated Wi-Fi networks usually deployed in airports and railway stations that give users the opportunity to connect to the Internet or their Intranet thanks to wireless connectivity.

This kind of network access was firstly deployed by providers in areas where the users are traveling – and thus where they should not have any network connectivity – but now, numerous hot spots are also deployed in private areas like hotels or

Chapter written by Franck VEYSSET, Laurent BUTTI and Jerôme RAZNIEWSKI.

companies, providing customers or visitors with the capability to connect to the Internet.

The hot spot architecture is based on the “captive portal” technology. This recent technology was created thanks to the deployment of public wireless networks, even if the idea behind this is also applicable to wired networks. Access control and authentication are performed thanks to the captive portal. The main strength of this technology is ergonomics as there is no impact on the client’s computer configuration.

4.2.2. Captive portal overview

A captive portal is composed of:

– a dynamic rules bases firewall,

– a Web server,

– an authentication framework and database,

– (optionally) a billing framework.

This can be (briefly) described as below:

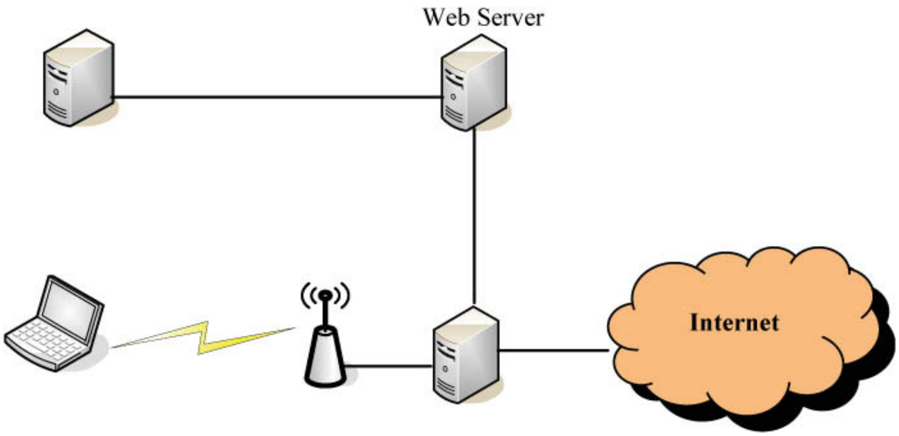


Figure 4.1. *Captive portal*

1) *Redirection.* When a computer associates with the "Open" Wi-Fi access point, it will firstly negotiate a DHCP lease. The wireless client will be redirected to the Web server whenever he will ask to go to the Internet (opening its browser and asking for www.joe.com). This redirection is performed thanks to a HTTP 302 (moved temporarily) code that is correctly understood by the popular Web browser. The captive portal will thus redirect the connection to a HTTPS Web server in order to authenticate the Web server using public cryptography and the use of Transport Layer Security (TLS) protocol. The presented Web page is the provider portal page where the user will always be redirected until he succeeds in his authentication to the hot spot.

2) *Authorization.* When the user authenticates himself to the captive portal (by providing a valid username/password or a valid token), the authentication framework will then authorize the user to communicate with the Internet by dynamically configuring the rule set applied on the firewall. Most captive portals rely only on the IP address to authorize the user on the firewall, while some others may also use the MAC address in order to prevent spoofing attacks on the MAC address.

3) *Connection.* When the firewall has configured the new rule set for the authenticated user, the template security policy (applied by the provider) is enforced and basically the user now has access to the Internet.

4) *Disconnection.* The user may be able to close the connection to the captive portal by sending a logoff through a specific Web page on the captive portal. Also, most of hot spot architectures use other techniques to detect if the user has left the architecture (e.g. by sending ARP probes or observing DHCP renewal).

4.2.3. *Security analysis*

4.2.3.1. *General overview*

The captive portal's access control is based on IP and/or MAC addresses. This may be trivially spoofable by common techniques and tools on most operating systems even if Unix-based are more flexible to perform such techniques.

Moreover – and probably more importantly for the hot spot customer – the hot spot architecture relies on "open" Wi-Fi access points; as such, there is no data encryption or integrity on the wireless network. This may be a serious issue if the user is not aware of this (for example, using some cleartext protocols such as HTTP or POP3 is not recommended as there is no encryption and thus are possibly eavesdropped by an attacker).

Basically, the main protection mechanism achieved by the hot spot is the entry point firewall that enforces a security policy between the wireless network and the Internet. Most of the time, the hot spot provider uses a security policy that prevents the Internet from attacking its wireless customers, and which enables its customers to use any protocol to the Internet without any filtering mechanisms (and especially the protocols used for connecting to Intranets such as IPsec and SL/TLS).

From a legal point of view, it is hard to prove that a specific user performs fraudulent actions using the hot spot architecture, as we will show later in this chapter, because of inherent security weaknesses regarding spoofing.

Requirements on the user's configuration are reduced to a few: a computer with a Wi-Fi network card and a Web browser. Providing a simple and compatible solution for user access is a provider requirement in terms of business.

4.2.3.2. *Security analysis*

The goal of this section is to describe the captive portal techniques to detect and overcome security issues that will be described below.

IP spoofing only

The attacker is able to retrieve a valid IP address – by a passive eavesdropping as there is data encryption on the wireless network – that is authorized to surf to the Internet. He will spoof its IP address in order to bypass the firewall rules that should only be based on IP filtering (level 3 on the OSI layer).

If the hot spot architecture does not have any correlation between IP and MAC addresses, it will be hard to detect the issue, but the legitimate user and the attacker will share the same IP address (with different MAC addresses) and thus it will be a network issue as the TCP packets sent by one of the parties will be dropped by the other (sending back RST to unsolicited SYN/ACK) leading to TCP disconnections.

If the hot spot architecture uses a MAC filtering mechanism (generally whenever the architecture is "flat", i.e. no routing between the user and the firewall), it will not be possible to bypass these filters by IP spoofing only.

MAC spoofing only

As the firewall filtering relies on IP addresses, it will not be possible to bypass the filtering mechanisms with a MAC only spoofing.

As the IP address is not authorized on the firewall rule set, all packets with an unauthorized IP address and an authorized MAC address will be rejected by the firewall.

IP and MAC spoofing

This is usually the most effective method as it will be hard for the architecture to distinguish between the flows between legitimate and illegitimate users. For this kind of attack, we observe two cases:

– If the legitimate user is still connected to the hot spot, the Internet access is unstable for both but is still possible (by using some tweaks).

– If the legitimate client is disconnected without logging off using the Web portal, then the attacker will be able to retrieve his profile and continue the surfing session (depending on the payment: post-paid, pre-paid).

This attack technique will make it possible to:

– access the Internet bypassing the firewall filtering rules;

– disturb the legitimate user connections.

4.2.3.3. *Possible improvements*

Hot spot providers are usually aware of these common issues regarding security and more specifically billing. As these issues are related to access control and lack of attack detection in common hot spot architecture, this section will discuss possible improvements that will aim at raising the overall difficulty of performing such attacks.

Access control improvements

A simple but effective improvement is to add the user's operating system detection to correlate the MAC/IP address with. The assertion is that most attackers will use Unix-based operating systems, unlike contrary to legitimate users who will rely on Microsoft Windows-based operating systems. Thus, if the same IP address has two different operating system fingerprints at the same moment, an IP spoofing attack is possible: this is simple but effective in practice as today it is hard to perfectly mimic other operating systems by TCP/IP stack tweaks.

This technique may be achieved by an active or passive operating system fingerprint. Obviously, passive is the best option as you might not be able to port scan your customers. In the open source world, the operating system passive fingerprinting can be performed by a well-known tool, *p0f*.[1]

[1] p0f: http://lcamtuf.coredump.cx/p0f.shtml, which is a passive operating fingerprinting tool relying on the analysis of TCP/IP headers such as the TTL at the IP level and TCP options at layer 4.

This technique will drastically improve the overall security of your hot spot architecture regarding fraud thanks to IP spoofing.

Device discovering improvements

One requirement for overcoming billing issues is to detect whenever the customer leaves the hot spot in order to stop the billing mechanism and to reconfigure the dynamic firewall to redirect the IP address to the captive portal. This is necessary to reduce the window of opportunity for the attacker.

To detect that a customer leaves the architecture, several options are possible: logoff window, MAC address lookup in ARP tables of network switches, ARP probes, ICMP probes, DCHP renewal, etc.

Logoff window

When a user is authenticated to the Captive Portal, a logoff window is accessible and triggerable. This logoff window is useful for:

– giving the customer the opportunity to manually stop the billing whenever he clicks on this window;

– periodically sending information to the captive portal in order to tell that the customer is still active; these probes are usually securely sent over SSL/TLS. If the captive portal does not receive the customer probes then it will consider that he has left the hot spot and thus will shut down the current authorization linked to the authenticated user.

This kind of technique is important in order to reach an acceptable level of security. If the legitimate user cannot use the logoff mechanism and if this logoff mechanism is cryptographically protected, the attacker will not be able to spoof this mechanism in order to fool the captive portal (telling it that the legitimate is still active). The captive portal will thus be able to detect this issue and will shut down the session for this particular user.

DHCP renewal

In this case, the captive portal retrieves information from DHCP servers. As DHCP leases are usually short timed, if the legitimate user leaves the architecture and does not renew his DHCP lease, then the captive portal will de-authenticate the legitimate user. The attacker must then mimic the DHCP renewal process in order to bypass this mechanism. Even if not perfect, this technique raises the bar for fraud attempts due to IP spoofing.

4.2.4. *Conclusions*

The captive portal technique is easy to implement and provides great ergonomics for the average user. There is no requirement on the client side regarding the operating system, wireless drivers or any security configuration (like WPA or WPA2). The drawback is that there is neither confidentiality nor integrity on the wireless side, leading to easy eavesdropping on data communications and possibly IP/MAC address spoofing.

The overall security of this architecture is mainly improved by techniques we describe later that make attacks much more difficult – but still possible – for non-skilled people. Hot spots cannot rely on enhanced security mechanisms standardized in IEEE 802.11 such as the IEEE 802.11i standard, because it impacts on the user's configuration and ergonomics which is contrary to the hot spot business model.

This kind of option is feasible when integrated in connection kits that configure the underlying operating system and applications in order to use WPA/WPA2 and then provide a next level of security. Even if technically feasible for the hot spot provider, it can be a major hurdle for the business as ergonomics is always one of the first requirements for the success of an offer.

4.3. Wireless intrusion detection systems (WIDS)

4.3.1. *Introduction*

Wireless networks based on the IEEE 802.11 standard are now very well known and widely deployed. This technology is present within a large set of equipment ranging from access points to wireless printers and cellular phones. Today, 802.11 chipsets are present in most recent laptops, and wireless access for company employees is widely used.

Unfortunately, network security risks are inherent with any wireless technology. Before any deployment, wee must take into account pros and cons, especially when the company's business is critical. It's security policy must be reviewed and improved whenever the deployment of wireless technology for employee access becomes attractive (from a business or ergonomics point of view).

Of course, thanks to enhanced security mechanisms at layer 2 (WPA/WPA2 with strong authentication) and above layers (SSH, IPsec, SSL/TLS, etc.), deploying secure wireless access is possible. The design and implementation of wireless access will take into account requirements regarding security (authentication, confidentiality and integrity) and ergonomics (ease of use, configuration, administration, etc.). Having robust wireless access will not prevent some attack attempts, especially regarding the use of rogue access points that will be interconnected to the company's physical network avoiding all deployed security mechanisms.

Thus, some important risks remain:

– denial-of-service attacks on layer 2 thanks to de-authentication and disassociation frames;

– misconfigured access points connected to internal networks;

– rogue access points not connected to internal networks (but mimicking legitimate access points);

– rogue access points connected to internal networks.

Potential attacks may have serious impacts on overall security of the company. The main issue related to these attacks is the fact that they are not easily detectable from the wired side: classic security monitoring relying on events on the wired side is blind regarding these issues.

WIDS intend to address this issue: how to observe the wireless radio traffic to detect abnormal events regarding the security policy of a protected company's physical location. These systems should be able to detect whether a rogue access point is connected to internal networks or not. These kinds of test are usually performed manually during security audits, but of course, security audits cannot be performed every day, every minute… This could be done by automated tools: WIDS.

These systems are designed to detect most attacks on the wireless side, such as denial of service, frame injection, rogue access point detection, etc. The last item is far from easy but is critical for the interest of these tools. Basically, one on the first steps is to manage a white-list of authorized MAC addresses (BSSID) and authorized network names (ESSID) in order to trigger alarms whenever some access points are not white-listed. Of course, these access points can be legitimate interfering access points (neighbors), which is the reason why the intrusion detection system must implement some techniques to check if the access point is interconnected to internal networks or not.

4.3.2. *Wireless intrusion detection systems architectures*

WIDS work at OSI layer 3. As attacks and issues occur on the wireless side, there is no interest in analyzing upper layers on the wireless side (by the use of classic intrusion detection tools such as Snort). As a matter of fact:

– upper layers attacks going over the wireless network will go through the access points and then will go on the wired network where they can be analyzed by appropriate intrusion detection systems;

– wireless network traffic may be encrypted thanks to TKIP/CCMP (implemented in WPA/WPA2) or IPsec/SSL/TLS protocols, so its analysis cannot be interesting.

Even if it is technically possible for a WIDS to perform analysis at any layer of the OSI model, they are not devoted to perform analysis at upper layers of the OSI model. They must address wireless-specific issues that may be detected only from the wireless side.

Two different types of WIDS architecture are possible:

– *Integrated* – intrusion detection is performed on the same physical equipment as the one that provide network access;

– *Overlay* – intrusion detection is performed on dedicated equipment that is independent of those that provide network access.

Integrated architecture

This architecture relies on 802.11 access points that already provide network access. Regarding costs, this solution may be attractive as a unique piece of equipment shares both the intrusion detection and network access parts, but both listening to the wireless traffic and providing network access requires the selection of a static wireless channel (no channel hopping possible) and powerful processing unit. Consequently, this architecture cannot be as effective as overlay architecture, especially because of the channel constraints (using the same channel for both the intrusion detection and network access).

Vendors who chose this integrated architecture usually scan the wireless network whenever the access point has enough resources to do so. There must be a minimal impact on network access availability. A balance between network access and intrusion detection efficiency must be achieved!

Overlay architecture

This architecture relies on dedicated equipment for listening to the wireless network. Of course, most vendors offer the opportunity to choose between the access point function and the intrusion detection function on the same piece of equipment. Thus, when deploying wireless network architecture, it is necessary to choose how many and where will you deploy access points and intrusion detection probes. This is a great advantage in terms of flexibility as it can usually be performed thanks to a nice Web GUI.

This architecture provides better results as the intrusion detection is performed on dedicated equipment. The main drawback is regarding the costs which are usually greater than integrated architecture. In any case, you should prefer a vendor that provides the opportunity to switch the same piece of equipment between the access point and intrusion detection probe!

4.3.3. *Wireless intrusion detection events*

The main events to be detected are:

– wardriving;

– frame injection;

– denial of Service;

– MAC spoofing;

– attacks on authentication mechanisms (e.g. EAP);

– attacks on confidentiality mechanisms (e.g. WEP);

– misconfigured access points;

– rogue access points;

– rogue access points connected to internal networks;

– fake access points;

– double attachment.

In this chapter, we will not focus on intrusion detection techniques and algorithms that would take a whole book to analyze and explain. However, it is important to understand that attacks on wireless networks do not evolve drastically over one or two years, so, intrusion detection systems should now be very effective on false positives and false negatives. As is the case for any intrusion detection system, false positives are a serious issue that can prevent the technology to be effective. If a high rate of false positives is observed, then the confidence in intrusion detection techniques will decrease drastically and its alarms will be deactivated or deleted. The intrusion detection system must evoke confidence in the network administrators who will be in charge of operating these systems; if this is not the case, in practice the intrusion detection systems alarms will be ignored and the architecture will be abandoned.

Another important point concerns the implementation of recent amendments on the IEEE 802.11 standards. Some of them have important impacts regarding security and implemented techniques to detect and analyze abnormal events. Standards such as 802.11n, 802.11e for QoS, 802.11r for fast roaming, etc., must be taken into account in the design of the WIDS by the vendor. Thus, when choosing the vendor, checking their ability to implement new standards is an important requirement!

4.3.4. *WIDS example*

This section is an example and cannot be fully exhaustive regarding all implemented techniques in WIDS.

A WIDS is generally composed of several parts: wireless probes, event collection, event aggregation, event correlation, event database storage, visualization GUI, administration GUI and other parts such as intrusion prevention techniques and geolocation that will be described in the following sections.

These architectures schematically need to:

– listen to the wireless network: which is quite easy thanks to a wireless network card in “monitor” mode;

– analyze the wireless traffic captures: using the mean of static signatures rule set or anomaly detection algorithms (for example, to detect MAC spoofing), these components are the code of the intrusion detection system;

– transmit the events to a central collector;

– aggregate events to reduce the overall number of events stored in the database;

– correlate events in order to reduce the number of events and also to enrich the semantics of these events (typically, a large number of de-authentications during a certain timeslot is likely to be a denial-of-service attack);

– detect if rogue access points are interfering (neighbors), legitimate or illegitimate;

– enrich the events database to provide the network administrator with precise alerts;

– provide the network administrator a GUI for supervision and administration.

4.3.5. *Rogue access point detection*

Rogue access point detection is usually performed thanks to wireless frames sent by any access point. The WIDS will compare the MAC addresses and ESSID network names with a user-defined white-list in order to trigger alarms whenever the comparison detects an unknown access point. However, this is not sufficient: as a matter of fact, the WIDS must evaluate if the rogue access point is interconnected to internal networks or is only an interfering access point.

Several techniques are possible. For example, one method is to rely on an internal equipment database in order to correlate the discovered MAC addresses on the wireless network with the MAC addresses on the wired network. The Open Source tool NetDisco (http://www.netdisco.org) is of great value for this task thanks to its discovering techniques that helps in building a map of all equipment in internal networks.

Thanks to this database, it is easy to use several heuristics to evaluate the rogue access point:

– MAC address +/– 1 of source emitted MAC addresses of access points: as a matter of fact, most access points have a BSSID similar to their wired Ethernet MAC address.

– Client's source MAC address: as a matter of fact, wireless operates at layer 2, thus these MAC addresses are learnt by the interconnected wired switches.

– Client's destination MAC address: these MAC addresses are potentially internal MAC addresses.

Thanks to these techniques, we can achieve an effective correlation mechanism which tells us if the detected rogue access point is interconnected to internal networks or not, which is the most interesting point in detection!

Of course, these techniques help us to identify if the rogue access point is critical or not, but it cannot define the physical location of the rogue access point precisely. Some WIDS also provide means to estimate the physical location of wireless equipment thanks to geolocation techniques which will be described in later chapters.

4.3.6. *Wireless intrusion prevention systems*

Intrusion detection has a serious drawback: it only provides detection. Intrusion prevention tries to mitigate the identified risks by using techniques to prevent the attacks from being effective. Today, most wireless intrusion detection vendors provide means to achieve prevention. For example, it could be interesting to prevent legitimate clients from connecting to a rogue access point.

If the detection system is able to detect a rogue access point interconnected with internal networks, it represents a serious threat for the company. However, as a detection system, nothing can be done regarding sending alarms to security operators in order to manually mitigate the issue. During the reaction period, malicious activities may occur and will not be prevented by anyone. This is one of the reason why wireless intrusion prevention systems were designed: to prevent the exploitation of wireless security issues.

The mitigation techniques usually rely on two different aspects: the first approach is to implement wireless mitigation techniques, while the second approach is to implement wired mitigation techniques.

Wireless mitigation techniques

This technique relies on the capability to easily perform denial-of-service attacks on 802.11 wireless networks. As a matter of fact, sending de-authentication or disassociation frames to a wireless equipment will disconnect the 802.11 connection. These frames are not signed, thus denial of service is quite easy, because sending these frames does not require special hardware or software. The techniques are very efficient and are often used to perform malicious attacks, so it is odd to see that intrusion prevention systems use attack techniques to implement mitigation mechanisms.

Even if effective, these techniques cannot be easily activated. As a matter of fact, if the detection mechanisms are fooled, it would be possible for the attacker to make the intrusion prevention systems perform attacks for him! This is the usual case with active protection: any detection error may lead to drastic issues. Just imagine that you prevent legitimate users from connecting to legitimate access points… We could advice the reader to carefully consider these kind of techniques.

Wired mitigation techniques

This technique relies on the operating of internal wired network equipments. It requires interconnecting the wireless intrusion prevention system with administrative tools. It will configure on-the-fly network equipment on which rogue access points are interconnected in order to deactivate the wired ports on switches.

Another option is to quarantine the rogue access point in a dedicated VLAN in order to analyze the security issue.

In any case, this mitigation technique can only be effective against rogue access points interconnected to internal networks. It is not effective against rogue access points that are not interconnected to internal networks and that try to catch legitimate wireless clients to perform attacks on them.

As usual, this mitigation technique must be carefully used, as an attacker may fool the tool in order to deactivate legitimate access points.

This architecture is quite interesting and efficient, but must be assisted by a human validation process in order to prevent configuration issues.

4.3.7. *802.11 geolocation techniques*

Geolocation is quite important regarding wireless intrusion detection. It aims to discover the physical location of 802.11 transmitters which is usually identified thanks to its MAC address or frame types (when the attacker spoofs the MAC address of legitimate equipment).

As a matter of fact, finding the exact location of the source attack under a denial of service is quite interesting. This technique aims at finding the location of malicious equipments, hence equipment that does not want to be discovered and that may implement techniques to fool geolocation tools (by modifying the transmit power for example). Other geolocation techniques exist that rely on cooperative tools that are installed in all equipment which advertises its physical location thanks to standardized methods.

Wireless geolocation techniques usually rely on the received signal strength index reported by the wireless interface and some mathematical propagation models depending on the physical environment of the wireless network (walls, concrete, etc.). They will thus provide some areas that maximize the probability of the presence of a given equipment according to their propagation models and acquired data.

4.3.8. *Conclusions*

These technologies are important whenever it is critical to detect wireless attacks and especially rogue access points interconnected with internal networks. Risks are increasing due to the deployment of 802.11 and it is quite hard today to mitigate these risks without dedicated infrastructures such as wireless intrusion detection (or prevention) systems. These technologies cannot overcome all wireless security risks, but may be able to mitigate most risks to an acceptable level. These tools are to be considered whenever deploying secure 802.11 network architectures is a requirement. Of course, as an additional tool, it requires administration and supervision to be effective.

4.4. Wireless honeypots

4.4.1. *Introduction*

Honeypots are becoming quite common in the world of security. The official definition of honeypots was given by Lance Spitzner: "*A honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource.*"

This definition is straightforward: honeypots have no operational function, so any use of them is suspicious. These suspicious events are errors or unauthorized uses which is very useful because the analysis of honeypots events will be easier than intrusion detection systems where it is necessary to distinguish attacks within an important flow of normal events. Honeypots help in learning more about attack tools, techniques and motivations of attackers. Observing a dedicated honeypot is much easier than observing a whole bunch of applications and computers on a large network.

Regarding 802.11 networks, one of the interests of honeypots is to evaluate the wardriving[2] myth. Are they (malicious?) people seeking for open access points? Is it a voluntary association or an automatic connection setup? If I am a target, what are the attacker's motivations?

We must be aware that open 802.11 access points are quite common in company networks and when it is discovered it can lead to severe issues as the internal networks are generally wide open. This may be of interest for the attacker who wardrives around the enterprise physical location in order to check if open access points are present. The Wi-Fi network range can be improved thanks to special hardware (antennas with a higher gain and cards with a higher sensitivity and transmit power); thus, today it represents a big risk for any large company that cares about security.

[2] http://en.wikipedia.org/wiki/Wardriving: wardriving is the act of searching for Wi-Fi wireless networks by a person in a moving vehicle, using a portable computer or PDA.

4.4.2. *Requirements*

Honeypots are quite recent technologies and more specifically wireless honeypots. To be honest, wardrivers should not discover any wireless honeypots as today this is quite rare technology. It will be more interesting for the honeypot operator as the attacker would be fooled easier than in classic honeypot architectures. This is a strong advantage for defense, giving us the opportunity to retrieve more information regarding the tools and motivations of the wardriver.

Of course, when deploying honeypots, special care must be taken into account during the design and implementation of the architecture. The honeypot must not be used as an entry point for attacks toward other networks; moreover, the architecture should be not easily detectable. This is a technology dedicated to security, so the architecture must not weaken the overall security of the architecture if the honeypot is shared with an already deployed architecture. These requirements are critical for both the security of the architecture and achieving the best possible results (i.e. retrieving information from attacks).

4.4.3. *Design*

The goal is to deploy a dedicated architecture for wireless honeypots where all activities on the wireless side and deployed services (that give the attacker opportunities to connect to) are logged.

Using an open access point is interesting if we want to catch people looking for open access points. Another option should be to configure the WEP protocol in order to check if the attacker performs an attack (i.e. finding the shared secret used to protect wireless data communications) to access the wireless honeypot (and this is thus considered as malicious activity!).

The wireless honeypot must provide the attacker with a bunch of real services (some are mandatory as DHCP) and emulated services. We can also design a network topology by emulating different operating systems and services on a single honeypot in order to be both similar to a real network and to be as stealthy as possible. As a matter of fact, different techniques are available to the honeypot designer, and it is possible to emulate operating systems, routing, jitter, services, etc., in order to fool the attacker and thus learn more about his motivations.

Retrieving a large amount of information is a requirement for any honeypot architecture. By observing DHCP leases, DNS requests, etc., we could imagine the motivations of the attacker: is it only to gain Internet access or is it to retrieve confidential documents?

Wireless configuration

The wireless honeypot can be implemented in common hardware like a personal computer. You could use a wireless card and driver that supports the “master” mode which is basically the capability to act as an access point:

– have a 802.11 wireless card acting as an access point (i.e. “master” mode) with an adequate configuration regarding the properties of the protected network (choosing “ESSID” which is the network name, the channel and the MAC address of the access point);

– have another 802.11 wireless card acting as a monitoring node (i.e. “monitor” mode) which will be dedicated to listen to client probes (i.e. “Probe Request” frames).

For example, choosing the network name is really critical. It will be not possible to choose the same network name as those already deployed in your wireless environment. As a matter of fact, as you will not provide the same network access, if legitimate clients connect to your honeypot in place of the legitimate network, this will cause a major issue. This is typically the case for employees who connect to their company networks through IPsec over open wireless access points. If your honeypot uses the same wireless network name, your architecture will catch some legitimate users which is clearly not intended. The network name must be carefully chosen according to the wireless network environment in order to maximize the result opportunities and minimize the network issues.

Network and service emulation

Low interaction honeypots are one of the best choices for wireless honeypots. Of course, this is not a strict requirement, but thanks to a low interaction honeypot you reduce the overall risk of being compromised and you reduce the operational needs (i.e. observing and analyzing the results). Law interaction honeypots is the best approach regarding security, as emulated services should not be vulnerable to classic attacks. On the contrary, high interaction honeypots may have vulnerabilities dedicated to being exploited and thus analyzed by the honeypot operator (and then requiring more effort in terms of analysis).

Niels Provos’ HoneyD (http://www.honeyd.org) is certainly to most frequently deployed low interaction honeypot. Its features are numerous and its configuration is quite simple thanks to many templates on the author’s website. Thanks to this tool, you will emulate a consistent network architecture by configuring network addresses, routing, operating systems and application services. The attacker may be then fooled and the honeypot operator will concentrate themselves on the analysis of the attacker’s activity.

4.4.4. *Expected results*

Several data sources must be analyzed:

– logs from the "monitor" mode wireless card: looking at "Probe Requests" which is a clear sign of discovery wireless attempts;

– logs from the "master" mode wireless card: looking at all successful associations;

– logs from DHCP server: detecting configured DHCP clients;

– logs from DNS server: observing the DNS requests[3] from attached clients;

– logs from HoneyD: observing interactions with the emulated network and services.

Another option is to capture all network traffic and use specific tools to go further in the analysis. We could also take advantage of intrusion detection systems tools such as Snort (www.snort.org) in order to detect classic attacks in the IP world (not on the 802.11 layer).

Thanks to this honeypot design, we will be able to collect numerous data sources and then analyze them to provide interesting statistics. These statistics will provide us with strong facts about wardriving and potential malicious uses of open access points. This may be critical in evaluating the effective risks of wireless access in company networks.

4.4.5. *Conclusions*

802.11-based wireless honeypots are a low-cost option to observe potential malicious uses of open wireless access points. This is quite different from WIDS, but, it is considered as an additional source of information regarding attacks from the wireless side.

Even if honeypots – especially wireless honeypots – are not widely deployed and are much more dedicated to research, these technologies are valuable whenever you want to evaluate the real risks you are facing. The main drawback is related to manpower for deploying and operating the honeypot architecture. As a final note, we strongly recommend this paper from Spanish Honeynet: http://honeynet.org.es/papers/honeyspot/HoneySpot_20071217.pdf, which summarizes the architecture and needs when deploying a wireless honeypot.

[3] Requests from installed softwares (automated connections) or requests from the user who connected to the wireless honeypot.

Chapter 5

Multimedia Content Watermarking

5.1. Introduction

Watermarking represents a viable solution to persistently and transparently associate additional information (a mark) with original multimedia data.

Traditionally, these techniques were devoted to copyright protection. In such a scenario, the mark represents the legal owner identification and should be detected in any replica of the marked content which still has a certain commercial value (i.e. in the attacked data). The mark can have a relatively small size (a data payload of about 64 to 1,000 bits is frequently considered) but it should be detected even when strong, intelligent and unknown attacks are performed.

In addition to copyright protection, watermarking applications may be interesting for a large variety of applications: in-band enriched video, indexing and retrieval, personalized HDTV, etc. The data payload is now significantly increased (e.g. 1,000 times) but the robustness constraints are alleviated (the mark should be recovered just after some mundane operations, like a change of file format or compression, for instance).

As has already been noticed, the watermarking paradigm covers heterogenous applications, very often with contradictory aims and challenges. By presenting in a unitary way the building bricks of multimedia watermarking, this chapter allows the reader to explore the state-of-the-art scientific and technical achievements, and to identify the future trends in this effervescent field. The following structure has been

Chapter written by Mihai MITREA and Françoise PRÊTEUX.

adopted. Section 5.2 presents some illustrative watermarking examples and gives the related main definitions. Section 5.3 aims to identify the peculiarities of each type of data (still image, video, audio, 3D) from the watermarking challenge point of view. Section 5.4 is devoted to the watermarking theoretical framework. Section 5.5 is a discussion about the gap existing today between watermarking potentiality and its industrial implementation. Finally, the perspectives of watermarking applications within the emerging multimedia services are outlined.

5.2. Robust watermarking: a new challenge for the information society

In its largest acceptation [COX 02], [ARN 03], [DAV 04], watermarking means to imperceptibly insert some additional data (a mark) into a host media (a still image, an audio/video excerpt, a 3D object, etc.), according to a secret parameter. This information should be detected in any replica of the marked media, despite the malicious transforms it might have suffered.

5.2.1. *Risks in a world without watermarking*

The previous definition is very generic and can be considered as a key entry in a world of various potential applications which are now to be summarized.

5.2.1.1. *Copyright protection*

When considering the Information Society in general and the Internet in particular, art producers find themselves in a quite awkward position. On the one hand, a digital dimension is added as a completing element which gives art a whole new perspective. On the other hand, this very dimension opens the door to author spoliation: any piece of digital/digitized art (be it image, video, audio, 3D, etc.) can be replicated anytime and anywhere with a simple click. For instance [IIP 05], in 2004, DVD piracy amounted to $512 billion. When expressing this phenomenon in terms of markup [IIP 05], it turns out to be more "profitable" than cocaine traffic (Figure 5.1). The watermarking can play a very active role in restricting (virtually eliminating) this type of fraud.

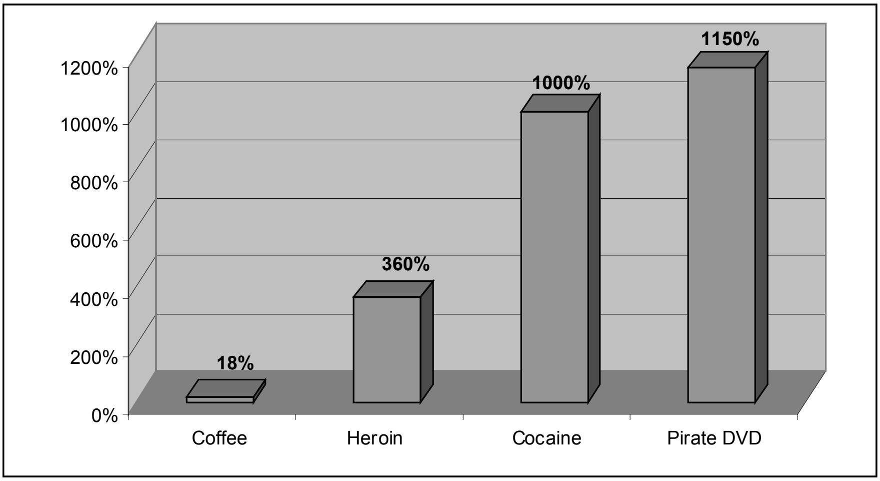


Figure 5.1. *Comparison of the black market markup for coffee, heroin, cocaine and pirate DVDs*

Consider a digital image, (Figure 5.2a) and assume there the case in which this image is sold on a digital support (Internet/CD/DVD, etc.). On the one hand, the true owner might be anxious because the buyer enters into the possession of an exact replica of the original. Consequently, the buyer may pretend he/she would be the intellectual property right owner for that image and may try to sell it again to somebody else, even at a lower cost (as he/she has no production costs to be amortized). Such a situation can be repeated as many times as a commercial interest in the image exists. On the other hand, even when differences between the original and that replica can be identified, it is very difficult to prevent author spoliation. Just for illustration, consider the case of the same image but with two different brightness levels or, even worse, the same image represented in two different file formats (jpg and ppm, for instance): although firm differences between the two files are encountered, it is difficult to ascertain which is the original and which is the replica.

In order to avoid such a situation [COX 02], [ARN 03], [DAV 04], some additional information connected to the author identity (e.g. a logo – Figure 5.2b) should be imperceptibly inserted into the original image itself (see Figure 5.2c). This logo should be detected in any replica which still has the same visual content as the original and/or in any part of the original image which still has any commercial value (see Figure 5.2d).

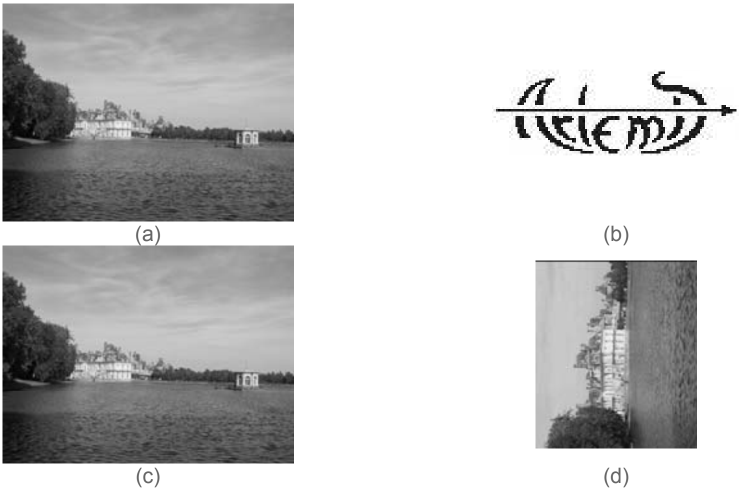


Figure 5.2. *Basic application of the watermarking mechanism: in order to protect the original image (a), the logo (b) is inserted, thus obtaining the marked image (c). This logo must be detected in any copy of (c), for example after rotation, conversion into a gray level representation and cropping of a region of interest (d)*

5.2.1.2. *Automatic control of multimedia data flows*

2006 began very well for commercial televisions in France [GIR 06]: the 118* phone number operators broadcasted the stream of the FIFA World Coup, thus resulting in huge benefits (e.g. of about €100 million in the first 6 months for M6).

In such a system, the client buys in advance some advertising time slots. However, he/she has no practical control either on the number if times or on actual time his/her clip is broadcasted. As suspicions always arise concerning an issue with such a budget, an automatic advertising broadcasting control system would very quickly find its place in the market.

Watermarking has already proved its efficiency in this respect [COX 02]. The principle is to imperceptibly insert in each advertising clip the owner ID. Then, an automated detector can continuously watch the TV channel and detect when and for how long the ID can be identified.

5.2.1.3. *Enriched multimedia*

The previous section is an example illustrating how an automatic watermarking detector may very easily and efficiently parse an additional stream of data inserted in the main data stream. Of course, there is no constraint, either concerning the syntax or the semantic of this additional stream: it can represent any type of enrichment information inserted at the producer side. The corresponding application field is very large, including automated archiving and indexing, automatic identification of a user profile, interactivity information, etc.

Figure 5.3 represents the example of a video sequence enriched with the interactivity information (a pinball game) by means of a watermarking technique [MIT 06-01], [TRA 03]. In such a scenario, the user can watch the movie in a very small round window which enlarges according to the points the user scores. Such an application would require about 6 kbytes of extra data which can be very easily inserted in any video sequence. Note that the application in Figure 5.3 does not require either additional transmission channel or changes in the video format standard.

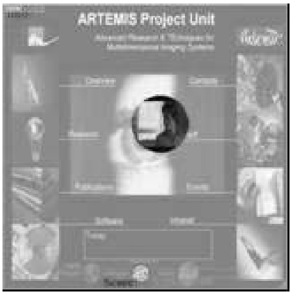


Figure 5.3. *Video enrichment by watermarking techniques: interactivity applications*

5.2.2. *Watermarking, steganography and cryptography: a triptych of related, yet different applications*

As watermarking reliability intrinsically depends on a secret parameter (on a key), it is a type of secret communication. While section 5.4 will present the proper secret communication model for watermarking, the this section identifies the watermarking relationship with two other types of secret communications, namely steganography and cryptography.

5.2.2.1. *Watermarking versus steganography*

Let us consider the text in the table below. Its meaning becomes completely different when reading every other line.

This is just a basic example of steganography: the message to be transmitted is inserted according to a secret algorithm into a host media. In contrast to watermarking, this message should be secret and, generally, it can no longer be recovered when strong alterations are encountered.

Others examples of steganograhy include the changing of the less significant bit in each pixel of an image, adding an un-audible echo in audio signals, etc.

My boss is always
working hard in his office, without
losing his precious time. He never
lets down the colleagues needing his help although he always
finishes his projects in the due time. Very often he expands his
working hours, sometimes completely skipping over
lunch breaks. He is the type of person who has no
vanity despite his outstanding results and his remarkable
computer skills.

Table 5.1. *An apparently flattering text…*

5.2.2.2. *Watermarking versus cryptography*

Cryptography may also be considered as a possible solution for copyright protection. Let us take the example of a Pay-TV broadcasting system. Prior to transmission, the TV signal is encrypted at the transmission side (i.e. it is transformed into a visually meaningless signal, by applying a known one-to-one function which depends on a secret parameter). The users who pay a subscription know the secret parameter and can decrypt the received signal and watch it on a TV screen. They may also be capable, with quite mundane equipment (a TV tuner and a home computer), of storing the decrypted TV signal on a hard disk and trying to further benefit from it. Actually, this is quite a common situation nowadays: the movies broadcasted over Pay-TV are further distributed via file sharing systems. Such a situation is possible because, in contrast to watermarking, at the decryption side the user enters into the possession of an exact copy of the original signal.

5.2.3. ***Definitions and properties***

While keeping an application-driven point of view, this section presents the proper definitions connected to the main features of watermarking [COX 02], [ARN 03], [DAV 04]: data payload, transparency, robustness, and probability of false alarm. A particular trade-off among these properties should be reached when designing any particular watermarking method.

5.2.3.1. *Data payload*

Data payload represents the total amount of information (expressed in bits) which is inserted into original content. According to the targeted applications, the requirements in this respect are very different. The earliest watermarking techniques inserted just 1 bit, i.e. a marked/unmarked decision used to be made after the detection. For copyright applications nowadays, it can be considered that a data payload between 60 and 70 bits would ensure the basic functionalities (e.g. the insertion of a serial number). More elaborated digital rights scenarios (e.g. the insertion of a visual logo, additionally setting special user rights like copy once, read many times) may increase data payload up to 1,000 bits per video/audio sequence. When shifting toward enriched media applications, the data payload is significantly augmented. Actually, the watermarking potentiality in this direction is restricted by the data payload. For instance, automatic video indexing may require the insertion of 1 bit per frame, which may be easily done. However, for interactive digital TV (iDTV), about 100 bits should be inserted in each frame, which is not feasible when strong attacks are applied to the enriched content [MIT 06-01].

5.2.3.2. *Transparency*

Transparency is connected to the human perception of the artefacts induced by the mark insertion. In this respect, fidelity and quality are defined. A watermarking method features transparency when no perceptual differences are identified between the marked and unmarked content (e.g. no visual differences in video or no audible differences in audio). A watermarked content features quality when the artefacts, although noticeable, are not disturbing for the human observers. Note that the quality property refers to the watermarked product and not to the watermarking procedure itself: no difference is made between the artefacts already presented in the unmarked content and those induced during the marking procedure. In other words, when applying a watermarking procedure featuring fidelity to very low quality content, the watermarked content will be also of a very low quality. As it can be seen, transparency is a subjective notion, its evaluation requiring many human subjects, with different professional backgrounds, artistic skills and of different ages. Moreover, the testing procedure is very complex and depends on many factors: particular noise/light conditions, original content peculiarities, etc. Consequently, solutions for objective and automatic evaluation of transparency have been sought. Taking into account that basically the same impediment is encountered in compression applications, several metrics inherited from this field are widely used in watermarking: SNR (signal-to-noise ratio), PSNR (peak signal-to-noise ratio), UIQI (Universal Image Quality Index [WAN 02]) and DVQ (Digital Video Quality [WAT 99]). However, the limitation is the same: these metrics lead to automatic rather than objective evaluation. For instance, depending on the original content or on the inserting procedure, the same numerical value for the SNR may correspond to transparent or perceptible artefacts.

5.2.3.3. *Robustness*

Robustness refers to the capability of recovering the embedded message from any replica of the marked product which still has a certain commercial value. Generally, for a watermarking application, two types of transformation are applied to the marked product. First, there are mundane operations any multimedia product suffers in its day-by-day usage: compression, change of file format, cropping, etc. The second class is represented by attacks, which are defined as malicious transforms specifically designed in order to turn the mark undetectable while keeping the same quality for the multimedia content. As these two classes are equivalent from the technical point of view, both of them will further be referred to as attacks.

For copyright applications, the mark should be recovered after a large variety of potential attacks; hence, robustness becomes a strong requirement. A total robustness would mean resistance against all the present and future attacks, which is unfeasible in practice. However, the force of the attacks is also restricted by the artefacts they induced: any malicious user aims at fading out the mark without decreasing the commercial value of the attacked product.

5.2.3.4. *False alarm probability*

The probability of false alarm measures the probability of mistaking unmarked content for marked content. Of course, the authorized users would never accept a practical watermarking system unless its probability of false alarm is arbitrarily low (e.g. lower than 10^{-10} or even lower). This limit should be evaluated using theoretical reasons. Note that for such a small probability, an accurate Monte Carlo estimation would require too much time to be performed.

5.2.4. *Watermarking peculiarities in the mobility context*

Ten years ago, the current capabilities of a mundane cell phone could have appeared unrealistic: in addition to voice services, mobile networks allow their clients to surf the Web, to stream on-demand video, to watch live TV or to play online games. Just like computers, cell phones seem to have no more restrictions concerning the media types that can be played, stored and processed. However, important hardware differences still exist. First, the bandwidth constraints are far more restrictive for mobile than for computer networks. Secondly, the memory and computational resources are still poor for cell phones, at least when compared them to a PC.

While the applications are basically the same, this resource difference determines the peculiarities of the watermarking for mobile phones.

First, the multimedia content is generally represented at lower qualities. Consequently, there is somehow less room to embed the mark and, therefore, there are stronger data payload constraints.

Concerning transparency, the problem is somehow alleviated: generally, the user is far less disturbed by the artefacts on a cell phone display than on a home cinema system.

Regarding robustness, it is generally accepted that the requirements are quite similar.

5.2.5. *Conclusion*

Digital watermarking aims to transparently and persistently associate some additional information with an original multimedia content. While traditionally this additional information was exploited for copyright assessment, nowadays the watermarking application area is likely to testify a rapid expansion.

To design a watermarking method means to reach the trade-off between data payload, transparency, robustness and probability of false alarm, according to the particular application to be addressed.

Although connected both to cryptography and steganography, watermarking is a well defined science. Cryptography can protect any digital content during transmission, but can leave it completely vulnerable after decryption. Concerning steganography, it can be considered as a watermarking limit case, in which the insertion algorithm is kept secret, the robustness is neglected and the data payload is drastically increased.

5.3. Different constraints for different types of media

5.3.1. *Still image and video, or how to defeat the most daring pirates*

This section summarizes the main issues connected to still image and video protection. This is basically the watermarking application field, mainly due to the huge economic benefit related to it. Consequently, every step made by watermarking technique developers is followed by one step made by pirates.

5.3.1.1. *Transparency*

It can be ascertained that the watermarking approach to still image/video transparency is inherited from compression techniques. Two mains issues connected to transparency are discussed in the literature.

Generally, instead of subjective evaluation, requiring many human observers, some similarity metrics between the original and the marked content are computed. The values thus obtained are compared to some widely accepted limits, and a decision about the method transparency can be made.

For instance, for still images, we may mention the SNR, the PSNR, and the UIQI. Denoting by $x=\left\{x_i \middle| i=1,\ 2,\ldots,\ N\right\}$ and $y=\left\{y_i \middle| i=1,\ 2,\ldots,\ N\right\}$, the original and marked content (in a vectorial representation), the definitions are the following:

$$SNR = 10\log_{10}\frac{\sum_{i=1}^{N} x_i^2}{\sum_{i=1}^{N}(y_i - x_i)^2}, \quad PSNR = 10\log_{10}\frac{N\max\limits_{i} x_i^2}{\sum_{i=1}^{N}(y_j - x_i)^2},$$

$$UIQI = \frac{\sigma_{xy}}{\sigma_x\sigma_y}\cdot\frac{2\overline{x}\overline{y}}{(\overline{x})^2+(\overline{y})^2}\cdot\frac{2\sigma_x\sigma_y}{\sigma_x^2+\sigma_y^2},$$

where:

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i, \quad \sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2, \quad \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \overline{y})^2,$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y}).$$

The second issue connected to transparency is perceptual shaping. This means to exploit the HVS (Human Visual System) peculiarities in order to better conceal the mark [MIL 04].

For video, the same types of measures are extended and/or new ones are defined; for instance, the study in [DUT 08-01] presents a detailed discussion on the transparency of watermarking methods for the MPEG-4 AVC stream.

5.3.1.2. *Robustness*

The wide array of watermarking attacks available for video can be categorized in four large classes, according to the way in which they act in order to prevent the mark from achieving its purpose: removal, geometric, cryptographic and protocol attacks [VOL 01].

Removal attacks do not attempt to crack the watermark security, but instead they try to render the information conveyed by the watermarking unreadable, regardless of the complexity of the watermarking system and the processing employed for watermark detection. This category includes denoising, quantization, remodulation and collusion attacks:

– Denoising attacks take advantage of the noise-like effect of the watermarking process. By applying a noise removal filter, an attacker hopes to remove the watermark signal as well. This operation uses the knowledge on the original (unmarked) signal statistics in order to optimize the denoising operation.

– Quantization attacks are actually techniques often employed in compression. These techniques use human visual system peculiarities as well as statistical properties of the original signal in order to eliminate as much information as possible, while keeping an acceptable visual quality.

– Collusion attacks [DOE 05] use several copies of the same video, marked with different watermark signals (either different messages or different keys are used for obtaining the signal). The attacker can then remove the mark by averaging among the available videos, by using only small portions of each video when creating the attacked version, etc.

Geometric attacks do not try to remove the mark, but rather try to destroy their synchronization. Thus, after such an attack the mark is still present in the video, but its location is unknown to the detector. This type includes rotations, bending (global de-synchronizations) and pixel jitter (local de-synchronizations) all present in the StirMark package. These attacks model the real-life camcorder scenario: the mark should be recovered from a movie which is captured with a hand camera in a theater. In theory, the watermark can be recovered if the synchronization is regained. This is usually done by complex synchronization schemes, which are not always practical because of the high complexity of the process. Another way to defeat these attacks is by employing invariant transforms (e.g. the Melin-Fourier transform for robustness against rotations).

Cryptographic attacks consist of trying to eliminate the mark without the knowledge of the key. It has been shown [COX 02] that with the knowledge of the embedded mark, its removal is trivial. One approach would be a brute-force search for the embedded information. Another approach, known as the Oracle attack, can create an unmarked version of the signal, assuming the pirate disposes of a detector. This type of attack is, however, very restricted due to its computational complexity.

Protocol attacks do not attempt to remove the mark or to render it unreadable, but to make it useless by creating some sort of ambiguity. This class of attack includes inversion and the copy attacks. Inversion attack creates a fake key so that when applying the detection procedure, the mark would indicate another owner for the content. Copy attack does not try to impair the detection or use of the mark, but rather it estimates the mark from the protected video in order to copy it to another, unmarked video.

Another classification [DOE 05] divides attacks into generic and specialized. Generic attacks are seen as the transformations applied to the video without any knowledge of the watermarking system, while specialized attacks correspond to the case in which the malicious user gathers as much information as possible about the watermarking system in order to remove the mark or render it unusable. The first category roughly corresponds to removal and geometric attacks, while the second corresponds to cryptographic and protocol attacks. Note, however, that in this case collusion attacks belong to the specialized category and are considered by the authors [DOE 05] as a cryptographic attack.

5.3.1.3. *Embedding technique*

An exhaustive state of the art on the watermarking embedding techniques is no longer possible: in just 20 years, the variety is so great and the details still so important in the overall performance that nobody can claim to know everything in this respect.

However, this section aims to present some main directions along which the majority of these techniques can be structured. Watermarking embedding techniques will be further classified according to the space in which the mark is inserted: spatial techniques (the mark is inserted directly into image/video frame) and frequency techniques (the mark is inserted into the coefficients of a certain transform applied to the original image/frame).

The earliest watermarking techniques belong to the former category. At a glance, they are intuitive and feature low complexity but lack robustness and generality. Some examples are as follows:

– *LSB (Least Significant Bit) modification:* the easiest way to insert a mark is based on idea that the LSB data are insignificant. Consequently, these bits can be replaced by a mark. Note that these methods have almost no robustness, and thus belong to steganography rather than to watermarking.

– *Information tagging:* here the principle is to add *tags* (small geometric patterns) to digitized images at brightness levels that are imperceptible to human eyes.

– *Quantization noise embedding:* the idea of this technique is based on the fact that quantization noise may be imperceptible to human eyes. First a watermark is injected into an image using a stream data to guide level selection in a predictive quantizer.

– *Statistical techniques* can be exemplified by the *patchwork* method. This method chooses *n* pairs of image points (a_i, b_i) and increases the brightness at a_i by one unit, while correspondingly decreasing the brightness of b_i. The sum of the differences of the *n* pairs of points should then be $2n$.

The frequency domain watermarking class contains practically all the methods nowadays under consideration. For the same embedding technique, the overall performances depend on the considered transform.

– The *discrete cosine transform* is perhaps the most intensively used transform in watermarking. Its near-optimal energy compaction property already imposed it as the core of many advanced compression techniques (e.g. MPEG compression). When considered for watermarking applications, the DCT results in very good robustness and transparency.

– The *discrete wavelet transform* is also very often considered for watermarking applications. The core of the JPEG 2000 compression standard, this transform has as a principal advantage its linear computation complexity. Nowadays, it is not clear whether DWT will take the watermarking leadership over the DCT.

– The *Mellin-Fourier transform* has excellent properties regarding the invariance with respect to rotations, but lacks robustness with respect to other types of image processing techniques.

– Several other types of transforms have been considered (like the Hough transform, singular value decomposition, etc.) but their utility is restricted to particular scenarios.

5.3.2. *Audio: the highest constraints on imperceptibility*

It is generally accepted that the HAS (human auditory system) is far more sensitive than the HVS (human visual system), i.e. for the same signal-to-noise ratio a perturbation is more unpleasant in an audio signal than in an image. Practically all the studies reported in the literature agree in this respect: the most restrictive constraint in audio watermarking is the transparency. On the other hand, this sensibility also reduces the force of attacks.

Consequently, the presentation in this section will be structured with respect to audio transparency and robustness peculiarities, and then some basic insertion techniques will be summarized.

5.3.2.1. *Transparency*

The study in [ARN 02] is completely devoted to quality evaluation for audio watermarking techniques applied to both high and low rate audio.

5.3.2.1.1. Subjective evaluation of transparency

A watermarking method features transparency when no audible differences can be perceived between the marked and the unmarked audio data [ARN 02]. The ultimate evaluation of the transparency is intrinsically subjective: it requires many human observers, of different ages and with different professional backgrounds, with various levels of knowledge about music. In order to enable a reliable evaluation of this property, a statistical test (called *the two alternative forced test*) was considered.

According to this testing procedure, a number of n pairs of the type (*audio1, audio2*) are randomly selected from the set of all possible combinations {(*original, original*), (*original, watermarked*), (*watermarked, original*), (*watermarked, watermarked*)} . The human observer is asked about the perceptual identity between $audio1$ and $audio2$, and each correct answer is considered a success (a *hit*). For transparent watermarking, the k number of correct answers follows a binomial law of n and $p = 0.5$ parameters. The critical region can be easily determined for a chosen α significance level. The type II statistical error probability can be also computed. It should be noted that: the probability of this type of error depends drastically on the sample size.

5.3.2.1.2. Objective evaluation of transparency

As can be very quickly noticed, such an evaluation would require a lot of time and money. Consequently, some objective measures of the artefacts induced by the watermarking procedure have been sought. The solution was identified in the field adjacent to audio compression.

In order to cut down both the time and money required by subjective tests, objective approaches try to model the listening behavior of human beings. Note that these solutions differ according to the data rate: for bitrates larger than 64kbit/s the

ITU-R BS.1116 and ITU-R BS.1387 (PEAQ – Perceptual Evaluation of Audio Quality) should be followed.

5.3.2.2. *Robustness*

The robustness requirement for a watermarking algorithm is highly dependent on the targeted application. For example, a simple insertion of auxiliary data for a covert communication would not require very high robustness. However, any manipulation of the cover data could result into an attack that would destroy the mark [STE 01]. This is an issue especially in the case of audio data, where such manipulations are very common and widely available. For instance, lossy compression (MP3, AAC), equalization and normalization are already mundane operations. The problem is even more serious in the case of copyright protection watermarking, as the watermark should also withstand malicious attacks.

A classification of the audio distortions (attacks) that may appear during normal or malicious processing is made [STE 01] according to the way in which the audio data are manipulated:

– *Dynamics attacks* are changes in the dynamic range (*loudness profile*) of the audio signal. This category includes the linear (e.g. the increasing and decreasing of its dynamic range) and non-linear (e.g. limiting, expansion, compression) modifications of the audio signal.

– *Filtering* is the cutting/increasing of the amplitude of certain parts of the spectrum. Equalizing (increasing or decreasing, instead of simply cutting off, certain parts of the spectrum) can also be considered filtering.

– *Ambience modifications* consist of introducing effects that recreate a certain ambiance by simulating the presence of a room, concert hall, etc. The most common effects are echo (or tape delay) and reverb (simulating multiple reflections of the same sound).

– *Conversion attacks* consist of data representation modifications, like the mixing of stereo audio data into mono, sample frequency modification (associated with filtering) and sample size modification (re-quantization – e.g. from 24 bit samples to 16 bit samples).

– *Lossy compression* of the audio data, such as MP3 and AAC, is also an example of very common user-available transformations. These compression techniques take advantage of the psycho acoustic models of the human auditory system in order to eliminate unnecessary data from the audio signal.

– *Noise addition* can be an effect of most of the attacks described above. However, it can also appear independently, induced by hardware components.

– *Modulation attacks* like vibrato (rapidly and repeatedly raising and lowering the frequency of a single note – this effect is used to add expression to instrumental notes), chorus (producing several sounds with the same timbre and nearly but not exactly the same pitch, in order to induce a shimmering effect for a single note), amplitude modulation and flanging (mixing two versions of the same signal, one of them having a small – smaller than 20 ms – and slowly changing time delay with respect to the other) are very rarely used in user-level post-production. If they occur, they are most likely to be attacks intended to remove the mark.

– *Time stretch and pitch shift* are changes in the length and pitch of the audio. They are used for fine-tuning an audio piece, fitting audio parts into fixed time windows, synchronizing the beats of two consecutive audio parts, etc., or they can appear as an unintentional side-effect of analog editing. These modifications, as well as signal cropping, are considered as an equivalent to geometric distortion attacks in image watermarking.

– *Sample permutations* are distortions unused in common audio processing. Thus, they can be considered as attacks designed to render an embedded watermark unusable. Apart from sample permutation, sample dropping (cutting out random-chosen samples), zero-cross-insertion (inserting multiple zero-valued samples at the zero-crossings of the signal) and other similar approaches fall into the same category.

Additionally, the mark should also be recovered after A/D and D/A conversions, e.g. microphone recording of an audio CD track.

All these attacks are very different from the audibility point of view. When attacking an audio excerpt, a would-be pirate is restricted by the quality of the resulting signal. In this respect, a subjective listening test is performed [STE 01] for some of the above-mentioned attacks and for different audio content (speech, classical, jazz, pop-rock music and urban-specific noises). The results are described as very different with respect to both attack and audio content. Considering the former criterion, as a general rule, the more efficient the attack in removing the mark, the lower the quality of the resulting content. Consequently, the practical applicability of the most harmful attacks (pitch shifting, sample copying and cutting, delay, enhancement, sample flipping, de-hissing and flanging) is drastically limited. When considering the latter criterion (the original audio content), louder and noisier content (such as urban noises or even pop-rock music) can cover the distortions introduced by the attacks much better than quieter content, such as spoken text and classical music.

5.3.2.3. *Embedding technique*

According to the embedding method, the audio watermarking techniques can be divided into five categories [KIM 03]: *quantization, spread-spectrum, two-set, replica* and *self-marking schemes*.

Quantization modulation schemes insert the watermark bits by sample quantization. The basic watermarking method is the following: given a sample *x*, it is quantized by using a D quantization step. Further on, a data bit is inserted either by adding (in the case of a "1") or by subtracting (in the case of a "0") a quantity *D*/4 from the quantized sample. In order to detect the embedded message, the inverse operation is performed: the quantized value is subtracted from the received sample [LIU 04]. If the difference is positive, the received bit is considered to be "1", otherwise it is considered to be "0". The data recovery is successful only when the added noise amplitude is below *D*/4.

Spread spectrum schemes are based on correlation detection [KIM 03]: basically, a pseudo-random sequence is inserted into the audio data and the watermark is detected by computing the correlation coefficient between the received signal and the pseudo-random sequence [COX 02], [MAL 04], [KIM 04], [HE 05], [STE 05], [LOB 03]. These methods have received a great deal of attention from researchers: by spread spectrum modulation, the data is spread, thus becoming a noise-like signal with very low power in the frequency bands. This noise can then be added to the signal without introducing noticeable distortions. In audio watermarking applications, however, due to the sensitivity of the HAS, audio masking techniques have to be employed.

Two-set embedding schemes are based on creating or modifying differences between sets of data belonging to the same original signal. For example, considering two audio data blocks, modifications can be introduced in the difference between the two block statistical properties, like the difference of means or variances (*patchwork schemes*), or in the overall block energies (*amplitude modification*). Out of these two approaches, patch algorithms are more popular, and they have reached robustness against echo addition, filtering and lossy compression [YEO 03], [CVE 03-01], [CVE 04-01].

Replica embedding schemes use the very signal to be protected as the audio watermark. The signal itself is modulated and then embedded either in the time or frequency domain. The advantage of this scheme is that the detector can calculate the watermark signal starting from the received signal, thus rendering the method very robust against de-synchronization attacks. Two watermarking approaches can be identified in this category [KIM 03]: *echo hiding* and *replica modulation*. When employing the former, the mark is inserted by adding a replica of the signal with a certain delay (time domain shift) and modulating this delay [KO 05]. When employing the latter, the signal replica is shifted in the frequency or phase domain [PET 01].

Self-marking methods embed the message into the original data by creating *self-evident* modifications [KIM 03]. In this respect, the time scale modification algorithm [MAN 01] embeds a message into the audio data by stretching or compressing the time scale between consecutive local extremes of the original audio data. Note that this type of data embedding is also a challenging attack for most watermarking schemes.

5.3.3. *3D data: watermarking versus heterogenous representations*

In contrast to audio/video signals, for a 3D digital object there is no objective representation. Consequently, the main difficulty in 3D watermarking is derived from the fact that the same object may have different types of representations. Just to exemplify, a 3D object may be represented as a mathematical equation or as a set of 3D points sampled from it.

As such, this chapter will be structured according to the way in which the 3D objects are represented.

When approaching 3D watermarking, many studies try to take advantage of the results already obtained for image/video watermarking: they first derive some virtual images from the original 3D model and further apply a 2D watermarking method to these images; we shall further classify these 2D/3D watermarking methods.

5.3.3.1. *NURBS surfaces*

Nowadays, NURBS (Non-Uniform Rational B-Splines surfaces) may be considered as the de facto standard for representing, designing and data exchanging geometric information [PIE 97]. Under a unified framework, NURBS makes it possible for both analytic shapes and free form entities to be represented. Moreover, NURBS algorithms are fast and numerically stable. Finally, the design with NURBS is intuitive. A NURBS surface is defined by means of:

– a set of control points which should be approximated by the surface;

– two knot vectors which determine the influence of the control points on the surface;

– a set of weights which somehow establishes how close the surface is to the control points.

When dealing with NURBS representations, the main advantage is the availability of a natural 2D surface parameterization which makes it possible to extend the 2D watermarking techniques to 3D data. The study in [LEE 02] seems to be the first approach to exploit the NURBS parametric definition in this respect. Two methods are there proposed: the first for steganography and the second for watermarking.

For the latter method, three virtual images are obtained by uniformly sampling the 3D model and by recording the *x*, *y* and *z* coordinates thus obtained. It was applied to 2 models (sampled as 128×128 and 64×64 points) and proved itself robust against control point modification by affine transformations, knot vector modification and surface approximation. The probability of a false alarm was evaluated as being lower than 10^{-7}. Moreover, the method performances seem to depend on the 3D model size. Finally, notice that this method is computationally complex.

A different approach is presented in [MIT 04]. The 3D object is represented by three virtual images derived from the control points. The mark is generated starting from a 64 bit message and is embedded by means of a spread spectrum technique in the DCT coefficient hierarchy. The method features good performance in terms of transparency and robustness. Its main weak point is the data payload. When the 3D objects are represented by very few control points, there is less room for mark embedding. While keeping unchanged the robustness constraint, the 64 bits cannot be inserted into an individual 3D object. Instead they should be spread over a set of such 3D objects. In some extreme cases, this set should be very large (e.g. 1,024 objects).

5.3.3.2. *Extended Gaussian Image (EGI) – Complex Extended Gaussian Image (CEGI)*

The oblivious method developed by Benedens [BEN 99] employs the model EGI or its discrete version, the orientation histogram. The variables to be modified were the normals of the object surfaces and their orientations. The method features robustness against model simplification and noise addition. In [KWO 03] and [LEE 03], the method is improved against the cropping attack by computing the EGI on some model patches and not on the entire model. The inserted data quantity is about 50 bits. However, the method is still vulnerable to model rotations and re-meshing. Another upgrade is brought in [LEE 05], by employing the CEGI of the object patches. This makes it possible to increase the data payload to 120 bits. Robustness against rotation is achieved by object realignment. Note that some other watermarking algorithms [DAR 04], [ZAF 04], [KAL 03] also use the object pre-alignment in order to render the mark robust to the rotation and translation.

Let us further go into detail. In order to compute the EGI, the following steps are performed:

– One normal is computed for each surface element. Note that any surface can provide normal vectors on two directions. However, the one which yields the greatest value for the angle between the normal and the line connecting the object mass center to the surface center is selected.

– The length (Euclidian norm) of the normal vector is proportional to the size of the corresponding surface element.

– The CEGI considers a complex value as a normal vector length: the module of the value corresponds to the surface area and the phase corresponds to the distance from a designated origin to the surface.

Further on, the normals are sampled into bins and one bit is embedded in each bin. In this respect, the following steps are performed:

– For each bin a normal is selected as a bin center. The normals that form angles with this bin center smaller than a threshold are assigned to the bin.

– The mean normal of each bin is computed.

– One or more feature elements are defined on each bin, in order to hold the mark. Originally [BEN 99], the feature element was the position of a center of mass, defined as the projection of the bin normal mean on a plane given by the bin center. In [LEE 05] the feature element is the angle formed by the mean normal of the bin with the bin center.

– The feature element of each bin is modified so as to encode one bit of data. In the case of the center of mass, it is displaced so as to fall into a certain region of a circle defined around the bin center. In the case of the angle between the normal and the bin center, the angle is modified in order to have a certain value θ_0 encoding a "0", or θ_1 encoding a "1".

By considering a set of normals, corresponding to multiple polygons, instead of just one, the robustness against simplification up to a certain level is achieved. However, these methods are not robust against re-meshing, as this operation can modify both the mean normal and the center normal of the bins.

5.3.3.3. *Spherical harmonic transform*

In [LI 04] the object is mapped into a sphere, then the mark is embedded into some spherical harmonic transform coefficients. In this respect, the model is first simplified in order to obtain a convex model – the *base model*. The deleted vertices are recorded as simplification parameters. The remaining vertices are projected onto a sphere, with the center situated inside the convex model. Further on, a vertex split is performed, in order to map the previously deleted vertices onto the sphere, by observing the previously recorded simplification parameters. With all the vertices mapped onto a unit sphere, the mesh can now be described by a function $f(\varphi,\theta)$. Further on, these functions are sampled into a grid of 64×64 points and the spherical harmonic transform $\hat{f}_t(l,m)$ is computed, so that $f(\varphi,\theta)=\sum_{l=0}^{\infty}\sum_{m=-l}^{l}\hat{f}_t(l,m)Y_l^m(\varphi,\theta)$, where $Y_l^m(\varphi,\theta)=k_{l,m}P_l^m(\cos\theta)e^{im\varphi}$, $k_{l,m}=\sqrt{\dfrac{2l+1(l-m)!}{4\pi(l+m)!}}$ and $P_l^m(x)$ is the associated Legendre polynomial of l degree and m order. The watermark bits, represented as $w[k]\in\{-1,1\}$, are embedded into a part of the spherical harmonic transform coefficients by simple addition.

The mark embedded according to this algorithm is immune to noise addition, object translation, rotation, scaling, re-sampling, filtering, enhancement and simplification. The quantity of inserted data is 24 bits. However, the transformation is computationally complex and the mark detection requires the original sphere mapping information. Furthermore, the method does not withstand object re-meshing.

5.3.3.4. *Range image*

The range image embedding methods [SON 02], [BEN 05] employ a virtual range image obtained by cylindrically scanning the object. Originally, a range image representation of a 3D object encoded the distance of the sample points of the object from a reference plane as the image pixels. In the case of the cylindrical range map, the reference plane is the surface of a cylinder built around the 3D object and the distances are defined along the cylinder radius.

In order to obtain the virtual image corresponding to the mesh, the method defines a cylinder positioned around the 3D model so as to fit it tightly. Further on, a grid is defined on the cylinder surface. Each image pixel corresponds to a grid element; its value is the radial distance from the grid element to the model. The message is embedded into this 2D representation by means of an SS image watermarking technique. The new, marked object is then recomputed, starting from the marked virtual image.

The method performances greatly depend on the considered image watermarking technique. Moreover, it is computationally complex, if we take into account the operations needed to transform the object into and from a range image. However, the method can be immune to common mesh transformations (re-meshing, simplification, vertex reordering, etc.). The method described in [SON 02] is non-oblivious and the quantity of inserted data is only 11 bits. In [BEN 05], two image watermarking techniques are employed: one that allows only a marked/unmarked decision and is non-oblivious, and another that allows the embedding of 64 bits and is oblivious. The watermark is protected against rotation, translation and scaling attacks by performing a supplementary step in order to reposition and normalize the object before watermarking or extracting the mark.

5.3.3.5. *Hologram*

In [KIS 03] a 2D representation, somewhat similar to the range map, is used for the 3D object, namely the digital hologram. In this case, the object is represented by a complex hologram function, $H(x,y) = A(x,y)\exp(j\phi(x,y))$. As opposed to the range map, this type of representation contains not only object structure data, but appearance (texture) data as well.

The embedded watermark, in this case, is not a binary message, but another hologram representation of a 3D object. In order to do this, the double-phase encoded signal of the hologram is computed:

$$H_d(x,y) = \{H(x,y)\Psi_1(x,y)\} \otimes IFT[\Psi_2(\xi,\gamma)],$$

where $\otimes$ is the convolution operator, *IFT* is the inverse Fourier transform, (ξ,γ) are coordinates in the frequency plane and $\Psi_1(x,y) = \exp[j2\pi b_1(x,y)]$, $\Psi_2(\xi,\gamma) = \exp[j2\pi b_2(\xi,\gamma)]$, b_1 and b_2 being two random matrices, with elements obeying an uniform distribution in the $[0,1]$ interval.

It can be shown that the double-phase encoding of a complex signal produces a Gaussian noise-like output.

In order to watermark the original object, the hidden object is double-phase encoded, scaled, and then added onto the original. The watermarked original object is then double-phase encoded for transmission.

At the detection side, the inverse transformation is simply performed. The inverse of the double-phase encoding has the same effect on an uncoded hologram: it transforms it into a noise-like signal. Thus, when decoding the hidden object, it is distorted by a Gaussian noise signal, representing the host object.

The method features transparency and good robustness against general and hologram-specific attacks: noise addition, quantization, cropping, and hologram occlusion.

Moreover, the hidden object cannot be recovered without the knowledge of both b_1 and b_2 matrices, which represent the (secret) key.

5.3.3.6. *Conclusion*

The peculiarities of each type of media with respect to watermarking applications are synoptically displayed in Table 5.2.

	Still images and video	Audio	3D
Data payload	At least a serial number		A few bits
Transparency	Objective measures (e.g. SNR) Measures based on perceptual models of the human visual and auditive systems		No representation-independent measure has yet been identified
		Human auditive system is very sensitive, thus constraining the transparency	
Robustness	Analog hole attacks (in-theatre camcorder video recording, microphone recording of audio tracks, etc.)		No analog hole exploitation is possible yet
	A wide array of developed attacks	Attacks are highly restricted by the sensitivity of the human auditive system	Change of representation

Table 5.2. *Media type peculiarities from the watermarking point of view*

Upon investigating this table, two problems arise:

– Despite these peculiarities, does a unitary theoretical model for watermarking exist?

– Is it possible to design a unitary watermarking method that is equally good in protecting video, audio and 3D data?

Both questions have positive answers, which will be presented in section 5.4.

5.4. Toward the watermarking theoretical model

5.4.1. *General framework: the communication channel*

From the theoretical point of view, a generic watermarking system can be modeled as a noisy channel (Figure 5.4). The copyright information is a sample from the information source and should be recovered at the detection side (i.e. it should be detected in the marked document). The elements that make watermark detection difficult can be modeled as the channel noise: the original document, the mundane processing and the attacks. The watermarking procedure itself plays the role of the modulation technique (i.e. the way in which the mark is inserted into the host document).

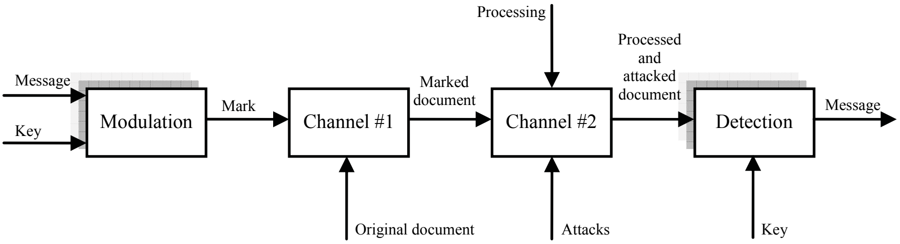


Figure 5.4. *A watermarking method as a noisy channel*

From the communication theory point of view, transparency and robustness are antagonistic requirements.

When considering transparency, the watermark is the noise that is added onto the host signal, thus altering the user's multimedia experience. In such a situation, a very high signal-to-noise (i.e. host to watermark) ratio is desired (e.g. larger than 30 dB). From the robustness point of view, the watermark is the signal whose detection is impaired by the noise (the original content and the attacks). Consequently, it is desirable to have a large SNR value when the watermark is the signal and the host document is the noise.

5.4.2. *Spread spectrum versus side information*

The two main mathematical frameworks in which watermarking can fit lead, on the one hand to the spread spectrum (SS) communication theory and on the other hand to the side information theory (SI).

SS methods have already been used in telecommunication applications (e.g. CDMA) providing a good solution for very low power signal transmission over noisy channels. An SS technique uses for the transmission the largest bandwidth available. Consequently, an SS method will spread the mark across the host media, occupying a much larger bandwidth than strictly necessary. Thus, the mark becomes a very low power signal, practically undetectable in any frequency sub-band. In practice, this approach is very robust against attacks, but limited in terms of data payload.

SI methods take advantage of the side information paradigm [SHA 58], [COS 83]. The side information principle states that a channel noise known at the transmitter and unknown at the receiver would not decrease the channel capacity (the maximum amount of information which can be theoretically transmitted). Thus, the original document (channel #1 noise in Figure 5.4) should no longer be considered as an impediment to watermark detection. Consequently, side information watermarking is *a priori* ideal. In practice, the methods taking this approach allow the insertion of a very high quantity of information, but only have a very weak robustness.

The following section describes the first joint method, named HIS (*Hybrid Informed & spread Spectrum method*) which takes advantage both of the robustness and transparency of the SS methods and the high data payload inserted by the SI approach [MIT 05-02].

5.4.2.1. *The HIS method presentation*

In order to pass from some side information theoretical concepts to a real-life application, this method adapts and extends the principles in [MIL 04], [MIT 05-01], [MIT 06-02]. Figure 5.5 is a synoptic representation of the method. We shall further detail each of the five blocks presented there. For clarity, they will be discussed in the following order: *mark generation*, *salient vector extraction*, *detection*, *informed embedding* and *channel*.

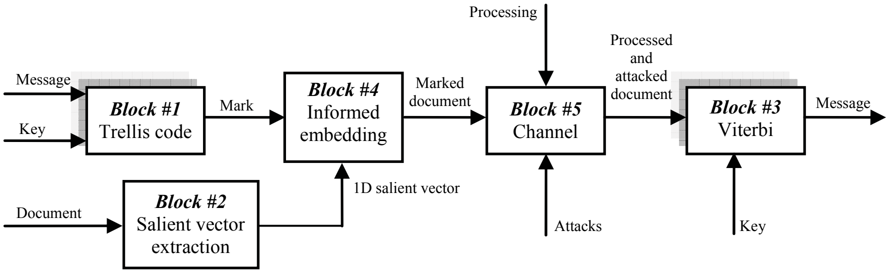


Figure 5.5. *Advanced method synopsis*

5.4.2.1.1. Block #1: mark generation

Let us take there a message of M bits (the copyright information) and the key. The aim of this block is to compute the mark to be embedded, starting from both the message and the key.

In order to be embedded as a mark into the original document, the M bits are encoded by means of a modified trellis code [LIN 83], [MIL 04]. The trellis has K states and 2 arcs exiting each state (each transition codes one bit). Each arc is labeled with an N length vector whose components are real numbers (and not bits like in basic trellis encoders). These labels are computed starting from the key, i.e. they are known only by the true document owner.

Note: the output of a trellis encoder depends on the input bit and on the previous $\log_2 K$ bits. Each combination of $(\log_2 K + 1)$ adjacent bits from the message to be embedded is replaced by an N length label. Consequently, the mark is a vector, denoted by *g*, with real components having an $M \times N$ length.

5.4.2.1.2. Block #2: salient characteristic vector representing the document

The aim of this block is to extract a vector denoted by c_0 which has the same $M \times N$ length as the mark and which contains salient information representing the document.

The watermark is inserted into some transformation coefficients of the document, and not directly into the document. The wavelet decomposition (DWT) proves its efficiency when protecting video and audio documents. The DCT is most suitable for 3D objects. The particular way in which these transforms are applied and the salient coefficients are selected is described in section 5.4.2.2.

5.4.2.1.3. Block #3: detection

Let us take a document which is supposed to be marked. The aim of this block is to establish whether or not the M *bit message has been embedded into the considered document.*

The first task is to extract from the document the salient vector susceptible to convey the mark (see Block #2 above). Then, the coefficients corresponding to the locations where the mark might have been inserted are recorded, thus obtaining a $\hat{c}_w$ vector with $M \times N$ real components.

This vector is the input of a Viterbi decoder [LIN 83]. The decoder is pair designed with the trellis encoder. The cost involved in the Viterbi algorithm is the (un-normalized) correlation coefficient between the input sequence and the transition labels. This cost is to be maximized. Hence, high detection performances are obtained when these labels are uncorrelated.

5.4.2.1.4. Block #4: informed embedding

This block [MIT 05-02] is designed by adapting the principles in [MIL 04]. Its aim is to embed the mark (the g *vector) into the document (represented by the* c_0 *vector). Under the informed watermarking framework, the crucial issue is to find a* c_w *vector which is as close as possible to the* c_0 *vector and for which the Viterbi decoder produces the same output as for the* g *vector.*

The c_w vector is computed by an iterative algorithm (Figure 5.6). In the first iteration, c_w is initialized with c_0. Further on, a vector denoted by b is computed by applying the Viterbi decoder to $c_w + n$, and by trellis encoding the resulting bits. Here, n is a vector of $M \times N$ length, whose components are sampled from a noise source modeling the channel perturbations. This noise is computed as a sum of a Gaussian noise – considered until recently as a universal model for the watermarking channel noise – and a noise that models the non-Gaussian effects [MIT 06-03], [MIT 06-04] of some transformations or attacks (e.g. MP3 compression for audio, the StirMark attack for video).

The c_w vector is now modified according to the following formula:

$$c_w \leftarrow c_w + \alpha \cdot (g - b) / |g - b| .$$

The α scalar value is computed as follows:

$$\alpha = R_t - R(g, b, c_w)$$

where $R(g,b,c_w) = c_w(g-b)/|g-b|$ and R_t is a scalar. The dot product between the c_w and the $(g-b)$ vectors is the un-normalized correlation coefficient.

The loop of b computation and c_w modification is repeated until the condition $R(g,b,c_w) \geq R_t$ is reached several times successively (e.g. 100 times – $N_j = 100$). If the equality between the g and the b vectors is reached before the $R(g,b,c_w) \geq R_t$ condition is achieved, then the b vector is computed without modifying c_w. If such a situation is encountered many times successively (e.g. 100 times – $N_i = 100$), then we consider that the g mark was successfully embedded into the c_w vector: regardless of the added noise, the decoder can recover the message.
The c_w vector thus computed replaces the c_0 salient vector and the marked document is obtained by performing the inverses of the operations in Block #2.

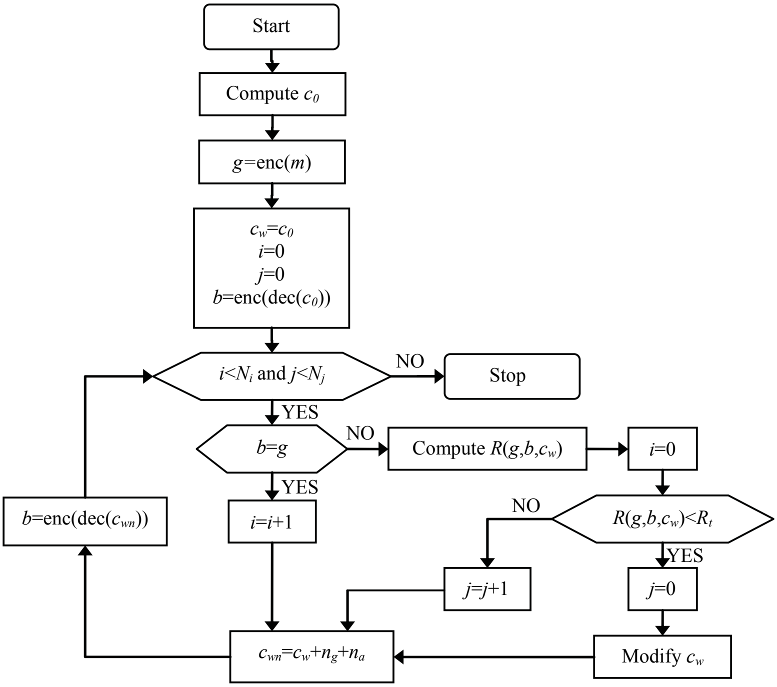


Figure 5.6. *The embedding algorithm synopsis. The enc and dec functions denote the trellis encoder and the Viterbi decoder, respectively. The n_g and n_a terms represent the Gaussian and non-Gaussian noise components, respectively, while m denotes the inserted (public) message*

5.4.2.1.5. Block #5: the channel

The marked document withstands a large variety of transformations.

Generic transformations include format or representation changes, compression, document editing and changing. Then, malicious attacks, dependent on the media type are performed. Generally, in watermarking, all these transformation effects are implicitly assumed to be Gaussian distributed. However, our recent studies [MIT 06-03], [MIT 06-04] on multimedia data statistical behavior brought to light that this Gaussian assumption does not hold for challenging attacks (like the StirMark video attack, for instance) and computed the corresponding models [MIT 07-01]. Consequently, in our watermarking scheme we consider two types of perturbations: (1) Gaussian distributed (denoted by n_g in Figure 5.6) which can model the generic transformations and (2) attack-specific noise sources (denoted by n_a in Figure 5.6).

5.4.2.1.6. Conclusion

The viability of the HIS watermarking approach was tested in collaboration with the SFR (Vodafone group) mobile service provider in France. The aim was to protect several types of multimedia data (video, audio, 3D) coded at low bit-rates (as low as 64kbit/s). In what follows, the detailed way in which the media peculiarities find their place under the method framework is presented.

5.4.2.2. *Video watermarking*

Let us take a color video sequence consisting of L frames. Each frame is represented in the HSV (hue-saturation-value) space; the V component is normalized to a $[0,1]$ interval.

In order to obtain the c_0 salient vector (one of the inputs of the embedding algorithm, (Figure 5.6)) the following steps should be taken:

1) The (9,7) 2D-DWT [CHA 98] is individually applied to each frame in the video sequence, at an N_r resolution level.

2) The coefficients belonging to the HL_{N_r} and LH_{N_r} low frequency sub-bands (gray-shaded in Figure 5.7) are sorted in a decreasing order of their values. The largest D coefficients in each frame are (randomly) shuffled and then recorded into the c_0 vector.

3) The original locations of the c_0 vector components are stored into a v vector.

Let us now specify the numerical values involved in video watermarking.

The experimental data consists of 20 video sequences, each of them having 1,000 frames (40 sec). The frame sizes are 192×160 pixels, corresponding to a Motorola V550 cell phone. These sequences are coded at 64 kbit/s.

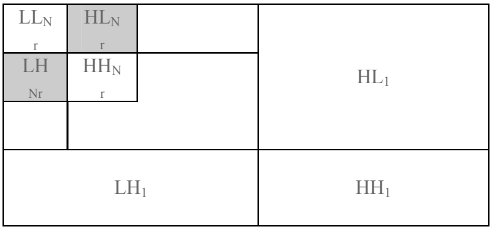


Figure 5.7. *The selected sub-bands*

The 2D-DWT is applied at a $N_r = 3$ resolution level.

The original message to be inserted is represented on $M = 1{,}000$ bits and corresponds to the binary ARTEMIS logo (for illustration, see Figure 5.10a). Each bit from this message is trellis encoded by a $N = 360$ real number label. These numbers are extracted from a random generator obeying a Gaussian distribution of $\mu = 0$ mean and $\sigma = 0.005$ standard deviation. The D number of DWT coefficients selected from each frame is $D = M \times N / L = 360$.

The R_t parameter involved in the embedding scheme (Figure 5.6) was set to $R_t = 2$. The noise generator (Figure 5.6) considers an n_g Gaussian noise of $\mu = 0$ mean and $\sigma = 0.2$ standard deviation and an n_a StirMark noise [MIT 06-03], [MIT 06-04].

In order to subjectively evaluate the transparency, 25 human observers of different ages were involved in experiments: 5 researchers deeply involved in the image/video processing, 5 researchers working in fields not connected to video processing, 5 persons with various educational backgrounds (foreign languages, history, law), 6 students, 1 film director, 1 film producer and 2 painters. They agreed that the method featured fidelity.

In order to also offer an objective measure of transparency, the UIQI was computed for each frame in the video sequence: their minimal, maximal and mean values were 0.9798, 0.9994, and 0.9981 respectively (a UIQI of 1 corresponds to identical images). Frames from original and marked *Fun* sequences are represented in Figures 5.8 and 5.9.

The method also features very good robustness. First, the resistance against mundane video processing was checked: change of file format (from mpg to avi), linear and non-linear filtering (Gaussian, Laplace, median), small rotations (each frame was randomly rotated up to 2 degrees), noise addition, spatial and temporal cropping (up to 25% of frames were randomly dropped). Each and every time, the visual logo was successfully recovered. Secondly, the StirMark attack was individually applied to each frame in the sequence: although the commercial value of the video sequence was completely destroyed during this attack, the logo was still recovered. Figure 5.10 illustrates the robustness. The logo recovered after the file format changing, Laplace filtering and the StirMark attack are represented in Figures 5.10a, b, and c, respectively.

The upper limit of the false alarm probability was evaluated at 10^{-12}.

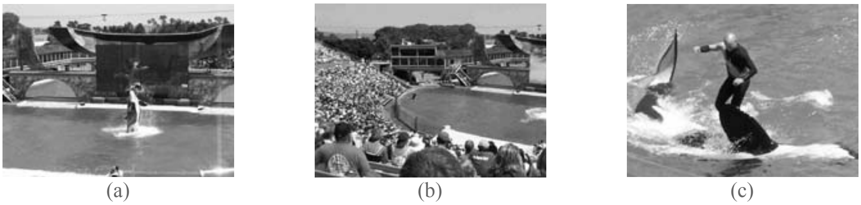


Figure 5.8. *Original frames sampled from the Fun sequence*

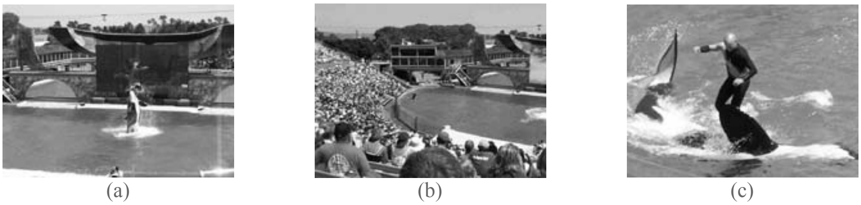
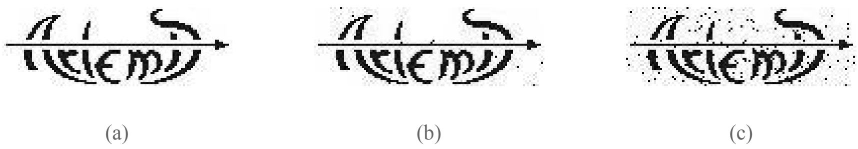


Figure 5.9. *Transparency for video watermarking: marked frames corresponding to the originals in Figure 5.8*



(a) (b) (c)

Figure 5.10. *Robustness for video watermarking: the ARTEMIS logo recovered after the file format changing (a), Laplace filtering (b) and the StirMark attack (c). Note that logo (a) is practically identical to the original logo*

5.4.2.3. *Audio watermarking*

Let us take a stereo audio sequence of L "frames", each frame having T seconds. The c_0 salient vector (Figure 5.6) is computed according to the following steps:

1) The (9,7) 1D-DWT is individually applied to the sum of the left and right channels in each frame, at an N_r resolution level.

2) The coefficients corresponding to the highest frequencies in the lowest sub-band (H_{N_r}, Figure 5.11) of each frame are sorted into a decreasing order of their absolute values and the largest D are recorded in the c_0 vector.

3) The original locations of the selected coefficients are recorded into a v vector.

In the experiments, the following numerical values are considered.

The original audio sequence is sampled at 44.1 KHz and is MP3 compressed at 64 kbit/s.

The message to be embedded has $M = 120$ bits.

The audio document is composed of $L = 120$ frames of $T = 0.3$ secs. The DWT is applied at $N_r = 5$ resolution level.

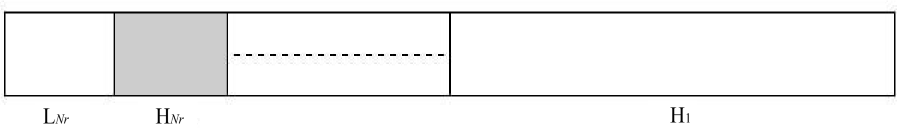


Figure 5.11. *The selected coefficients in an audio frame wavelet decomposition*

The labels in the trellis encoder are represented on $N = 360$ real numbers, sampled from a Gaussian random number generator of $\mu = 0$ mean and $\sigma = 0.01$ standard deviation. Consequently, $D = M \times N / L = 360$ coefficients are recorded from each frame.

The noise source involved in the embedding procedure (Figure 5.6) is composed of an n_g Gaussian generator of $\mu = 0$ mean and $\sigma = 0.6$ standard deviation and of an n_a generator modeling the MP3 compression effects.

This method is applied to a corpus of 100 audio files, of different types: classical, opera, jazz, rock, speech (native speakers in French and English).

When applying the method to music, a good quality is obtained (but, unfortunately, not the fidelity). However, for the speech sequences, a very high fidelity is obtained. This evaluation was independently carried out by 10 human observers (4 researchers, 5 people with different professional backgrounds and 1 musician). In order to objectively evaluate the transparency, the SNR ratio is computed on each sequence, and an average value of 26 dB is obtained.

The method robustness was evaluated by considering the following audio processing: MP3 and OGG compression, stereo to mono conversion, file format conversion, non-linear filtering, noise reduction filtering, equalization, echo addition and flanging. The mark was recovered each and every time, except for the flanging case.

5.4.2.4. *3D object watermarking*

Taking into account its inherent advantages in watermarking (section 5.3.3.1), the NURBS representations have been considered for 3D object protection by means of the HIS method.

An $S(u,v)$ NURBS surface (a function of two variables u and v) is specified by:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j} P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j}}, \quad 0 \le u,v \le 1,$$

where:

– $\left\{P_{i,j}\right\}_{(i,j)\in\{0,\ldots,n\}\times\{0,\ldots,m\}}$, represent a bidirectional control net (a matrix of control points). Each $P_{i,j}$ has three components, corresponding to the three Euclidean coordinates: $P_{i,j}=[P^x_{i,j},P^y_{i,j},P^z_{i,j}]$.

– $\left\{w_{i,j}\right\}_{(i,j)\in\{0,\ldots,n\}\times\{0,\ldots,m\}}$ denote some positive scalars called weights. The larger a weight component, the closer the NURBS surface to the corresponding control point.

– U and V are two knot vectors (non-decreasing sequences of real numbers):

$$U=\left[\underbrace{0,0,\ldots,0}_{p+1},u_{p+1},u_{p+2},\ldots,u_{r-p-1},\underbrace{1,1,\ldots,1}_{p+1}\right]$$

$$V=\left[\underbrace{0,0,\ldots,0}_{q+1},v_{q+1},v_{q+2},\ldots,v_{s-q-1},\underbrace{1,1,\ldots,1}_{q+1}\right],$$

where $r=n+p+1$ and $s=m+q+1$.

– $N_{i,p}(u)$, $i\in\{0,\ldots,n\}$, and $N_{j,p}(v)$, $j\in\{0,\ldots,m\}$ are the p^{th} and q^{th} degree non-rational B-spline basis functions defined on the U and V knot vectors:

$$N_{i,0}=\begin{cases}1 & u_i\le u\le u_{i+1}\\ 0 & \text{otherwise}\end{cases},$$

$$N_{i,p}(u)=\frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u)+\frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u).$$

It can be noticed that the knot vectors determine the influence area of a $P_{i,j}$ control point (the area in the (u,v) space where $N_{i,p}(u)\cdot N_{j,q}(v)\neq 0$).

When inspecting this NURBS representation from the watermarking point of view, two observations can be made. On the one hand, for any NURBS surface, the control points are organized into a matrix structure. The rows and columns in this matrix correspond to the two orthogonal directions with respect to which the NURBS surface is represented [PIE 97]. On the other hand, the control points are, in fact, a set of three real numbers that establish the x, y and z positions in the Euclidean space.

Consequently, for any NURBS surface, the control points can be organized into three matrices, one for each Euclidian axis. *These three matrices can be considered as three virtual images representing the NURBS surface.*

Generally, a 3D object is represented by a set of NURBS surfaces rather than a unique surface. Hence, for a 3D object, there are several sets of three virtual images.

Further on, the 2D-DCT (discrete cosine transform) is individually applied to each of these virtual images. The coefficients obtained on each image are concatenated together into a vector which is sorted in decreasing order.

Finally, the c_0 salient vector for the 3D object is obtained by recording the coefficients having the rank $[r_{\min}; r_{\max}]$ in the sorted vector of DCT coefficients; $r_{\min} = r_{\max} - M \times N + 1$.

In the experiments, the numerical values presented below have been considered.

The message to be embedded is represented on $M = 20$ bits. The trellis encoder has $K = 8$ states. The labels consist of $N = 12$ real numbers. These numbers are extracted from a random generator obeying a uniform distribution in the $[-0.01;\ 0.01]$ interval. In order to ensure good detection properties for the Viterbi algorithm, the un-normalized correlation coefficient between any two labels is lower than 0.01.

When building up the salient vector, the rank $r_{\max}$ is equal to the total number of coefficients divided by 25. This relation holds regardless of the number of control points for the 3D object.

The noise involved in the informed embedding stage was obtained solely from an n_g Gaussian generator of $\sigma = 1.2$ standard deviation (the n_a component was not used for 3D data). The R_t parameter depends on the 3D object to be marked; it can be automatically determined by a search process. In all cases, R_t belonged to the $[0.1;\ 1.5]$ interval; hence, a search with a resolution of 0.1 does not significantly alter the applicability of the method.

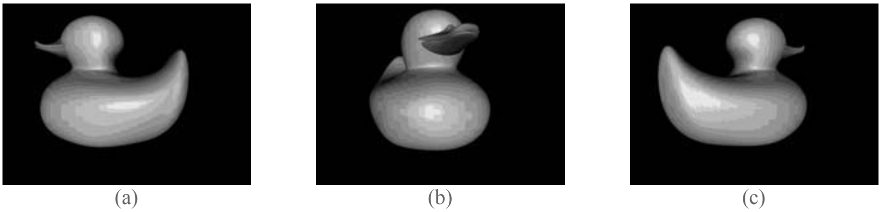
The experimental database consists of 100 objects. It has a heterogenous content: various models, spare parts from car industry, 3D cartoon characters, *etc.* The objects contain different numbers of NURBS surfaces (e.g. from 1 to 200). Each surface has a different number of control points (e.g. from 4 to 10,000). Views from such an object are presented in Figure 5.12.

The first type of experiment is devoted to *transparency*. In order to evaluate this property, 10 human observers were involved in our experiments (4 researchers, 4 teenagers and 2 graphic designers). They generally agreed that no disturbing artefacts can be identified in the marked objects. Views from a marked object were presented in Figure 5.13. Note that here the transparency is evaluated only by subjective means (human observers). Actually, no objective measurement for the differences between two 3D objects represented by NURBS has yet been defined.

The second type of experiment was devoted to *robustness*. The marked models are first corrupted by considering the attacks known in image watermarking. These attacks were applied to the virtual images characterizing the marked 3D object. Good robustness was obtained with respect to JPEG compression (Figure 5.14a), row/column insertion/elimination (Figure 5.14b), affine transformations and the StirMark attack (Figure 5.14c).

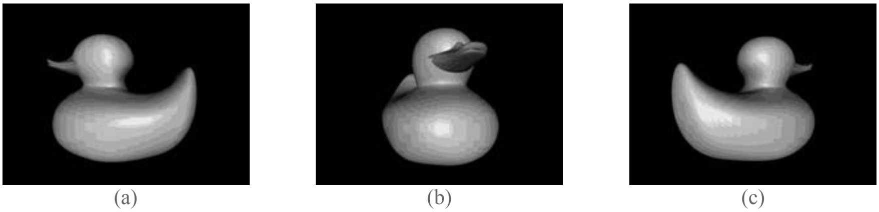
Particular attention should be paid to the object transformation by means of NURBS specific operations. For instance, it is known that several NURBS representations may correspond to the same 3D object. It is then a crucial issue for a mark embedded in such a representation to be recovered from any other NURBS representation of the same object. However, taking into account the discussion in [PIE 97], [MIT 06-02], the usual operations applied to NURBS surfaces (knot insertion/removal/refinement, degree elevation/reduction, changing the control point order and affine transformations) have a direct correspondence on the control point matrices and, consequently, on the associated virtual images.

We then tested the robustness against these NURBS operations by applying the corresponding attack to the virtual images and we obtained good results: all the $M = 20$ embedded bits have been recovered.



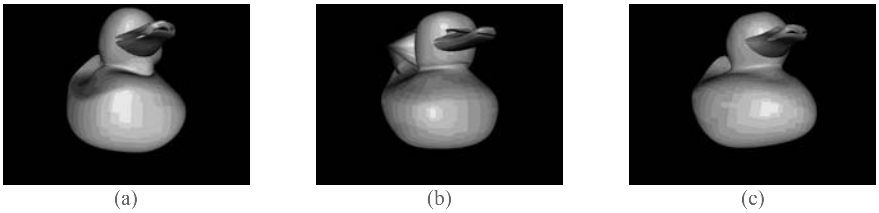
(a) (b) (c)

Figure 5.12. *Views from the original Duck model*



(a) (b) (c)

Figure 5.13. *Views from the marked Duck model*



(a) (b) (c)

Figure 5.14. *Robustness for 3D watermarking: the message was successfully recovered even after compression (a), row/column elimination (b) and the StirMark attack (c)*

5.4.3. *Watermarking capacity*

The main issue in watermarking is to determine the capacity of the channel equivalent to channel 1 and channel 2 from Figure 5.4 (i.e. the capacity of a channel having three noise sources: the original content, the perturbations and the attacks). From the practical point of view, this means determining the maximal theoretical data payload which can be inserted into a document while observing prescribed transparency and robustness constraints. This section summarizes some of the studies related to this issue.

For image watermarking, the two main studies exploring data payload limits are presented below.

The first study [LIN 01] considers an 8×8 DCT gray-level image watermarking scheme and uses JPEG compression as the noise model. It considers an adaptation of a continuous channel to a discrete alphabet channel, in order to define *adjacent symbols*, i.e. input symbols that may lead to the same output symbol. The results are obtained when considering the robustness of the mark against JPEG compression. The obtained capacity is 28,625 bits for a 256×256 gray level image, at a JPEG quality factor of 75. However, when decreasing the quality factor of the JPEG compression, the capacity quickly drops towards zero.

The second study [MOU 02] follows a more general approach to the image watermarking capacity problem. It considers the noisy channel model and define a distortion measure for the watermark and for the attack, given by the square error between the marked/attacked image and the original image. Then, the watermarking capacity problem is considered as a mutual-information game between the attacker and the data hider. The payoff function is considered to be the difference between the mutual information of the watermark and the attack. In order to compute the channel capacity, the authors consider the original image coefficients as obeying the Gaussian law and being independent and identically distributed. The results are derived from the rate-distortion theory. The capacities found for still images are between 30 kbits and 100 kbits, depending on the image and when considering the image distortion after watermarking and attack as twice the distortion induced by the watermark.

Concerning video watermarking, the capacity has been investigated in both DCT and DWT domains [MIT 06-03], [MIT 06-04]. The side information paradigm is considered and, consequently, the effects of the host data noise are disregarded. Both studies employ the frame-wise coefficient hierarchy as the insertion domain and consider the following types of attacks: Stirmark random bending, Gaussian filtering, sharpening and small rotations. By performing an accurate statistical investigation, it is shown that the popular Gaussian assumption concerning the behavior of attacks is not valid. Further on, the capacity upper and lower limits are computed using the Shannon formula for arbitrary additive noise. As a comparison, the Gaussian noise case is also considered. For DCT watermarking, the capacity is found to be between 0.73 bits per frame (the worst case – 5 degree rotation) and 331.51 bits per frame (the best case – Gaussian filtering). For DWT watermarking, the capacity is found to be between 0.15 bits per frame (the worst case – random ± 2 degree frame rotations) and 267.87 bits per frame (the best case – Gaussian filtering). Recent studies also dealt with video capacity in the MPEG-4 AVC domain [DUT 08-02].

Concerning audio capacity, the study in [CVE 03-02] aims to maximize the data payload under the constraint of MP3 compression attacks. The practical limits found for the data payload are quite low: for a common bit error rate (10^{-6}) and for an audio stream compressed at 64kbit/s, only 25 bits can be inserted each second.

The study in [CVE 04-02] deals with the steganography (i.e. no noise for Channel #2 in Figure 5.4): the aim is to find the maximum quantity of information which can be embedded into an audio sequence without altering its quality, by supposing that no attack will occur. The channel capacity is found to be about 1,000 times higher as compared to the MP3 robustness case. Note that in order to compute the capacity, both the original data and the attack effects are assumed to be Gaussian distributed

To the best of our knowledge, no study on 3D watermarking capacity has yet been reported in the literature.

5.4.4. *Conclusion*

Any watermarking application can be represented by a noisy channel. According to this model:

– the mark to be embedded is the signal to be transmitted;

– the original media and its processing (be they malicious or not) stand for the noise;

– the embedding/detection technique correspond to the modulation/demodulation techniques.

Such a model allows us to address two key issues in watermarking: how to design good embedding/detection techniques and how to compute the channel capacity.

Concerning the former issue, SS and SI techniques offer good robustness and large data payload, respectively (see Figure 5.15). In order to get the trade-off between transparency, robustness, and data payload, an original method synergistically combining SS and SI principles was presented. Note that this method is equally good in protecting audio, video and 3D data.

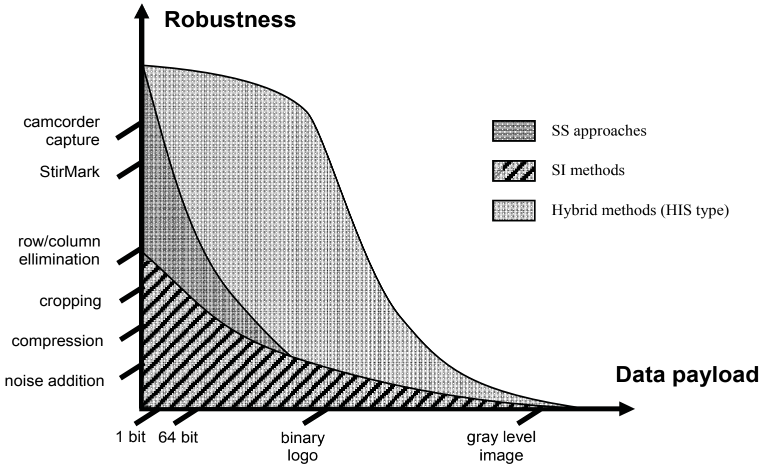


Figure 5.15. *SS, SI and conjoint approaches to video watermarking*

5.5. Discussion and perspectives

5.5.1. *Theoretical limits and practical advances*

Although the watermarking field already disposes of a sound fundamental support, and the economic interest is vivid and continuously increasing, there is still a shortage in related commercial products. This section presents the results of a Web-based investigation on the technical capabilities of the products the main actors in the field are offering. Note that this section is merely informative and has no advertising intentions.

5.5.1.1. *Copyright protection*

Maybe the most important of the watermarking applications, copyright protection, is addressed by several important actors in the multimedia industry.

Philips developed two products [PHI 06], namely *CineFence* and *Video Fingerprinting* for digital cinema and for home video, respectively.

CineFence deals with both video and audio components. Concerning video, it allows 35 bits to be inserted into a sequence of 5 minutes, by a real time procedure. The frame rates are 24fps and 48 fps. The detection is robust against camcorder capturing and subsequent compression down to 1 Mbit/s MPEG – 2, 400 kbit/s DivX, and VideoCD [LAB 06] (i.e. MPEG – 1, resolution 352x240 NTSC and 352x288 PAL). The same data payload (35 bits in 5 minutes) is inserted into the audio component. The sampling frequencies are 48 kHz and 96 kHz. The detection is robust against over the air microphone capturing and subsequent MP3, WMA or AC3 compression, amplitude compression, time scaling, D/A and A/D conversion, resampling, noise addition echo addition, and filtering (all-pass and band-pass). The *CineFence* product is DCI compatible (DCI – Digital Cinema Initiative; see section 5.2).

Video fingerprinting allows video extracts of 5 s to be recognized. The application is robust against low rate video compression, scaling, cropping and noise addition. The software product is capable of monitoring up to 1,000 video channels in parallel and was developed to combat P2P piracy.

Digimarc, a company devoted to watermarking solutions [DIG 06], practically addresses all the multimedia fields: still pictures, movies, TV, audio (music and speech) and ID documents, for both analog and digital forms. It is the owner of more than 100 patents in the field. For copy protection, it developed two products: *ImageBridge* is a solution to manage on-line channel programs and to report on-line logo and image use, while *MyPicturesMarc* was designed to ensure watermark robustness against the usual image manipulations.

The Alphatech software company provides four products [ALP 06], namely *Audiomark*, *Eikonanak*, *Videomark* and *Volmark*, in order to protect audio, still image, video and 3D content, respectively.

iTrace was designed [SAR 06] at the Sarnoff Co., in order to transparently watermark digital cinema content. The mark detection is robust against camcorder capture and data compression.

Thomson [THO 06] provides Nextamp, a software product for video watermarking. Nextamp has quickly found its place on the market and was also subject to preliminary academic evaluations.

The academic research carried out at the Franhofer Institut für Graphische Datenverarbeitung in Darmstadt, Germany resulted in a prototype technology [FRA 06] to protect MP3 audio files.

5.5.1.2. *Broadcast monitoring*

Another traditional application of robust watermarking is broadcast monitoring. The main solution is provided by Philips as the *CompoTrack* product which has three components: *CompoTrack Wav*, *CompoTrack MPEG*, and *CompoTrack H.264*.

CompoTrack Wav allows 37 bits to be inserted in any extract of 45 s. The detection robustness is expressed in terms of D/A and A/D conversion, amplitude compression, resampling, speed change, and noise addition.

CompoTrack MPEG provides a datapaylod of 21 bits in any 90 s and robustness to compression (MPEG-1, DivX, WMV), shifting, cropping, scaling, noise addition, D/A conversion, median filtering. The marked video stream can be directly embedded into the MPEG-2 stream.

CompoTrack H.264 has the same performances as above but was designed for MPEG-4 compatibility.

5.5.1.3. *Forensic tracking*

Forensic tracking is meant to identify the content and the source for audio/video and ID documents.

The Philips *RepliTrack* inserts a data payload of 21 bits in 90 s of video. As it was designed to protect video on optical disks, the mark detection is robust against MPEG-2, DivX, and WMV compression.

5.5.2. *Watermarking and standardization*

Looking at the watermarking studies from a standardization point of view, it becomes obvious that many difficulties should be overcome:

– there are not too many convergence points among the research studies carried out at different laboratories;

– it is not yet stated what should be standardized;

– the multimedia content producer point of view was not yet clearly stated.

This context, still being ill-defined, requires the distinction between the watermarking issues to be standardized and the existing standardization efforts.

5.5.2.1. *What should be standardized?*

As a preliminary step in a standardization process, this section tries to identify some requirements any watermarking technique should observe.

5.5.2.1.1. What should be protected?

The mark should be embedded into the multimedia content itself and not into some (auxiliary) metadata (file headers, for instance). Note that nowadays, as a palliative solution to property right protection, some producers keep their file formats secret (this is mainly the case for 3D data). With the progresses in software engineering, such an approach will soon be obsolete.

5.5.2.1.2. Which representation should be used for multimedia data?

For a real-life application, content representation should not play any role in method performances. For instance, it should be possible for the mark embedded into the mesh representation of the original 3D object to be recovered starting from the NURBS representation of the marked object.

5.5.2.1.3. How long should the key be?

The length (in bits) of the key (i.e. the information known only by the true owner) should be large enough to ensure the computational security: a malicious user should not be capable of tying down the mark by a brute force search in the key space. However, note that in some countries (such as France and the USA), an upper limit for the amount of secret information is set by law.

5.5.2.1.4. How much public information should be inserted?

As already discussed, the answer to such a question is quite fuzzy. The majority of studies consider 64 to 96 bits as a reasonable quantity to identify the owner (they may correspond to a serial number). However, for some practical applications, a larger number of bits would be required (e.g. not only to identify the owner, but the buyer, the date of purchase, and some conditions of use, as well); hence, we may expect to deal with up to 256 bits.

5.5.2.1.5. Oblivious or non-oblivious watermarking?

This time, the answer can be very clearly stated: oblivious. Let us consider the scenario according to which an independent authority would be in charge of protecting the property rights. To grant the access to the original (unmarked) object, even for such an authority, can be considered as a fault in security.

5.5.2.1.6. Spread spectrum or informed watermarking?

Such an issue should not be the object of standardization. In the near future, some hybrid techniques are to be advanced. After some years of sound studies in side information channels, a related approach will be expected to impose itself. Actually, it is the interface rather than the method that is likely to be standardized.

5.5.2.1.7. Robustness under a standardization framework?

By its very definition, robustness is a dynamic concept: each advanced method will be followed by a devoted attack. Hence, in this respect, a standard specification would state only some minimal requirements; moreover, it is to be periodically updated.

5.5.2.1.8. Transparency under a standardization framework?

The transparency is a subjective notion: the way in which the artefacts induced during the embedding procedure are perceived by the human senses depends on age, professional background and on the interest in the content itself. Hence, an objective measure (a distance between the original and the marked object) should be defined. However, note that such a measure is not watermarking-specific: it can be inherited and/or adapted from an other field, such as compression or indexing.

5.5.2.1.9. Which is the limit for false alarm probability?

This upper limit depends a lot on the targeted application. In a standardization approach, values close to 10^{-10} may be considered.

5.5.2.2. *Watermarking standardization efforts: a concise state of the art*

When addressing copyright-related problems, several standardization approaches exist, be they issued by MPEG (Moving Picture Experts Group), OMA (Open Mobile Alliance), DCI (Digital Cinema Initiatives) or ISMA (Internet Streaming

Media Alliance). Among these, some documents address these problems without considering watermarking as a solution: MPEG IPMP, OMA DRM, ISMA encryption and authentication. The only documents explicitly dealing with watermarking applications are the DCI DRM and the MPEG PAT.

5.5.2.2.1. MPEG (Moving Picture Experts Group)

In MPEG, all the problems which can be related to copyright protection are grouped under the IPMP (Intellectual Property Management and Protection) or PAT (Persistent Association Tool) frameworks.

MPEG-4 requirements [MPE 01] address this issue in Authentication, IPMP and Sharing Tools, and Requirements for IPMP sections. The intellectual property rights protection mechanisms in MPEG-4 are provided by supplementing the coded media objects with an optional intellectual property identification dataset, carrying information about the contents, type of object and pointers to rights holders [MPE 02]. This IPMP framework leaves the details of IPMP systems design in the hands of application developers. While MPEG-4 does not standardize IPMP systems itself, it does standardize the MPEG-4 IPMP interface. This interface consists of IPMP Descriptors and IPMP Elementary Streams.

MPEG-7 requirements [MPE 05] state the following standard obligations concerning IPMP. A mechanism for pointing to content rights should be provided, but these rights will be not directly described. Rights management information and technological protection measures used to manage and protect content should be accommodated and not interfered with. Applications that distinguish between legitimate and illegitimate content, both inside and outside trusted domains, should be supported. Content identification by international identification conventions should be enabled.

MPEG-21 IPMP [MPE 06] must preserve the confidentiality of the user (while keeping a possibility of overriding the confidentiality barrier) and allow at any moment the transfer and modification of rights according to the sale conditions. The possibility of interoperability with other IPMP standards is mandatory. The possibility of a request for the presence of a mark is also included.

The *MPEG-21 PAT section* (Persistent Association of Information with Digital Items – Requirements) must allow the creation of a persistent association between the content and some metadata, regardless the transport format. The definition of this association must follow a set of constraints:

– it must declare its maximal data payload;

– the modification or deletion of the association can be done only by its creator;

– it should be usable in a streaming environment;

– it must not alter the content beyond a certain limit.

Against all efforts, the MPEG-21 PAT [MPE 06] are not yet standardized. Although technical report [ISO 04] identifies seven watermarking properties, offering a first tool for method evaluation, it barely scratches the surface, thus leaving the problem open to interpretation. This is a consequence of the fact that the MPEG has not yet decided whether standards concerning watermarking should be stated or not.

5.5.2.2.2. OMA (Open Mobile Alliance)

OMA DRM [OMA 03] provides requirements concerning *security*, *rights*, and *privacy*. These requirements strictly define a framework under which the multimedia digital objects are manipulated. In this respect, all the considered tools belong to cryptography. No entry points for potential watermarking applications are provided. The considered DRM applications are transparent for the end user and run only on OMA-compliant terminals.

5.5.2.2.3. DCI (Digital Cinema Initiative)

The DCI specification [DCI 05] presents one of the most complete frameworks for multimedia content protection. Within this specification, both cryptography and watermarking (whether fragile or robust) are supported, thus enabling a holistic approach to video/audio protection.

On the one hand, some of the protection issues are here alleviated, as the DCI specification is created only for digital content used in cinemas. Hence, only a few users (with known identities) have access to the digital content and only on specialized terminals. In consequence, the protection scheme is specified only for these terminals and for the transport network.

On the other hand, watermarking is here considered as a backup protection system against in-theatre camcorder capture. This imposes very strong constraints on robustness (warping, frame rate changing, time/spatial cropping). Moreover, both the video and audio components are to be marked.

In order to allow the identification of the video and the moment when the illegal recording has taken place, the inserted mark is required to carry at least 35 bits of data. It should contain a time stamp with a 15 minute precision, indicating the quarter of the hour, the hour, day and year, this stamp being represented on 16 bits. The other 19 bits record the serial number or location code. The 35 bits of the mark should be inserted in every 5 minute segment of video. The marking of the video is considered as being done during playback, thus imposing real-time constraints.

The detection of the mark should allow positive identification of the video starting from a 30 minute excerpt.

In terms of robustness, the DCI-compliant video watermark must survive the following attacks:

– digital-to-analog (D/A) and analog-to-digital (A/D) conversions;

– re-sampling and re-quantization;

– contrast and color enhancements;

– resizing;

– letterboxing (black masks added in order to make the video suitable for TV screen format);

– aperture control;

– low-pass filtering and anti-aliasing;

– brick wall filtering;

– noise reduction filtering;

– frame swapping;

– compression;

– scaling;

– cropping;

– overwriting;

– noise addition;

– collusion;

– format conversion, including among formats that imply special and temporal resolution changing (e.g. between NTSC and PAL);

– horizontal and vertical shifting;

– arbitrary scaling (aspect ratio changing);

– camcorder capture (note that this means a robustness against geometric attacks as well);

– low bit rate compression (500 Kbps for H264 and 1.1Mbps for MPEG-1 compression are given as example values).

The audio watermark is required to survive:

– D/A and A/D conversions;

– radio frequency and infrared transmission;

– multiplexing and de-multiplexing operations;

– re-sampling;

– down conversion;

– compression and expansion;

– pitch shift;

– linear speed changes up to 10%;

– pitch-invariant time scaling up to 4%;

– lossy compression (coding);

– non-linear amplitude compression;

– additive and multiplicative noise;

– equalization;

– echo addition;

– band pass filtering;

– flutter and wow;

– overdubbing.

Note that the robustness is named *survivability* in this standard, as the watermarking is considered for *forensic* purposes.

5.5.2.2.4. ISMA (Internet Streaming Media Alliance)

ISMA is an organization aiming to promote interoperability among open standards. Actually, its specifications represent additional layers on existing standards. For instance, ISMA Encryption and Authentication v.1.1 [ISM 05] is an additional layer on the MPEG-4 IPMP. It addresses the *confidentiality* and *data integrity* issues by considering solely cryptography tools. Actually, section 10 in [ISM 05] defines the core of a cryptography method, which can be further upgraded/replaced. This violates the MPEG-4 IPMP philosophy, according to which the cryptography/watermarking tools should not be specified.

5.6. Conclusion

The main goal of watermarking is to offer the industrial players viable methods to imperceptibly and persistently associate some supplementary information with original multimedia content.

Although traditionally this supplementary information was only used for copyright certification, it has been proved that it can also be exploited for emerging multimedia enrichment applications.

After a short, yet dense history, digital watermarking can hope to answer three fundamental questions:

– What are the watermarking theoretical limits?

– What are the place and role of watermarking in the information and communication society?

– How can we develop methods approaching the watermarking theoretical limits?

In the traditional domain of copyright protection, the efforts are directed toward:

– the creation of a first watermarking-based DRM system;

– the creation of a standardized framework for watermarking applications

– the modeling of the main modules of a watermarking scheme.

In the emerging domain of enriched media, several directions can be explored:

– inserting the user profile directly in the multimedia product;

– creating a remote control system by means of watermarking;

– specifying and implementing an intelligent transmission channel.

5.7. Bibliography

[ALP 06] http://www.alphatecltd.com/watermarking/watermarking.html.

[ARN 02] ARNOLD M., "Subjective and objective quality evaluation of watermarked audio tracks", *Proc. of the IEEE Second International Conference on WEB Delivering of Music*, 2003, pp. 161-167.

[ARN 03] ARNOLD M., SCHMUNCKER M., WOLTHUSEN S., *Techniques and Applications of Digital Watermarking and Content Protection*, Artech House, 2003.

[BEN 99] BENEDENS O., "Geometry-based watermarking of 3D models", *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 46-55, 1999.

[BEN 05] BENNOUR J., DUGELAY J.L., "Tatouage d'objets 3-D via la carte de profondeur asociée", *Vol. CORESA 2005 – 10èmes journées Compression et représentation des signaux audiovisuels*, Nov. 2005, pp. 133-138.

[CHA 98] CHALDERBANK A., DAUBECHIES I., SWELDEN W., YEO B., "Wavelet transforms that map integers to integers", *Appl. Comput. Harmon. Anal.*, vol. 5, no. 3, pp. 332-369, 1998.

[COS 83] COSTA M., "Writing on dirty paper", *IEEE Transactions on Information Theory,* vol. IT-29, pp. 439-441, 1983.

[COX 02] COX I., MILLER M., BLOOM J., *Digital Watermarking*, Academic Press, 2002.

[CVE 03-01] CVEJIC N., STEPPANEN T., “Robust audio watermarking in wavelet domain using frequency hopping and patchwork method”, *Proc. of the 3rd International Symposium on Image and Signal Processing and Analysis*, vol. 1, pp. 251-255, 2003.

[CVE 03-02] CVEJIC N., TUJKOVIC D., STEPPANEN T., “Increasing robustness of an audio watermark using turbo codes”, *Proc. of the IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 217-220, 2003.

[CVE 04-01] CVEJIC N., TUJKOVIC I., “Increasing robustness of patchwork audio watermarking algorithm using attack characterization”, *Proc. of the IEEE International Symposium on Consumer Electronics*, pp. 3-6, 2004.

[CVE 04-02] CVEJIC N., SEPPANEN T., “Channel capacity of high bit rate audio data hiding algorithms in diverse transform domains”, *Proc. of the International Symposium on Communications and Information Technologies*, pp. 84-88, 2004.

[DAR 04] DARAS P., ZARPALAS D., TZOVARAS D., STRINZIS M.G., “Watermarking of 3D models for data hiding”, *Proc of the IEEE Intl. Conf. on Image Processing*, vol. 1, October, pp. 47-50, 2004.

[DAV 04] DAVOINE F., PATEUX S. (Eds), *Tatouage de documents audiovisuels numériques*, Hermes/Lavoisier, 2004.

[DCI 05] DCI Digital Cinema System Spec, version 1.0a, July 2005.

[DOE 05] DOERR G., DUGELAY J.L., “Collusion issue in video watermarking”, *Traitement du Signal*, vol. 22, no. 6, pp. 563-574, 2005.

[DIG 06] http://www.digimarc.com/patent/watermarking_applications.asp.

[DUT 08-01] DUTA S., MITREA M., PRETEUX F., BELHAJ M., “MPEG-4 AVC domain watermarking transparency”, *Proc. SPIE*, vol. 6982, 2008.

[DUT 08-02] DUTA S., MITREA M., BELHAJ M., PRETEUX F., “A comparative study on inserting strategies in MPEG-4 AVC watermarking”, *Proc. SPIE*, vol. 7075, 2008.

[FRA 06] http://www.iis.fraunhofer.de/amm.

[GIR 06] GIRARD L., “Le marché publicitaire emballé par le football et le téléphone”, *Le Monde*, May 25, 2006.

[HE 05] HE X., SCORDILIS M.S., “Improved spread spectrum digital audio watermarking based on a modified perceptual entropy psychoacoustic model”, *Proc. of the IEEE Southeast Con.*, pp. 283-286, 2005.

[IIP 05] http://www.iipa.com/special301_TOCs IIPA (International Intellectual Property Alliance).

[ISM 05] ISMA Encryption and Authentication, version 1.1, December 2005.

[ISO 04] ISO/IEC JTC 1/SC 29/WG 11 N6829, Draft of PDTR for potential second edition of ISO/IEC 21000-11.

[KAL 03] KALIVAS A., TEFAS A., PITAS I., "Watermarking of 3D models using principal component analysis", *Proc. of the Intl. Conf. on Multimedia and Expo – ICME,* vol. 3, pp. 637-640, July, 2003.

[KIM 03] KIM H.J., "Audio watermarking techniques", invited lecture, *Pacific Rim Workshop on Digital Steganography*, Japan, 2003.

[KIM 04] KIM H.J., KIM T., YEO I.K., "A robust watermarking scheme", *Proc. of the International Symposium on Circuits and Systems*, vol. 5, pp.696-699, 2004.

[KIS 03] KISHK S., JAVIDI B., "3D object watermarking by a 3D hidden object"**,** *Optics Express*, vol. 11, no. 8, pp. 874-888, 2003.

[KO 05] KO B.S., NISHIMURA R., SUZUKI Y., "Time-spread echo method for digital audio watermarking", *IEEE Transactions on Multimedia*, vol.7, no. 2, pp. 212-221, 2005.

[KWO 03] KWON K.R., KWON S.G., LEE S.H., KIM T.S., LEE K.I., "Watermarking for 3D polygonal meshes using normal vector distributions of each patch", *Proc. of the IEEE Intl. Conf. on Image Processing*, vol. 2, pp. 499-502, September, 2003.

[LAB 06] http://www.labdv.com/leon-lab/video /videocd.htm.

[LEE 02] LEE J.J., CHO N.I., KIM J.W., "Watermarking for 3D NURBS graphic data", *Proc. of the IEEE Workshop on Multimedia Signal Processing*, pp. 304-307, December, 2002.

[LEE 03] LEE S.H., KIM T.S., KIM B.J., KWON S.G., KWON K.R., LEE K.I., "3D polygonal meshes watermarking using normal vector distribution", *Proc. of the IEEE Intl. Conf. on Multimedia and Expo – ICME*, vol. 3, pp. 105-108, July, 2003.

[LEE 05] LEE J.W., LEE S.H., KWON K.R., LEE K.I., "Complex EGI based 3D-mesh watermarking", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, no. 6, pp. 1512-1519, 2005.

[LI 04] LI L., ZHANG D., PAN Z., SHI J., ZHOU K., YE K., "Watermarking 3D mesh by spherical parameterization", *Computers and Graphics*, vol. 28, no. 6, pp. 981-989, 2004.

[LIN 83] LIN S., COSTELLO D.J., *Error Control Coding: Fundamentals and Applications*, Prentice Hall, 1983.

[LIN 01] LIN C.Y., CHANG S.F., "Zero-error information hiding capacity of digital images", *Proc. of the 2001 IEEE International Conference on Image Processing*, vol. 3, pp. 1007-1010, October, 2001.

[LIU 04] LIU Y.W., SMITH J., "Watermarking sinusoidal audio representations by quantization index modulation in multiple frequencies", *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 373-376, 2004.

[LOB 03] LOBOGUERRERO A., BAS P., LIENARD J., "Iterative informed audio data hiding scheme using optimal filter", *Proc. of the International Conference on Communication Technology*, vol. 2, pp. 1408-1411, 2003.

[MAL 89] MALLAT S., "A theory of multiresolution signal decomposition: the wavelet representation", *IEEE Tans. on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, 1989.

[MAL 04] MALIK H., KHOKHAR S., RASHID A., "Robust audio watermarking using frequency selective spread spectrum theory"**,** *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 385-388, 2004.

[MAN 01] MANSOUR M.F., TEWFIK A.H., "Audio watermarking by time-scale modification", *Proc. of the IEEE International Conference on Acoustics: Speech, and Signal Processing*, vol. 3, pp. 1353-1356, 2001.

[MIL 04] MILLER M., DOERR G., COX I., "Applying informed coding and embedding to design a robust high-capacity watermark", *IEEE Trans. on Image Processing*, vol. 13, no. 6, pp. 792-807, 2004.

[MIT 04] MITREA M., ZAHARIA T., PRETEUX F., "3D object protection: watermarking method for NURBS surfaces", *Proc. MediaNet 2004*, pp. 35-42, Tozeur, Tunisia, November, 2004.

[MIT 05-01] MITREA M., PRETEUX F., VLAD A., "Spread spectrum color video watermarking in the DCT domain", *Journal of Optoelectronics and Advanced Materials*, vol. 7, no. 2, pp. 1065-1072, 2005.

[MIT 05-02] MITREA M., PRETEUX F., NUNEZ J., Procédé de Tatouage d'une Séquence Video, French Patent no. 05 54132, extended to EU under the no 1804213 (filled-in 2005).

[MIT 06-01] MITREA M., DUTA S., ZAHARIA T., PRETEUX F., "Ensuring multimedia content enrichment by means of data hiding techniques", *Proc. SPIE*, vol. 6383, 2006.

[MIT 06-02] MITREA M., DUTA S., PRETEUX F., "2D Approaches to 3D watermarking: state-of-the-art and perspectives", *Proc. SPIE*, vol. 6064, pp. 232-243, 2005.

[MIT 06-03] MITREA M., DUTA S., PRETEUX F., VLAD A., "Data payload optimality: a key issue for video watermarking applications", *Proc. SPIE*, vol. 6315, 2006.

[MIT 06-04] MITREA M., DUTA S., ZAHARIA T., PRETEUX F., "Ensuring multimedia content enrichment by means of data hiding techniques", *Proc. SPIE*, vol. 6383, 2006.

[MIT 07-01] MITREA M., DUMITRU O., PRETEUX F., VLAD A., "Zero memory information sources approximating to video watermarking attacks", *LNCS*, vol. 4707-III, pp. 445-459, 2007.

[MOU 02] MOULIN P., MIHCAK M., "A framework for evaluating the data-hiding capacity of image sources", *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 1029-1042, 2002.

[MPE 01] MPEG-4 Requirements, version 16, ISO/IEC JTC1/SC29/WG11 N3930, January 2001.

[MPE 02] MPEG-4 Overview, version 21, ISO/IEC JTC1/SC29/WG11 N4668, March 2002.

[MPE 05] MPEG-7 Requirements, version 18, ISO/IEC JTC1/SC29/WG11 N6881, January 2005.

[MPE 06] MPEG-21 Requirements, ISO/IEC JTC1/SC29/WG11 N7778, January 2006.

[OMA 03] OMA DRM Requirements, version 2.0, 15 May 2003.

[PET 01] PETROVIC R., "Audio signal watermarking based on replica modulation", *Proc. of the 5th International Conference on Telecommunications in Modern Satellite: Cable and Broadcasting Services*, vol. 1, pp. 227-234, 2001.

[PHI 06] http://www.business-sites.philips.com/ contentidentification/products/index.html.

[PIE 97] PIEGEL L., TILLER W., *The NURBS Book*, Springer Verlag, 1997.

[SAR 06] http://www.sarnoff.com/news/index.asp?releaseID=116.

[SHA 58] SHANNON C.E., "Channels with side information at the transmitter", *IBM Journal*, pp. 289-293, 1958.

[SON 02] SONG H.S., CHO N.I., KIM J.W., "Robust watermarking of 3D mesh models", *IEEE Workshop on Multimedia Signal Processing*, pp. 332-335, December, 2002.

[STE 01] STEINBACH M., PETICOLAS F., RAYNAL F., DITTMANN J., FONTAINE C., SEIBEL S., FATES N., FERRI L., "StirMark benchmark: audio watermarking attacks", *Proc. of the International Conference on Information Technology: Coding and Computing*, pp. 49-54, 2001.

[STE 05] STERLING M., TITLEBAUM E.L., DONG X., BOCKO M.F., "An adaptive spread-spectrum data hiding technique for digital audio", *Proc. of the International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 685-688, 2005.

[THO 06] http://www.thomson.net/EN/Home/Technology/technology_solutions/content security.htm.

[TRA 03] TRAN S., PREDA M., PRETEUX F., FAZEKAS K., "Exploring MPEG-4 BIFS features for creating multimedia games", *Proc. of the IEEE International Conference on Multimedia and Expo(ICME'03)*, pp. 429-432, Baltimore, WA, vol. 1, July, 2003.

[VOL 01] VOLOSHINOVSKIY S., PEREIRA S., PUN T., EGGERS J., SU J.K., "Attacks on digital watermarks: classification, estimation-based attacks, and benchmarks", *IEEE Communications Magazine*, pp. 118-126, 2001.

[WAN 02] WANG Z., BOVIK A., "A universal image quality index", *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81-84, 2002.

[WAT 99] WATSON A., HU J., MCGOWAN J.F., MULLIGAN J.B., “Design and performance of a digital video quality metrics”, *Proc. SPIE*, vol. 3644, pp. 168-174, 1999.

[YEO 03] YEO I.K., KIM H.J., “Modified patchwork algorithm: a novel audio watermarking scheme”, *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 4, pp. 381-386, 2003.

[ZAF 04] ZAFEIRIOU S., TEFAS A., PITAS I., “A blind robust watermarking scheme for copyright protection of 3D mesh models”, *Proc. of the IEEE Intl. Conf. on Image Processing*, pp. 1569-1572, vol. 3, October 2004.

This page intentionally left blank

PART 2

Off-the Shelf Technologies

This page intentionally left blank

Chapter 6

Bluetooth Security

6.1. Introduction

Bluetooth is a wireless communication technology intended to simplify short-range connections between devices. Bluetooth was originally envisaged to replace cables between computers and different peripherals like a printer, a scanner, a pointing device, a mobile phone, a Personal Digital Assistant (PDA), a digital camera, a CD drive, a microphone, a speaker, etc.

Bluetooth also has an opportunistic use. The technology can be used when several devices which were not intended to meet need to communicate. For example, a user can transfer data from a cellular phone to a laptop which belongs to another user. He can also print his data on the closest printer or execute some tasks which require the cooperation of other devices.

In 1994, Bluetooth was conceived when a research team, steered by Dr Jaap Haartsen and Dr Sven Mattisson at the Swedish telecommunications manufacturer Ericsson, began to study the feasibility of a low-power and low-cost radio interface in order to eliminate cables between mobile phones and their accessories.

The specification of Bluetooth was developed by the “Bluetooth Special Interest Group” (SIG) trade association. The SIG was founded in September 1998 by Ericsson, IBM Corporation, Intel Corporation, Nokia Corporation and Toshiba Corporation. In 1999, 3Com Corporation, Lucent Technologies, Microsoft Corporation and Motorola joined the SIG.

Chapter written by Franck GILLET.

In July 1999, the IEEE 802.15 working group for Wireless Personal Area Networks (WPANs) proposed the Bluetooth specification version 1.0. The ISM (Industrial, Scientific and Medical) frequency band in the 2.4 GHz spectrum was chosen for Bluetooth operation because this band requires no license. In the specification proposed in 2005, the effective range of Bluetooth devices is defined from 32 feet to 320 feet (from 10 to 100 meters) and the throughput is equal to 723.2 Kbps or 2.1 Mbps.

On 28 March 2006, the "Bluetooth Special Interest Group" announced the next generation of Bluetooth which would be able to achieve throughput 100 times greater than the actual version, that is, from 1 to 100 Mbps. The throughput of Bluetooth technologies which is used in more and more devices should increase in the years to come, allowing high definition video applications and audio transfer. The new "Ultra-Wide Band" technologies will be inserted in the standard. Today Bluetooth is implemented by more than 10,000 electronics companies.

Even if Bluetooth should not be envisaged to replace the infrared technologies, today several constructors choose Bluetooth technology over infrared. The advantage of Bluetooth in comparison with the infrared technology IrDA ("Infrared Data Association") is its capability to connect multiple devices over a single adapter. Bluetooth allows point-to-point and point-to-multipoint connections by minimizing users' intervention. Moreover, infrared channels need a direct view to work, which restricts uses.

Bluetooth technology belongs to WPANs which are used for wireless communication between devices close to one person (about 10 meters around). To standardize the WPAN technologies, the Institute of Electrical and Electronics Engineers (IEEE) created a working group with the following task groups:

– IEEE 802.15.1 defines Bluetooth specification for a basic rate (1 Mbps);

– the IEEE 802.15 Task Group 2 for WPANs developed recommended practices to facilitate coexistence of WPANs and Wireless Local Area Networks (WLANs);

– IEEE 802.15.3 is chartered to draft and publish a new standard for high-rate (20 Mbit/s or greater) WPANs;

– IEEE 802.15.4 is chartered to investigate a low data rate solution with multi-month to multi-year battery life and very low complexity;

– IEEE 802.15.5 gives the necessary mechanisms to enable mesh networking in WPANs.

The word Bluetooth is inspired by the King of Denmark Harald I who was nicknamed Harald Blåtand. Blåtand means “Blue Tooth”. Harald “Bluetooth” was famous for having united Denmark, Norway and Sweden. In the same way, Bluetooth technology is designed to allow collaboration between differing industries such as the computing, mobile phone and automotive markets. The Bluetooth logo merges from the initials for Harald Blåtand written in the runic alphabet.

This chapter is organized in the following way: section 6.2 introduces the Bluetooth technical specification; the section 6.3 displays the problems of security of Bluetooth technology, recent attacks to which this technology has been subjected and resolutions which were created. Finally section 6.4 gives a synthesis of Bluetooth security.

6.2. Bluetooth technical specification

6.2.1. *Organization of Bluetooth nodes in the network*

Bluetooth links operate according to cooperation between a master device and slave devices. A set of devices defines a cell called “piconet”. A piconet is composed of one master and seven active slaves at most (star topology). A master may communicate directly with slaves. The slaves cannot communicate directly with each other. The master is responsible for initiating connections and controlling slaves’ traffic.

A master can manage only 7 active slaves. However, it can manage up to 255 slaves in “parked” mode. The devices in parked mode are synchronized on the clock of the master device, but they do not have a physical address in the piconet. The master device can set a slave from a parked mode to an active mode at any instant.

The slave devices can have multiple master devices. Thus, several piconets can interconnect and form a “scatternet”. Each device can take part in one or many piconets (maximum 3), either as a slave or as a master in a piconet and a slave in another one.

Figure 6.1 illustrates a scatternet which contains three piconets.

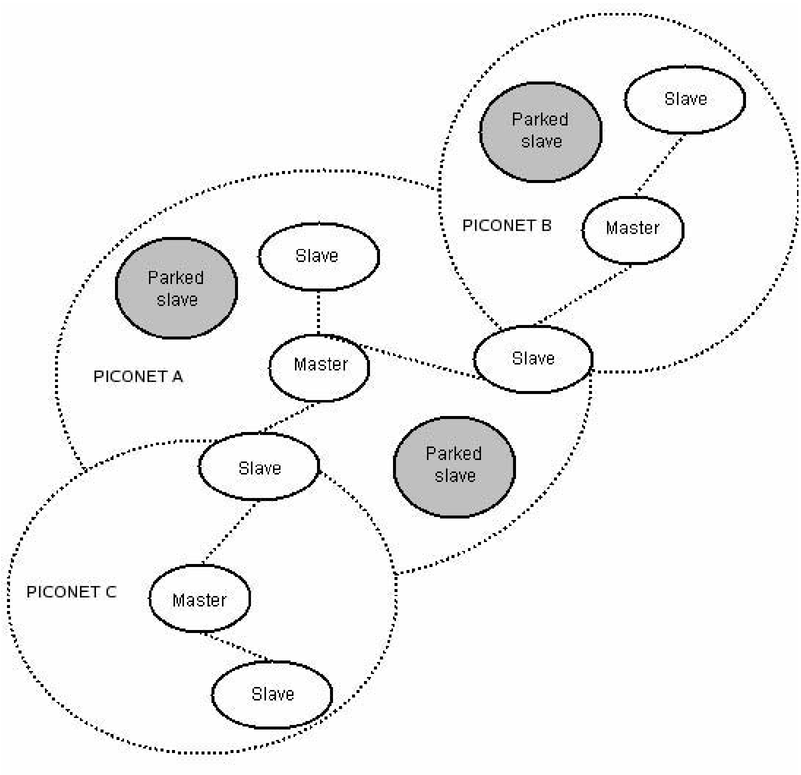


Figure 6.1. *An example of a Bluetooth network*

6.2.2. *Protocol architecture in a Bluetooth node*

The Bluetooth protocol stack allows Bluetooth entities to interconnect, to exchange data and to execute interactive and interoperable applications. The Bluetooth protocol architecture is represented in Figure 6.2. To gain a global understanding of this technology, we will analyze each protocol. The presentation of these protocols will be from the lowest layer to the highest layer, according to the point of view of a receiver.

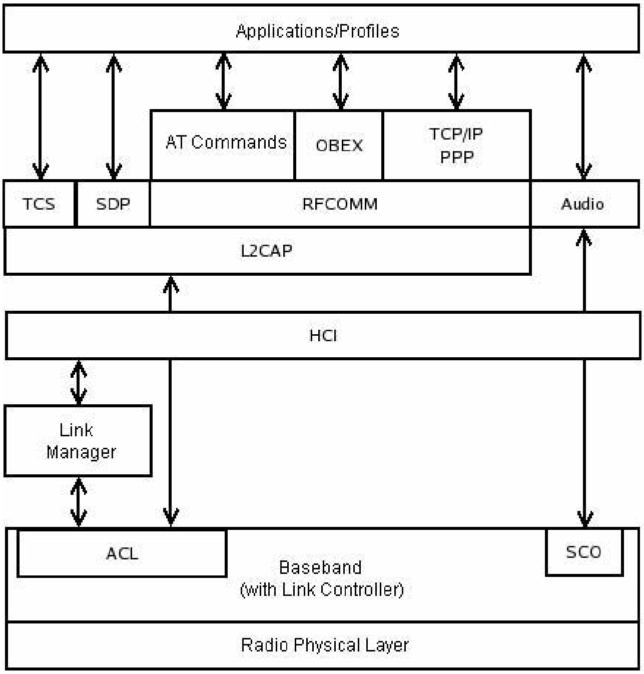


Figure 6.2. *Bluetooth protocol architecture*

6.2.3. *Radio physical layer*

This layer is responsible for the transmission and reception of information on a physical channel. The specification of this layer defines the physical characteristics of the channel.

Bluetooth devices operate in the ISM (Industrial, Scientific and Medical) band reserved to industry, science and medicine. This frequency band stretches over 83.5 MHz, from 2.400 GHz to 2.4835 GHz. In order to comply with regulations in each country, a guard band is used at the lower (2 MHz bandwidth) and upper band edges (3.5 MHz bandwidth).

This band is divided into 79 physical channels spaced out by 1 MHz. RF (Radio Frequency) channels are ordered in channel number k and centered on frequency f(k) = 2402+k MHz, where k = [0.78].

To avoid interference, Bluetooth applies a Frequency-Hopping Spread Spectrum (FHSS) method. The basic piconet physical channel is divided into time slots, each

625 μs in length. Each time slot is related to a hop frequency among the 79 physical channels. Because the period of a slot is 625 μs, consecutive hops can occur at a rate of 1,600 hops per second.

The duration of a packet is variable. A packet may extend over one, three or five consecutive time slots. The frequency is fixed for the duration of the packet. The frequency in the first slot determines the frequency of a multi-slot packet.

Figure 6.3 illustrates the used frequency-hopping spread spectrum method.

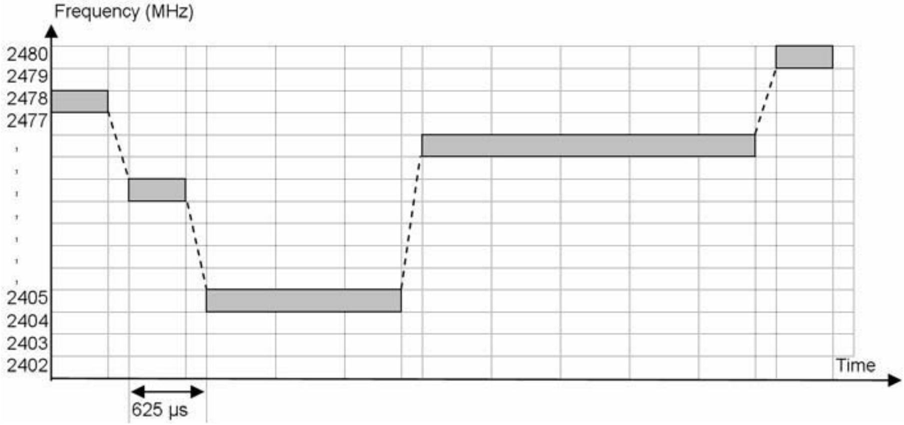


Figure 6.3. *Illustration of the frequency-hopping technique used by Bluetooth technology*

The frequency hops follow a pseudo-random sequence. To communicate with each other, all devices participating in the piconet use the same frequency hopping sequence. The channel hopping sequence is defined by the clock of the piconet master.

The modulation is Gaussian frequency shift keying (GFSK). In this type of modulation, a Gaussian filter is applied before employing a frequency shift keying modulation. Because the input signal is binary, a basic frequency shift keying (FSK) would cause quick transitions in frequency and therefore a wide bandwidth. The Gaussian filters smooth frequency deviations.

There are three classes of Bluetooth devices according to output power levels at the antenna connector. The maximum output power has an impact on the expected range. Table 6.1 describes the device power classes.

Power class	Maximum output power	Expected range
1	100 mW (20 dBm)	100 meters (328 feet)
2	2.5 mW (~ 4 dBm)	20 meters (66 feet)
3	1 mW (0 dBm)	10 meters (33 feet)

Table 6.1. *Bluetooth power classes*

Most of the device producers choose power class 3.

6.2.4. *Baseband*

The baseband is the architectural layer which manages physical and logical channels. It also provides multiple functions such as error correction, hop selection, flow control, security and power control.

The basic piconet physical channel is characterized by a pseudo-random hopping sequence through all 79 channels in the ISM band. The data transmission on a physical channel has a rate of 1 Mbps.

A time-division duplexing (TDD) scheme is used for full duplex transmission. The master and the slaves alternately transmit. The master transmission shall always start at even-numbered time slots, and the slave transmission shall always start at odd-numbered time slots. In addition, the channel used for the master-to-slave packet is used for the immediately following slave-to-master packet. Figure 6.4 represents this transmission mechanism with single-slot and multi-slot packets.

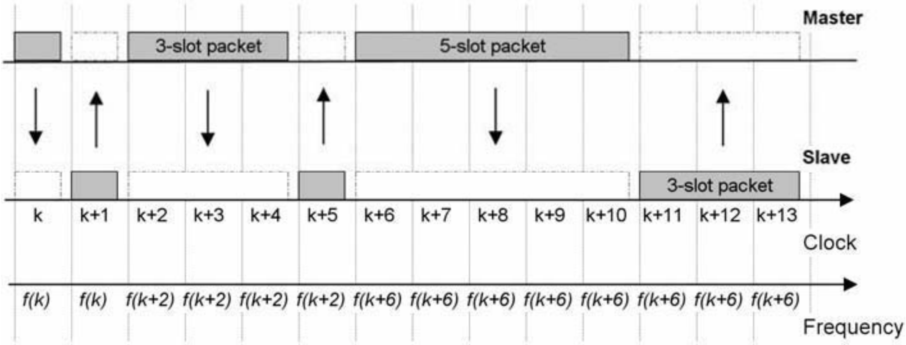


Figure 6.4. *Packet transmission between a master and a slave device*

The general packet format is shown in Figure 6.5.

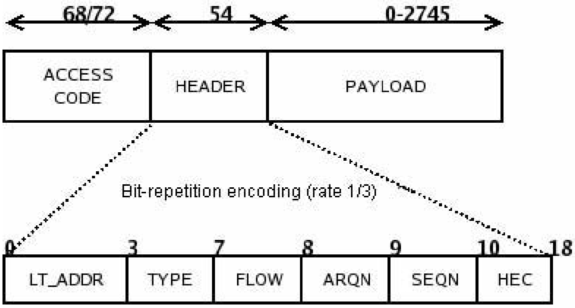


Figure 6.5. *General packet format in Bluetooth*

Each packet begins with an access code. If a header follows, the access code is 72 bits long, otherwise the access code is 68 bits long (shortened access code only). This access code is used for synchronization, definition of channel parameters and packet identification. All packets exchanged on the same physical channel have the same access code.

The packet header is 54 bits long. It contains link control information and is divided on six fields:

– LT_ADDR, a 3-bit logical transport address;

– TYPE, a 4-bit code which defines the type of the packets (SCO or ACL logical transport, slot occupancy, data or control packets);

– FLOW, a bit used for flow control of packets over the ACL logical transport;

– ARQN, a 1-bit acknowledgement code to inform the source of a successful transfer of payload data;

– SEQN, a sequential bit to distinguish odd and even messages;

– HEC, an 8-bit code to check the header integrity.

Finally, the payload part contains data. It ranges from zero to a maximum of 2,745 bits. Its format depends on the type of the packet.

6.2.5. *Link controller*

The baseband includes a link controller. The link controller defines how the piconet is created and how devices can be added to and released from the piconet. To support these functions, multiple states are defined.

By default, a device is in STANDBY state in which it economizes energy. If it wants to discover other devices, it enters in INQUIRY state in which it transmits discovery messages over different hopping frequencies. The devices which are in INQUIRY SCAN state can answer the discovery messages. Then the device which is in INQUIRY state obtain addresses and clock offset of the answering devices.

When the master device needs to establish a connection, it enters the PAGE state to synchronize with the slave. Since the master's clock is not automatically synchronized with the slave's clock, the master device cannot determinate when a slave will answer it and on which frequency. That is why the master transmits a set of identical messages on different frequencies and expects a slave's answer between two transmissions. The slaves in PAGE SCAN state listens on the frequency defined by the hopping sequence according to the device's address BD_ADDR and answer the master's request.

Once the connection is established, the devices are in CONNECTION state and data can be exchanged. A connected slave device may enter into many sub-states in which it is more and less active. When a slave does not need to communicate but wishes to stay synchronized on the physical channel, it enters PARK state. This state is not very active. The active address LT_ADDR of the slave becomes invalid and the slave obtains two addresses: PM_ADDR and AR_ADDR. The slave becomes "parked". Finally, when a device is no longer active, it enters STANDBY state.

6.2.6. *Bluetooth device addressing*

The baseband also defines device addressing. There are four types of addresses to identify a Bluetooth device: BD_ADDR, LT_ADDR PM_ADDR, and AR_ADDR.

BD_ADDR corresponds to "Bluetooth Device Address". It is a unique 48-bit device address. It is similar to an MAC (Medium Access Control) address. The address is given by the IEEE Registration Authority.

LT_ADDR means "Logical Transport Address". It is the 3-bit address assigned to each active slave in a piconet. This property explains the limitation in the number of active members in the piconet. This address is assigned by the master to the activated slaves. The all-zero LT_ADDR is reserved for broadcast messages.

PM_ADDR means “Parked Member Address”. It is the 8-bit address reserved to members in parked mode. When the slave is activated, it shall be assigned an LT_ADDR, but shall lose the PM_ADDR.

To allow parked mode, the master device keeps a slot to send synchronization data. Also, an access window is defined in order for a parked slave to send requests to become active in the piconet.

AR_ADDR means “Access Request Address”. This address is used by the parked slave to determine the slave-to-master half slot in the access window where it is allowed to send access request messages. The AR_ADDR is assigned to the slave when it enters the PARK state and is valid only for as long as the slave is parked. AR_ADDR is not necessarily unique; different parked slaves may have the same AR_ADDR.

6.2.7. *SCO and ACL logical transports*

The packets used on the piconet are related to the logical transports in which they are used. Three logical transports with distinct packet types are defined: the SCO logical transport, the eSCO logical transport and the ACL logical transport. The eSCO logical transport is an extension to SCO (Synchronous Connection-Oriented) logical transport (possibility of retransmission). We describe the SCO and ACL logical transport types. The logical transport type is defined by the field TYPE in the packet header.

The SCO logical transport is a symmetric and synchronous connection-oriented link. This logical transport type is appropriate in real-time communication such as voice communication. The SCO can be considered as a circuit-switched connection between the master and a specific slave. Indeed, the master maintains the synchronous logical transports by using reserved slots at regular intervals. To ensure the support of time-bounded information, SCO packets are never retransmitted. To improve reliability, the packets are checked for errors. The master may support up to three SCO links to the same slave or to different slaves. A slave may support up to three SCO links from the same master or two SCO links if the links originate from different masters.

ACL (Asynchronous Connection-Less) logical transport is an asynchronous connection-less link. The master may exchange packets with any slave in the slots not reserved for synchronous logical transports. This logical transport provides a packet-switched connection between the master and all active slaves participating in the piconet. Between a master and a slave, only a single ACL logical transport exists. To guarantee data integrity, packet retransmission can be applied.

6.2.8. *Link Manager*

The Link Manager is used for setup and control links between two devices. The Link Manager Protocol (LMP) is used to communicate and negotiate between the link managers on two devices that are connected by an ACL logical transport. The LMP is used to control and negotiate the Bluetooth connections. The LMP messages are interpreted by the Link Manager and processed by the link controller in the baseband.

For example, a connection can be established when a paging message is sent if the slave's address is known, or when a message of inquiry followed by a message of paging if the slave's address is not known.

The Link Manager also supports the security procedures like authentication, pairing, link key management and encryption. A device may use a security mechanism at the link layer. This security mechanism is started before any communication. The authentication and encryption are based on a shared secret key. The pairing procedure is used to generate this key when two devices meet for the first time.

The pairing procedure is based on a PIN code in order to restrict service access to only the allowed users. PIN means "Personal Identification Number". The initiator sends a request to the responder. If the PIN code is correct, the association is accepted. This procedure can imply a manual input by the user.

6.2.9. *HCI layer*

The lower layers that were presented earlier are integrated in the Bluetooth controller. The HCI layer provides a standard command interface to the Baseband controller and Link Manager. HCI stands for "Host Control Interface".

This layer ensures the interoperability between different implementations of higher layers and the Bluetooth controller. This is the interface between the host software and Bluetooth firmware.

There are three types of HCI messages: command messages, event messages and data messages. The command messages are used by the higher layers to command the Bluetooth controller. The event messages sent by a Bluetooth controller makes it possible to notify higher layers that a command was executed. The data messages are used to exchange data between lower and higher layers.

Figure 6.6 gives a functional view of the HCI.

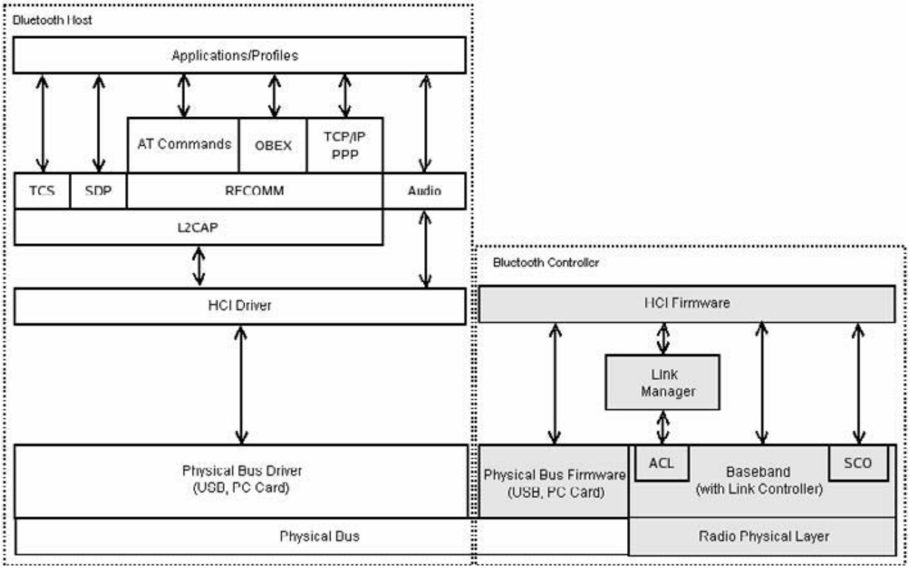


Figure 6.6. *The HCI*

6.2.10. *L2CAP layer*

L2CAP stands for “Logical Link Control and Adaptation Protocol”. It provides higher level protocol multiplexing, packet segmentation and reassembly. It also configures and controls Quality of Service (QoS). The L2CAP layer provides logical channels which are mapped to L2CAP logical links supported by an ACL logical transport.

There are three types of L2CAP channels: bidirectional signaling channels, point-to-point and bidirectional connection-oriented channels, and point-to-multipoint unidirectional connectionless channels. They are identified with the field CID (Channel Identifier) defined in the L2CAP protocol.

L2CAP supports upper layer data packets up to 64 KB in length. That is why there is a packet segmentation and reassembly operation. When establishing a connection, L2CAP negotiates the maximum size of payload data to prevent a buffer overflow on a device which uses smaller payloads. This functionality is necessary because not all devices are able to support a packet size of 64 KB.

L2CAP provides multiplexing for packets from higher layers. The higher layer protocol (RFCOMM for example) can be determined thanks to the PSM (Protocol/Service Multiplexer) field. The PSM field is equivalent to TCP/UDP ports in IP. For example, a PSM value equal to 3 means that the higher layer is RFCOMM. Vendor-specific PSM values are available. That is why a user may have trouble being sure that there is not a backdoor.

6.2.11. *Service Level Protocol*

This layer is a set of protocols providing a service to applications. The following protocols will be described: SDP, RFCOMM, TCS, AT and OBEX.

SDP stands for “Service Discovery Protocol”. It provides a means for applications to discover available services in remote devices and their characteristics (description, encoding, etc.). The service discovery may be started after a connection is established.

This service is a means of information. A service which is not listed in the remote SDP server can be used. For example, the DSL modems by Orange France supporting Bluetooth have a special stack where the available services are not listed. On the other hand, some implementations on phones or PDAs (Personal Digital Assistants) require the available services to be recorded on the SDP server.

RCOMM is the abbreviation of “Radio Frequency Communication”. The RFCOMM protocol provides emulated RS-232 serial ports over the L2CAP protocol. This service is based on RS-232 specifications. It provides the same type of data transfer as the serial ports and it supports up to 60 emulated ports.

Many devices communicate via the RFCOMM protocol. Bluetooth headsets may use RFCOMM. An authentication can be requested on some implementations of the Bluetooth stack and RFCOMM.

TCS stands for “Telephony Control protocol Specification”. It enables the services of telephony and is based on the SCO logical transport. The AT protocol consists of a command set to manage modems. AT is the abbreviation of the word “Attention”.

OBEX (for “OBject EXchange”) is a communication protocol that facilitates the exchange of binary objects between devices. It is used for example to exchange calendar data, address book or simple files. This protocol comes from IrDA ("Infrared Data Association") which defines standards for communication over infrared light. Therefore, OBEX is adapted to narrowband channels like Bluetooth.

OBEX is similar to HTTP in the concept and the functionalities because the clients use a reliable transport protocol to connect to a server. OBEX makes it possible to upload data by the PUSH command and to download data by the PULL command.

OBEX and HTTP differ in their transport protocol, transmission format and session support. HTTP is transported over a TCP/IP link while OBEX is implemented on a Bluetooth RFCOMM stack. HTTP sends human-readable text while OBEX uses a binary format that is easier to parse by devices with limited resources. Finally, HTTP transactions are stateless. In OBEX, a single connection may bear many related operations.

6.2.12. *Bluetooth profiles*

A profile defines a set of modules necessary when using Bluetooth applications. The specification of profiles is aimed at assuring interoperability between Bluetooth applications. The profile specification defines the way to implement a usage and the specific part of the Bluetooth protocol stack used. Each device supports at least one profile.

There are numerous profiles. The following list present some profiles:

– the Advanced Audio Distribution Profile (A2DP) is used to transfer audio stream. For example, it can be used from an MP3 player to a Bluetooth headset;

– the Audio/Video Remote Control Profile (AVRCP) provides a standard interface to control TV or High Fidelity remote equipment. It may be used with A2DP or VDP profiles;

– the Basic Printing Profile (BPP) allows devices to send texts, e-mails, vCards (electronic business cards) or other objects to a printer. It does not depend on printer drivers and it is appropriate to embedded devices, such as digital cameras or mobile phones, which have vendor specific drivers that are complicated to update;

– the Cordless Telephony Profile (CTP) allows cordless phones to communicate via Bluetooth. Mobile phones may be used as cordless phones connected to a gateway in a personal computer or a base station;

– the Dial-Up Networking Profile (DUNP) provides a standard to access the Internet via Bluetooth. It allows a connection to a mobile phone used as a modem. This profile is based on SPP (Serial Port Profile, described later) and implements a set of AT commands;

– the File Transfer Profile (FTP) provides access to the file system on a Bluetooth device. This includes support for listing files in a directory, sending or receiving files and deleting files. It is based on the GOEP profile (described below);

– the Generic Object Exchange Profile (GOEP) gives basic functions to exchange data between devices. It is based on OBEX;

– the Hands-Free Profile (HFP) is commonly used to allow hands-free kits, in the car for example, to communicate with a mobile phone. SCO is the logical transport to carry an audio signal;

– the Human Interface Device (HID) profile provides support for devices such as a mouse, joystick or keyboard. It provides a low latency link with a low power requirement;

– the HeadSet Profile (HSP) is suitable for the link between a headset to a device like a mobile phone. It is based on SCO logical transport for an audio signal and a subset of AT commands to control the channel (volume adjustment, ring, etc.);

– the InterCom Profile (ICP) proposes using devices like an interphone or walkie-talkie. This profile is based on TCS which is based on the SCO logical transport;

– the Phone Book Access Profile (PBAP) allows the exchange of phone book entries between devices. It may be used between a hands-free kit and a mobile phone to display the name of the incoming caller;

– the Serial Port Profile (SPP) uses RFCOMM. It emulates a serial port and provides a wireless alternative to applications based on the RS-232 standard;

– the Video Distribution Profile (VDP) allows the transport of a video stream. It can be used for streaming a recorded video from a media center to a portable player or from a camera to a television. Many video codec like H.263 or MPEG-4 must be supported.

6.3. Bluetooth security

Each type of wireless technology is vulnerable to security attacks. Most attacks are similar to the attacks on wired networks. However, it is generally easier to attack a wireless network since the access to physical media is open. A malicious user next to the victim is enough to start an attack.

The easiest and most well-known way to attack is to capture the signal and listen to the victim's communication. It allows the attacker to obtain confidential data, like passwords, and to access the user's data. To avoid this type of attack, cryptography is generally used, where the messages sent in the air are encrypted. The attacker is not able to decrypt the messages without the secret key.

Another type of attack consists of usurping the identity of a confident person and accessing the victim's data. These attacks are countered by using authentication methods.

Each protocol defines its security mechanisms in authentication and encryption. Each protocol has attacks and specific defenses. The following section will describe security mechanisms and attacks in Bluetooth.

There are different types of key in Bluetooth. The link key is a 128-bit random number. It is used during authentication process to derive the encryption key. The lifetime of a key depends on its type, i.e. whether the key is semi-permanent or temporary. A semi-permanent key can be used after the enclosure of a session. For example, it can be used to authenticate Bluetooth devices which share the key. The lifetime of a temporary key is limited to a session. At the end of the session, the key is rejected. The temporary keys are often used for point-to-multipoint connections.

The PIN code has a major role in Bluetooth security. It serves to authenticate the user. The length of the PIN code may vary between 1 and 16 bytes. However, there are often 4 bytes. An increase in this length is recommended for applications requiring a high security level.

6.3.1. *Security mode in Bluetooth*

There are three modes of security for Bluetooth access between two devices:

– security mode 1: non-secure;

– security mode 2: service level enforced security;

– security mode 3: link level enforced security.

In mode 1, the device is not secure. Authentication and data ciphering are not assured. This mode is used when malicious attacks are not expected and commodity is more important than security.

In mode 2, the security procedures are initialized only when a channel in the L2CAP layer is established.

In mode 3, the security procedures are started before a channel is established. This mode is integrated in the Bluetooth module. It is independent of the security mechanisms applied in the application layer. This mode provides authentication and data ciphering. It is based on a shared secret key for each pair of devices that want to communicate. The pairing procedure is used to generate the secret key when two devices communicate for the first time.

6.3.2. *Authentication and pairing*

In mode 3, a secure channel between two Bluetooth devices is established through pairing. The pairing contains three steps:

– creation of the initialization key (K_{init});

– creation of the link key (K_{AB});

– mutual authentication.

When these steps are finished, the devices can derivate a ciphering key in order to protect their communication.

Before the pairing procedure, a PIN code is set on the two devices. On some devices (wireless headsets for example), the PIN code is fixed and cannot be changed. In such cases, the PIN code is set on the other device. Therefore, two devices which have a fixed code cannot be associated and cannot communicate.

6.3.2.1. *Creation of the initialization key (K_{init})*

The initialization key may be symbolized by K_{init}. This key is based on the E22 algorithm (see Figure 6.7). K_{init} is generated from the following parameters:

– the Bluetooth Device Address BD_ADDR;

– the PIN code;

– a pseudo-random number IN_RAND.

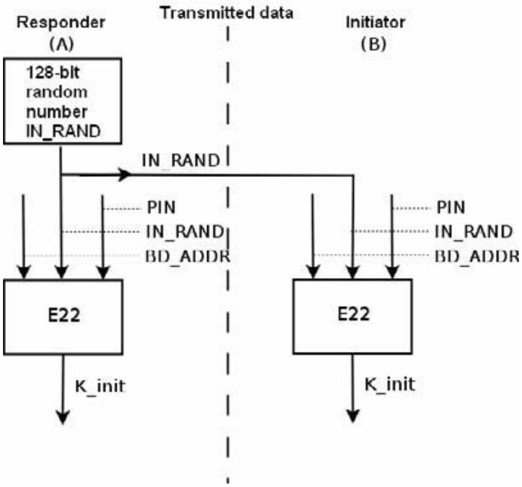


Figure 6.7. *Creation of the initialization key (K_{init})*

Data cannot be exchanged between the two devices. The initialization key is a temporary key. K_{init} is used to create the future key K_{AB}. It is not used for data transfer.

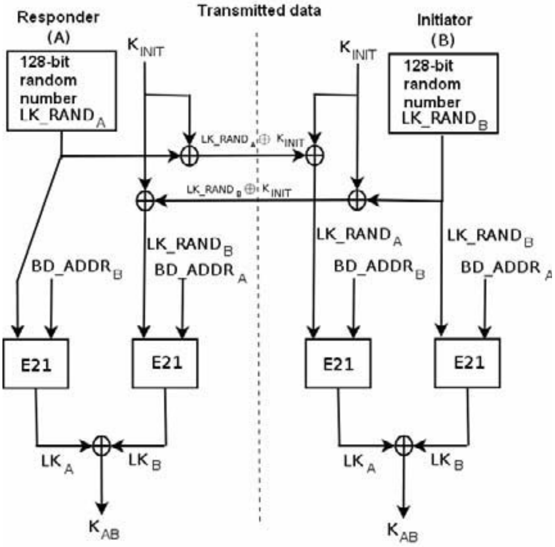


Figure 6.8. *Creation of the link key (K_{AB})*

6.3.2.2. *Creation of the link key (K_{AB})*

The link key K_{AB} is based on a new transfer of pseudo-random number. This transfer is ciphered with the key K_{init}. The XOR operator is used for encoding and decoding (Figure 6.8). Each device computes a result and then sends it. The E21 algorithm generates the key K_{AB} from the two device addresses BD_ADDR_A and BD_ADDR_B and the two pseudo-random numbers LK_RAND_A and LK_RAND_B.

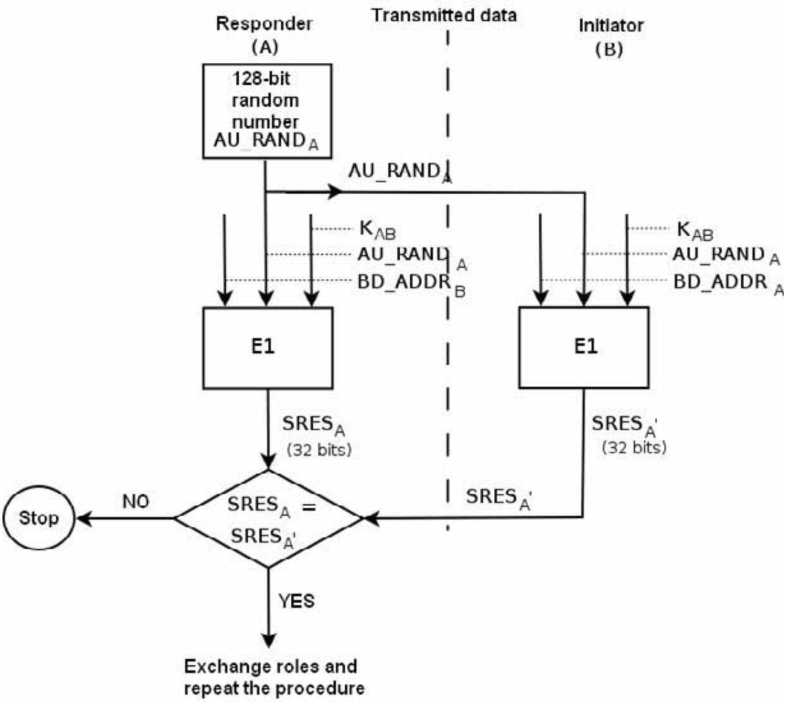


Figure 6.9. *Mutual authentication*

6.3.2.3. *Mutual authentication*

After the creation of the link key K_{AB}, a mutual authentication is performed (Figure 6.9). This operation is based on a challenge/response mechanism. The responder generates a 128-bit pseudo-random number, called AU_RAND_A and sends it. The initiator computes a 32-bit result, symbolized by SRES, from Au_RAND_A using the E1 algorithm. It sends the result to the responder. The responder computes the same algorithm and verifies the initiator's result. If the result is correct, the procedure may repeated in which the initiator becomes the responder and vice versa.

6.3.3. *Bluetooth encoding*

In Bluetooth, the transmitted data are ciphered with the E0 algorithm. This algorithm is a stream cipher to protect the communication. E0 generates a pseudo-random sequence which is combined with data through the XOR operator. The result is the ciphered message. E0 accepts a cipher key which may have a variable length. In general, the length of the key is 128 bits.

6.3.4. *Attacks*

6.3.4.1. *Attacks on the pairing [4]*

During the pairing, the values IN_RAND, AU_RAND$_A$, AU_RAND$_B$ and SRES are transmitted without encryption. Only the values LK_RAND$_A$ and LK_RAND$_B$ are encrypted (as a result of a XOR operation with the key K_{init}). Table 6.2 synthesizes the exchange during the pairing procedure. The values transmitted without encryption can be easily captured by an attacker sniffing Bluetooth packets.

In 2005, the researchers Yaniv Shaked and Avishai Wool showed that it can be possible to exploit this weakness and acquire the PIN code. Using an exhaustive attack, an attacker can test all possible PIN codes. For each tested PIN code, the attackers can execute an E22 algorithm (because IN_RAND and BD_ADDR are known) and obtain a hypothetical initialization key. This hypothetical key can be used to decrypt the second and the third message in order to calculate the supposed link key K_{AB}. Using K_{AB} and AU_RAND$_A$ (from message 4), the attacker can compute SRES and compare the result with SRES contained in message 5. Multiple PIN codes are tested until the two results correspond.

Yaniv Shaked and Avishai Wool showed that a PIN code with 4 figures can be acquired in less than 300 milliseconds by using a computer with a modest Intel Pentium III processor (450 MHz). With a Intel Pentium IV Hyper-Threading (3 GHz), they cracked a 4-figure PIN code in 63 milliseconds and a 7-figure PIN code in 76.127 seconds.

#	Src	Dst	Name	Length	Note
1	A	B	IN_RAND	128 bits	Plain text
2	A	B	LK_RAND_A	128 bits	XOR with K_{init}
3	B	A	LK_RAND_B	128 bits	XOR with K_{init}
4	A	B	AU_RAND_A	128 bits	Plain text
5	B	A	SRES	32 bits	Plain text
6	B	A	AU_RAND_B	128 bits	Plain text
7	A	B	SRES	32 bits	Plain text

Table 6.2. *Transmitted values during the pairing procedure*

This attack requires that the attacker is present during the pairing procedure and that it is able to sniff communication between the two devices. This condition restricts the attack possibilities. Shaked and Wool ameliorated this attack in order to crack a PIN code at any time. The idea consists of encouraging the two devices to replay the pairing procedure in the hearing of attackers. Thus, the attacker can capture the necessary data to start the attacks described above.

After pairing, the two Bluetooth devices do not need to generate a link key K_{AB} once again. After E21 computation, the link key is recorded by both devices. When they meet again, the devices may start directly with the mutual authentication step. Shaked and Wool discovered three malicious methods to force both devices to replay the pairing procedure.

The first method uses a functionality of LMP protocol. We assume that the devices directly start the authentication step. Thus, the responder sends to the initiator the value AU_RAND and waits for a correct result SRES. Shaked and Wool noted that the Bluetooth specification allows a device to forget or lose the recorded key. Then the initiator may send a message to the responder in order to inform it that the key was lost. This message is named "LMP_not_accepted". This attack consists of sending a wrong message "LMP_not_accepted" when the value AU_RAND is received. Then the responder believes that the key is lost and the pairing procedure will be replayed in vain in the hearing of the attacker.

The second malicious method is more difficult to do but can be accomplished it. When the two devices meet, the attacker sends the value IN_RAND before the responder sends a message AU_RAND. Then the initiator receives a wrong message IN_RAND and believes that the key of the responder is lost and the pairing procedure is replayed.

During the authentication step, the responder sends a message AU_RAND to the initiator and waits for a correct message SRES. After the message AU_RAND is sent, if the attacker manages to respond to a potentially wrong message with a random SRES, the authentication step will be repeated. After several trials, the responder will deduce an authentication failure and the pairing procedure will be started again.

These three methods allow the attackers to mislead two Bluetooth devices which had been perfectly secured by replaying the pairing in vain in the hearing of the attacker. Once the transmitted messages are recorded, the attacker can crack the PIN code in a very short time.

Until this attack appeared, it was recommended to start the pairing every time two devices met, even if they already shared a secret key. These recommendations are based on the hypothesis that there was no fault in the pairing. To conclude, Shaked and Wool's experiments prove that the repetitions of pairing between two devices must be avoided.

6.3.4.2. *Cryptanalytic attacks*

Bluetooth is a very popular piece of technology and today it is used everywhere in the world. This is why its encoding algorithm E0 was subjected to several attacks and cryptanalysis.

In 1999, Hermelin and Nyberg [5] proved that an E0 key can be discovered with 2^{64} operations instead of 2^{128} operations. This attack is based on the hypothesis that the attacker has a recording of 2^{64} transmitted bits. Fluhrer proved a theoretical attack to discover the key with a complexity of about 2^{65} operations after a computation of 2^{80} operations. Fluhrer concluded that the security of the E0 algorithm is equivalent to the security given by the 65-bit key and longer keys would not ameliorate the security of the algorithm.

In 2004, Lu and Vaudenay [7] published a statistical attack which recovers the encryption key with 2^{40} computations using the first 24 bits of 2^{35} payload frames. Then this attack was improved to 2^{37} operations for a first computation and 2^{39} for the actual key search.

In 2006, Éric Filiol proved some weaknesses in the Bluetooth core encryption algorithm E0. The complexity of the attack is estimated around 2^{35} operations.

6.3.4.3. *Attacks on the Bluetooth stack*

The vulnerabilities and attacks mentioned below do not result from security faults in the Bluetooth specifications. This section describes the security faults in Bluetooth implementations.

6.3.4.3.1. Bluetooth snarfing [8]

With some Bluetooth products, the connection with a device may be done without an alert. As a result, a device can access the data, including the phone directory, the agenda and the phone number identifying the device in the cellular network. The attacker can copy or erase entries in the phone directory. It can also execute remote commands on the victim's device.

This attack is possible on all devices which answer the discovery messages. These devices are said in visible mode. In order to counter this attack, it is recommended to deactivate some Bluetooth functionalities or set the device to invisible mode. According to [9], some malicious applications are also able to access invisible devices. This attack gaining access to information stored on selected mobile phones is due to an incorrect implementation of Bluetooth.

6.3.4.3.2. Bluejacking [9]

This malicious method consists of misusing the pairing procedure. The attacker manages to display an unsolicited message on any device which is located in its range. In the first steps of pairing, the name of the initiator's device is displayed on the responder's device. The Bluetooth specification allows a very long name up to 248 characters. As a result, the display of the name can be used to send a message.

This method has already been used in marketing campaigns to promote certain products.

6.3.4.3.3. Bluebugging [10]

In a Bluebug attack, the vulnerability exploited is the existence of a hidden RFCOMM channel on certain mobile phones that use Bluetooth technology. This channel enables connections without the initiator's authentication. The attacker establishes a serial port connection through the serial port profile of the victim's device and can access a set of commands. In particular, the attacker can phone from the victim's device, connect to the Internet, send or receive SMS messages, etc.

6.3.4.3.4. Bluetooth wardriving [11]

This attack allows locating the users with a Bluetooth device. Because each device broadcasts a unique 48-bit address, an attacker can identify a user and follow his localization. Many techniques can be used.

The attacker can install several devices in a zone. These Bluetooth spies can localize the close users. If the victim has set his device to visible mode, the Bluetooth spies may periodically broadcast requests and record the received response messages. The response message contains the device's address.

The localization of a Bluetooth device is possible if it is in invisible mode. The attack consists of listening to the communication between victim's device and others. The devices communicate using a specific access code. This code is generated from the master device's address. The complete addresses are transmitted in the packets for frequency hopping synchronization. However, these packets are transmitted only during the establishment of a connection.

To protect the devices against these attacks, an anonymity mode is necessary. A Bluetooth device must be able to periodically configure a new unpredictable address. However, collisions between addresses (when two devices have the same address) must be avoided.

6.4. Conclusion

Bluetooth is a comparatively recent technology. More and more devices support it. This technology allows short-range wireless communication (up to 100 meters) between many devices. The goal of Bluetooth is to specify an integrated circuit on a large scale that can be installed on a multitude of types of equipment and at a very low cost.

The applications of Bluetooth technology are numerous. We can name some examples: wireless communication between a cellular phone and a hands-free kit or a headset; wireless communication between a personal computer and different peripherals like a mouse, a keyboard or a printer; replacement of a wired serial port on medical equipment, etc.

The Bluetooth specifications have been written by considering the security issues. However, some problems concerning the pairing procedure have been discovered. Also, the encryption algorithm E0 is regularly questioned. Finally, numerous errors are the result of implementation issues. This implies Bluetooth insecurity, especially since Bluetooth can decompose into many sub-layers on which remote attacks may start.

To conclude, it is important to consider the discovered weaknesses on protocols and algorithms, thus the data at the application level must be secured. Finally, a good management of the devices is an answer to this insecurity. For example, it is recommended to deactivate all services which are not used. Deactivation of Bluetooth when it is not needed is also recommended!

6.5. Bibliography

[1] *Specification of the Bluetooth System version 1.2,* volume 0, 05 November 2003.

[2] B. MILLER, C. BISDIKIAN, *Bluetooth Revealed,* Prentice Hall, 2001.

[3] H. LABIOD, H. AFIFI, *De Bluetooth à Wi-Fi : sécurité, qualité de service et aspects pratiques*, Hermes, 2004.

[4] Y. SHAKED, A. WOOL, "Cracking the Bluetooth PIN"*, ACM/Usenix MobiSys 2005*, pp. 39-50.

[5] M. HERMELIN, K. NYBERG, "Correlation Properties of the Bluetooth Combiner Generator, Information Security and Cryptology", *Lecture Notes in Computer Science*, 1787, pp. 17-29, 1999.

[6] S. FLUHRER, *Improved key recovery of level 1 of the Bluetooth Encryption System*, Cryptology ePrint archive, 2002.

[7] Y. LU, S. VAUDENAY, *Cryptanalysis of Bluetooth Keystream Generator Two-Level E0*, ASIACRYPT 2004, pp. 483-499.

[8] A. LAURIE, *Serious flaws in Bluetooth security lead to disclosure of personal data,* URL: http://www.thebunker.net/resources/bluetooth.

[9] *BluejackQ*, URL: http://bluejackq.com.

[10] *Bluebug*, URL: http://trifinite.org/trifinite_stuff_bluebug.html.

[11] *Wardriving*, URL: http://www.wardriving.com/blue.php.

This page intentionally left blank

Chapter 7

Wi-Fi Security

7.1. Introduction

In order to make a secure communication in a Wi-Fi network, it is necessary to equip the environment with a certain number of functions that can be achieved either by the infrastructure by itself that is used to build the network or by adding new elements to it. To be more precise, it is essential to intervene with four main types of infrastructure elements; the infrastructure that allows the authentication of clients and network equipment, the hardware and the software that are necessary to achieve security on the radio interface, the network elements that are necessary for packet filtering and detection of attacks, and the equipment needed to manage remote access when users are moving:

– *Authentication infrastructure*. The IEEE 802.1x standard recommends the usage of the RADIUS server (Remote Authentication Dial-In User Server). Authentication can be conducted by a server located in the visited domain or outside it. This architecture establishes a trust circle, through which an authentication message is supported by multiple servers linked together by security associations.

– *Radio security*. Radio security's aim is to ensure the confidentiality, integrity and packet signature. These services are delivered by protocols such as WEP (Wired Equivalent Privacy), TKIP (Temporal Key Integrity Protocol) or CCMP (Counter mode with CBC MAC Protocol), standardized by the IEEE 802 Committee. The protocols use keys derived from a master key, after the authentication process.

Chapter written by Guy PUJOLLE.

– *Packet filtering*. The reliability of this operation is based on the packet signature using keys derived from the authentication process. Using this mechanism, the frames that enter the distribution system are safe (no risk of spoofing or disguise). Filtering systems (access point or portal) manage the privileges of IP packets (destruction of illicit packets) and make it possible to deliver and bill services for QoS (Quality of Service).

– *Access to remote services (roaming)*. Access to remote services may be designed generically under the VPN (Virtual Private Network) service. For example, implementation of secure inter-domains can be achieved using IPsec or SSL protocols.

7.2. Attacks on wireless networks

While listening on the wireless radio link is the obvious attack, other attacks also exist. This section summarizes some of these attacks. We also introduce the main methods that can be brought into wireless networks, algorithms and security protocols in order to stop such attacks. Normally, network security attacks are divided into passive and active attacks.

The risks associated with wireless networks based on IEEE 802.11 can be the result of one or more of these attacks. Consequently, these attacks may cause the loss of proprietor information, legal costs and recovery, a tarnished image and loss of network services.

7.2.1. *Passive attacks*

An attack is called passive when an unauthorized person obtains access to a resource without changing its content. Attacks may be passive eavesdropping or traffic analysis, sometimes called analysis of traffic flow. Both of these passive attacks have the following characteristics:

– *Eavesdropping*. The attacker listens to the transmissions in order to retrieve the content of messages. For example, a person listens to the transmissions over a LAN network between two stations or listens to transmissions between a wireless phone and a base station.

– *Traffic analysis*. The attacker obtains information by monitoring transmissions to detect types or classical models in the communication. A considerable quantity of information is contained in the flow of messages transmitted between both communicating parties, i.e., the transmitter and receiver.

7.2.2. *Active attacks*

An attack is called active when making unauthorized changes are made to messages and data flows or files. It is possible to detect this type of attack. Active attacks may take the form of one of the four following types, either singly or as a combination:

– *Masquerade.* The attacker impersonates an authorized user and obtains access to certain privileges.

– *Replay.* The attacker monitors the transmissions (passive attack) and retransmits messages to a legitimate user.

– *Message modification.* The attacker alters a legitimate message by deleting, adding, modifying or rearranging the contents.

– *Denial-of-service.* The attacker prevents or prohibits normal usage of the management of the communication medium.

The last type of attack is a formidable threat for software security solutions. This is due to the easily jeopardized security, in particular in case malicious modifications are possible in a software program responsible for implementing and controlling protocols.

7.2.3. *Denial-of-service attacks*

The denial-of-service attack is one of the simplest attacks to implement and is generally very difficult to deal with in wireless networks. Denial-of-service is achieved when the subject of the attack is inundated with messages and cannot respond to the demand. In the classical case, hackers employ large numbers of computers and send a continuous flow of messages that converge to the subject under attack. The parade is difficult since the attack can be sudden and it is difficult to predict such convergence.

In a wireless network, a denial-of-service consists of making a large number of requests to the access point until it crashes. Currently it is impossible to prevent a user from sending such types of request flows, even if he/she is not allowed to be connected. At each request, the access point must execute a series of instructions prior to the refusal. The only known counter-measure is to determine the point from which the attack is coming and launch a human neutralization.

Many denial-of-service attacks can be achieved using the ICMP (Internet Control Message Protocol). This protocol is used by routers to transmit supervision messages making it possible, for example, to indicate the reason of a problem to the user. A denial-of-service attack against a server consists of generating ICMP messages in bulk and sending them to the server from a number of important sites.

To flood a server, the easiest way is to send messages like ping messages asking it to return a reply. Server can be inundated by control messages of other types of ICMP.

7.2.4. *TCP attacks*

The TCP protocol works with some port numbers which determine a socket address, i.e. a network access point. This socket address is formed by the concatenation of the IP address and the port number. Each application has a port number, for example, 80 for an HTTP application.

A TCP attack may happen if the access point is forced to behave in the way defined by the attack. An attacker can use a classical port to enter a computer or a company network. The user opens a TCP connection on a port that corresponds to an application to run. The hacker starts to use the same port disguised as that user and send the responses. Eventually, it may extend the responses to that user so that he/she receives the requested information and suspects nothing.

We will see later in this book how firewalls are trying to address this kind of attack by blocking certain ports. Remember that every wireless network must be connected to the intranet of a company through a firewall that controls attacks from one side or another of the intranet. *A priori*, the firewall is rather meant to protect intranet attacks entering through the wireless network.

7.2.5. *Trojan attack*

In a Trojan attack, the attacker introduces into the terminal station a program that makes it possible to memorize the login and the password. This information is sent to the outside by a message to an anonymous mailbox. Various techniques may be used for this, from a program that replaces the login manager to a hacker program that spies on what is happening in the terminal.

This type of attack is fairly classic in wireless networks since a user can interfere with, via the access point, a PC and install spyware in it, allowing him/her to take the place of the user.

7.2.6. *Dictionary attacks*

Many chosen passwords are in the dictionary, so it is very easy for a machine to try them all. Many experiments have demonstrated the simplicity of this attack and have measured that the discovery of half the passwords of employees of a large company could be achieved in less than two hours.

A simple solution to address this attack is to complicate passwords by adding capital letters, numbers and symbols like !, ?, &, etc.

The dictionary attack is one of the most common attacks in wireless networks that are protected only by user passwords.

7.3. Security in the IEEE 802.11 standard

This section introduces security mechanisms implemented in Wi-Fi environments. These mechanisms are directly implemented in marketed hardware and not after the fact.

We begin by introducing three security mechanisms offered by the standards. We will notice that technological progress helping these encryption and signature mechanisms are not resistant to attack. The reason for this is that the designers of the standard have not opted for a sufficiently advanced technology to resist the effects of time.

Despite these limitations, there are some solutions to effectively protect a Wi-Fi network. This is particularly the case for WPA2 technology, implemented in the latest generation of products, or VPN and smart card mechanisms, which can be applied to existing products.

7.3.1. *IEEE 802.11 security mechanisms*

The access points used in wireless networks broadcast data to all stations in their emission range. As a result, a malicious user can enter the area of a network and retrieve information in order to obtain access to the network.

To overcome this problem, a client must establish a relationship, called an association with an access point.

A complete association with an access point requires the client to pass through three states:

1) non-authenticated, non-associated;

2) authenticated, non-associated;

3) authenticated, associated.

Figure 7.1 illustrates the states of authentication in 802.11 for a WLAN station. It describes the different states of a system and the transaction among these states. 802.11 exchanged frames may be of two types, data or management frames. To pass from one state to another, the WLAN station and the access point have to exchange management frames.

To authenticate a WLAN station in 802.11 wireless networks, a specific security mechanism, the WEP, has been defined.

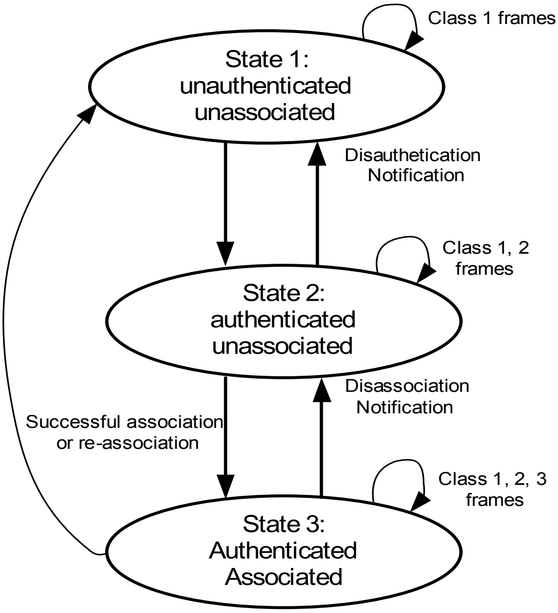

Class 1 frames
State 1: unauthenticated unassociated
Disauthetication Notification
Class 1, 2 frames
State 2: authenticated unassociated
Disassociation Notification
Successful association or re-association
Class 1, 2, 3 frames
State 3: Authenticated Associated

Figure 7.1. *State machine for authentication in a 802.11 network*

7.3.2. *WEP (Wired Equivalent Privacy)*

Since transmissions are broadcast on a radio wave, it is necessary to introduce a mechanism to protect communications from malicious eavesdropping. WEP is based on a symmetric cipher RC4 stream and was created to satisfy access control, privacy, authentication and integrity.

WEP is defined as an optional protocol, and the WLAN stations and the access points are not obliged to use it. The mechanisms defined in WEP are also optional: a station can use the authentication mechanism, for example, but not the encryption algorithm, and vice versa.

7.3.2.1. *Access control*

Access control is designed to control access and not to permit users without authorization access to the medium. Generally, access control has two functions: authentication and authorization. Authentication makes it possible to check the identity of the client who wants to be connected, while authorization gives him the permission to enter the network. It is possible to be authenticated but not authorized.

Access control can be done on both sides of the communication: client and server sides. If the client uses the access control server to enter the network, the reverse is also possible. By demanding the server to be authenticated, the client may allow (or not) the server to access its information. For example, when a client connects to the bank, not only can it verify that the server is indeed that of the bank but it can also give it more or less rights, for example, through applets.

7.3.2.2. *The SSID*

The network identifier or SSID (Service Set ID) is the first mechanism of security offered by WEP for network access control. The SSID is the name given to a network or domain. The term "network name" is primarily used at the network configuration.

All stations and all access points belonging to the same network must have the SSID, even if the WLAN stations are in ad hoc mode or in infrastructure mode. If one or more stations enter a network under the control of an access point, they must provide the SSID to the access point. WLAN stations can access the network if they have the correct SSID. The SSID is the only mandatory security mechanism in Wi-Fi networks.

7.3.2.3. *The ACL (Access Control List)*

Some Wi-Fi manufacturers implement the ACL on MAC addresses of the terminals. In this case, an access point performs the combination of a terminal only if the MAC address of the terminal is in its ACL. The MAC address is a unique address of every Wi-Fi or Ethernet card. According to this address, WLAN stations can be recognized in the network.

The ACL is an optional mechanism and can be configured only by the administrator of the access point. This option is rarely used because it is unreliable, as we shall observe.

7.3.2.4. *Confidentiality*

The transmitted frames in wireless networks are protected by encryption. Only the decryption using the proper static WEP key, shared between the terminal and the network, is allowed. This key is obtained by concatenation of a secret key of 40 or 104 bits and an initialization vector (IV) to 24 bits. It is dynamically changed for each frame. The size of the final key is 64 or 128 bits.

From the obtained key, the RC4 algorithm performs the encryption of data in stream cipher. The RC4 key has a length of 8 and 2,048 bits. The key is placed in a generator of pseudo-random numbers, called RC4 PRNG (Pseudo-Random Number Generator), from RSA Laboratories. This generator determines a sequence of pseudo-random bytes known as key stream or Ksi.

This series of bytes is used to encrypt a message, or Mi, with a classical Vernam protocol, performing exclusive XOR (⊕) between Ksi and Mi. The result obtained from the exclusive XOR gives a new value, called Ci, such that:

Ci = Ksi ⊕ Mi

In the WEP algorithm, the Mi is composed of data that are concatenated to the ICV (integrity check value). The encrypted frame is then clearly sent with its IV. The IV is an index which makes it possible to find the keystream, enabling us to decode the data. The encryption process is illustrated in Figure 7.2.

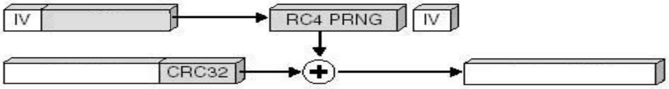


Figure 7.2. *Encrypting a WEP packet*

7.3.2.5. *Authentication*

Two types of authentication procedures are available in WEP: open authentication and shared key authentication, which is a method of challenge/response.

Open authentication is the default process. It contains no explicit authentication: a terminal can associate with the access point which is broadcasting its SSID and listen to all the data in transit within the BSS.

Shared key authentication provides a better level of security using a sharing key mechanism. The authentication occurs in four steps, as shown in Figure 7.3:

1) When a WLAN station requires an association with an access point, it sends an authentication request.

2) When the access point receives this frame, it sends to the WLAN station a frame containing a challenge of 128 bits generated by the WEP protocol.

3) The station copies the challenge in an authentication frame and encrypts it with the secret key, then its sends them to the access point.

4) The access point decrypts the message with the help of the secret key and compares it with the sent message. Then, it sends the result of authentication to the WLAN station.

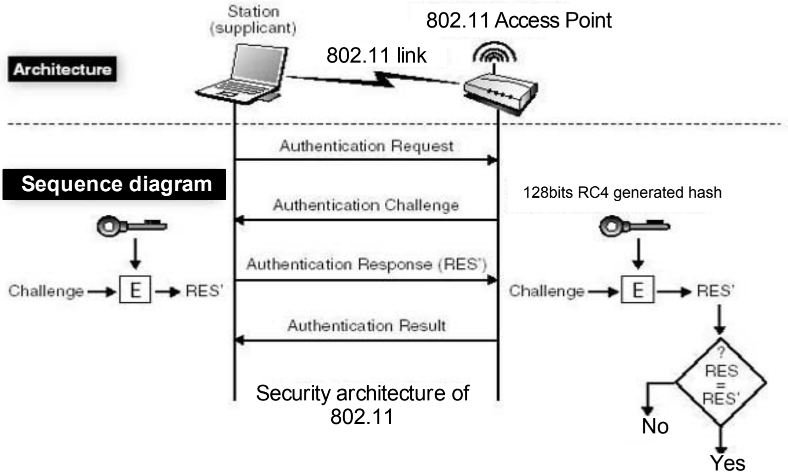


Figure 7.3. *WEP authentication*

7.3.2.6. *Data integrity*

The ICV is a CRC (Cyclic Redundancy Check) 32-bit based on the block. To prevent changes to the messages conveyed, the ICV is encrypted with the same key as that used for encryption.

7.3.3. *WEP shortcomings*

Even though the use of security mechanisms is a major step forward, Wi-Fi includes some shortcomings which make it easy to be attacked. Indeed, the set of WEP security mechanisms contains some weakness.

The shortcomings of WEP are not related to the RC4 encryption algorithm but to all implemented mechanisms, such as the initialization vector control or integrity. Each one of these mechanisms has defects, making it possible to break the WEP relatively quickly.

Regarding RC4, it was shown in August 2001 by Scott Fluhr, Itsik Mantin and Adi Shamir, in their article “Weaknesses in the Key Scheduling Algorithm of RC4”, that this algorithm can be broken and the shared secret key can be recovered. Since then, the method to crack a WEP key in just a few minutes has been further improved.

At the end, the weaknesses of WEP make it unreliable to manage privacy, authentication and data integrity.

7.3.4. *A unique key*

The original standard defines a key size of 40 bits, which is much too short to counter attacks by brute force, which would take no less than a dozen hours to break.

Since then, all manufacturers identified a key size of 104 bits, for what is called WEP 2, which is much more resistant to brute force attacks. In WEP, the key management is static, one secret key is shared by all stations in the network and the access point. If all the stations use the same key, it is even easier for an attacker to retrieve the data, hence the role of the IV in the WEP, which make it possible to define different encryption flows for the same shared secret key.

Another major drawback is that WEP does not prevent the replay. The shared secret key is manually configured at the stations and access point and is almost never changed. An attacker is not obliged to carry out an attack to recover the key as soon as possible. It is enough to build, day after day, a database of elements transmitted over the network and recover the shared secret key.

7.3.5. *IV collisions*

The collision attacks are passive attacks, which can break the key from collected clear text data. This type of attack is mainly based on the WEP functioning, including IV collisions, as well as the weaknesses of RC4. As explained above, the shared secret key defined in the WEP is static and almost never changes. The IV is concatenated with this key in order to create different flow encryption. The IV is 24-bit and there may be up to 2^{24} or 16 million different keys.

The low number of IVs is one of the weaknesses of WEP. If there is a collision of IVs, the encryption flow used is the same, since the IV and the shared secret key do not change. The probability of breaking the algorithm is proportional to the increase in IV collision.

Directly linked to collisions, the major drawback of the IV is its implementation. The IEEE has not specified how it should be implemented and has left this to manufacturers. Some of them define the IV to 0 when initializing the card. They then will increase by 1 with each transmission and reset all 2^{24} transmissions (maximum number of IVs) to 0.

If we assume that the IV is set to 0 when connecting to a station and then incremented by 1 at each transmission, the traffic is constant over the network with a throughput of 11 Mbps, giving a maximum speed of 7 Mbps, and the average size of a frame is 1,500 bytes, the following calculation:

1,500 bytes x 8 bits x (1/7 Mbit/s) = 0.00171 s

shows that a frame is sent on average every 1.71 ms. In fact, the time of issuance itself is shorter, the value of 1.71 taking into account the broadcasting time of the supervision messages sent by the access point. This calculation does not take into account the possible interference or collisions arising. These lead to a drop in the flow and consequently an increase in transmission time. Moreover, the size of a frame may be less than 1,500 bytes. In this case, its transmission time is less than 1.71 ms.

Since there are 2^{24} possible IVs, it is sufficient to listen to the traffic for about 8 hours for an IV collision to happen:

$0.00171 \text{ s} \times 2^{24} = 28{,}761 \text{ s} = 8 \text{ h}.$

In some cases, the IV can be randomly assigned. Although this seems more reliable, according to the *birthday paradox*, there is a chance that the same two IVs appear every 4,823 frames, or after 8 s, and 99 chances out of 100 that they appear every 12,430 frames, or after 21 s.

The fact that the IV is transmitted in clear figures in the frame can also be seen as a weakness, hence it is enough to listen to the network for some time to recover enough frames encrypted with the same IV and therefore with the same key.

It then performs an XOR (exclusive or) between two of these encrypted frames. This is equivalent to an XOR between the two clear texts. If IV is the initialization vector, K the RC4 secret key of 40 or 104-bit and RC4(IV||K) the secret key of 64 or 128 bits:

$C1 = P1 \oplus KS$ where $KS = RC4(IV \,||\, K)$

$C2 = P2 \oplus KS$

$C1 \oplus C2 = (P1 \oplus KS) \oplus (P2 \oplus KS) = P1 \oplus P2 \oplus KS \oplus KS$

since $KS \oplus KS = 0$, then:

$C1 \oplus C2 = P1 \oplus P2$

It remains to separate the two clear texts. It is possible to retrieve them because there are many redundancies in the data sent. It is just necessary to launch a so-called “known plaintext” attack. The encrypted data correspond to the frame LLC, which is encapsulated in the IP packet containing the TCP or UDP segment of user data. All these frames, packets and segments have the headers to allow correct forwarding of data. An attacker can force a user to send a text. For example, it sends an e-mail and waits for the user to synchronize his email. He would have just to find a small portion of the plain text that it can deduce quite easily since the IP, TCP and UDP headers are highly predictable.

The XOR of two encrypted texts with the same encryption may be sufficient protection, but it is actually not. In fact, it is sufficient to listen to the network for longer and wait for a new IV collision to deduce clear text. The RC4 key should be changed at least every 2^{24} packets, otherwise the data is vulnerable to IV collisions.

J. Walker discussed the mechanism of the initialization vector as prevention against the reuse of key and concluded that the way WEP uses the RC4 generates a significant reuse of IV and thus of the keystream, therefore making it ineffective. A moderately busy network can finish the space of the IV in a few hours, sometimes within minutes. The fact that the access points are configured for most shared key mode only exacerbates the situation. Even if an anti-collision system is used, the size of the IV is too small to prevent collision.

7.3.6. *RC4 weakness*

WEP has a drawback, which is linked to the RC4 algorithm itself.

The key used by the RC4 algorithm in WEP is a concatenation of the IV and the shared secret key. There are classes of weak key in RC4, in which a pattern in the first three bytes of the key causes an equivalent pattern in the first bytes of the keystream. The RC4 key of WEP uses the IV values, called resolvents, of the shape (3 + B, 255, N), B being a byte of the shared secret and N any value between 0 and 255. About 60 resolvents values are enough to find a byte of the shared secret. A quick calculation shows that we obtain a resolvent value every 216 frames, or 60 occurrences after about 4 million (2^{22}) packets.

The number of frames needed to obtain a key to 40 bytes is 60×216×40 = 518,400 frames. For a key of 104 bytes, we obtain 1,347,840 frames. However, the first three bytes (24 bits) of this key corresponds to the IV, which is sent in clear text in each frame.

This drawback, which facilitates the deduction of the key by statistical attacks, is based on the fact that the encrypted data corresponds to the frame with the known header. This attack is completely passive and relies on the use of a specific IV class. The keystream obtained with IV reveals information about the secret key. An attacker can determine it by processing enough packets.

Weak keys number about 1,280 for a 40-bit key and 3,328 for a 104-bit key. When the key size increases, the number of corresponding weak keys also increases in a linear and not exponential fashion.

This devastating attack, combined with an active attack to generate enough traffic, makes it possible to retrieve the encryption key in less than 10 minutes. It is on this vulnerability that hackers are using tools such as AirSnort to recover the WEP key.

Some vendors of access points have removed the IV to reduce the effectiveness of passive attack.

7.3.6.1. *The SSID*

The SSID is used to access the wireless Wi-Fi network. It is sent periodically in clear text by the access point in beacon frames. It is quite easy to recover the SSID, either through a sniffer tool, which makes it possible to retrieve all the data on a network, or software such as NetStumbler.

A new feature ensures that the SSID is transmitted in clear text by the access point over the network. This mechanism, called Closed Network, prohibits the transmission of the SSID through the beacon frame. When the network is closed, the user must manually enter the network name (SSID), while in an open network, the user station directly joins the access point without having to manually configure the SSID.

Even if the network is closed, the SSID can be recovered by other means. Indeed, the SSID is transmitted in clear text during the association phase (REQUEST) of a station with the access point, sniffing the network being enough to retrieve it during the association of the station.

Another disadvantage of the SSID is the name given to it by the manufacturers. The SSID is usually preconfigured at the access points. Each manufacturer uses and names a default SSID, such as WaveLan Network at Lucent or Tsunami at Cisco Systems.

If the SSID is not changed by the user, any person who knows the brand of the access point may attempt to use the default SSID to access the network. Moreover, if the SSID is not modified, it is likely that the password used to configure the access point is not modified either.

7.3.6.2. *ACL*

The first drawback of the ACL is that it is optional and very rarely used. Moreover, even if a person has an MAC address which is not in the ACL, it can always listen to the network and identify authorized MAC addresses which are transmitted in clear. Once authorized MAC addresses are known, it is possible to substitute its own MAC address with an authorized MAC address, which most drivers of Wi-Fi cards allow.

7.3.7. *Attacks*

7.3.7.1. *Replay attack*

The flaw in the authentication mechanism *shared key authentication* is due to the properties of XOR. When a user authenticates using this mechanism, the access point sends him a clear text, or Challenge Text, that the user must encrypt to prove that he possesses the same shared secret key as the access point. The attacker that wants to authenticate will just have to listen to the dialogue between the user and the access point, obtain the Challenge Text message (P) by the access point and the Challenge ciphertext (C) sent by the user, then reuse it.

Having recovered C and P, it is easy to deduce the encryption flow KS. An attacker tries to authenticate by simply sending a request to authenticate to the access point and then waits for it to send back a Challenge Text. Once it is received, the attacker encrypts it using the previously calculated encryption flow KS. He forges one 802.11 frame, which incorporates the Challenge Text added with the calculated FCS of the frame so that it will be validated by the access point. The access point will not notice anything and hence authenticates the attacker.

7.3.7.2. *Denial-of-service attack*

The purpose of an attack is not necessarily to break an encryption algorithm to retrieve the key and listen or enter the network. Some attacks have the sole function of sabotaging the network by preventing its operation. This attack, called denial-of-service (or DoS), is widespread in all types of networks. In Wi-Fi networks, the easiest DoS is noise interference. Networks operating in the 2.4 and 5 GHz frequency bands, using a radio device that uses the same band with a higher signal transmission than Wi-Fi can cause interference and thus a drop in overall performance network, or even prevent it working completely. This attack is the easiest to implement. It is also unfortunately unstoppable.

The good functioning of the network is based on the transmission of control and management frames. However, these frames are not authenticated. So, it is possible to disrupt the network by changing certain attributes of these frames, any changes resulting in a malfunction of the network.

The following two examples illustrate the DoS attack:

– *Disauthentication and disassociation frames*. These types of frames allow either disauthentication or to disassociation from an access point. An attacker can use one of these messages to behave as the access point (rogue AP) or a station in order to disconnect a given station from the network, which must then reconnect. The massive transmission of this type of message can prevent the reconnection of the station.

– *Reservation mechanism*. Media reservation is based on sending RTS/CTS frames. When the media is restricted to transmission between a source and a destination, the source station sends an RTS frame, which is received by all network stations. If the RTS is not destined to them, these stations will extract the Duration /ID field which gives the time occupation of the media in order to determine the reservation duration. After this period, the stations believe that the media is no longer reserved and try to access the media again if they have data to send. If an attacker sends an RTS frame by including in the Duration/ID field the time of maximum occupancy (32 ms) and renew the sending of the frame every 32 ms, it prevents access to the support of all stations in the cell, and no transmission is possible, again a DoS.

7.4. Security in 802.1x

Wireline or wireless local area networks are often deployed in environments that allow unauthorized equipment to be attached or unauthorized users to access the network using attached equipment. For example, in some public areas where buildings are accessible to the public, a corporate network can provide connectivity to the LAN. In such environments, it is desirable to restrict access to services offered by the local network to authorized users and equipment only.

Originally designed for the management of secure access to wired packet switching networks, the IEEE 802.1x authentication protocol, or Port Based Network Access Control, makes it possible to block the flow of data from an unauthenticated user. It became the most important standard for authentication today; it has been also applied in wireless networks: a client who cannot be successfully authenticated will be rejected by the access point of the wireless network; if this one is 802.1x compliant, or by a controller located between the access point and the corporate network that monitors the incoming flows.

This section examines the operation of the 802.1x protocol in detail.

7.4.1. *802.1x architecture*

The 802.1x architecture relies on the three functional entities (see Figure 7.4):

1) The supplicant or client 802.1x. This is a terminal wishing to use the resources offered by a communications network.

2) The authenticator or controller. This system controls a port for network access. It may be a switch in a wired network or access point in a wireless network. The flow of 802.1x client data is divided into two classes of frame:

– The frames used by the EAP (Extensible Authentication Protocol), defined by RFC 2284 in March 1998.

– Other frames that are blocked when the port is in the “not authorized” state. If successful in the authentication process, the port passes to the “authenticated” state and offers a free passage to all frames, meaning all user services.

3) The authentication server, typically RADIUS (RFC 2865 in June 2000). It is responsible for performing the authentication process with the 802.1x client. During this phase, the authenticator does not foster dialogue between the two entities but acts as a simple passive relay.

For clarity, in the next sections we will call the user station port the client 802.1x port, or supplicant, and the access point the access point port or authenticator. The authenticator has two ports: a port not controlled which, if chosen, does not control the traffic, and a controlled port that allows (or not) authenticated users’ packets to pass.

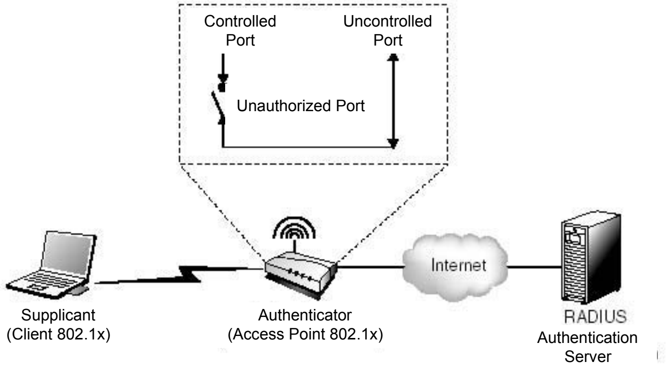

Controlled Port
Uncontrolled Port
Unauthorized Port
Internet
Supplicant (Client 802.1x)
Authenticator (Access Point 802.1x)
RADIUS Authentication Server

Figure 7.4. *802.1x architecture*

7.4.2. *Authentication by port*

The 802.1x standard defines a control network access based on ports. Its function is to authenticate and authorize equipment attached to the port of a local network.

In IEEE 802.11 wireless networks, a port is an association between a station and an access point. The controlled port behaves like a switch with two states. In the unauthorized state, only the frames dedicated to EAP authentication are not blocked. In the authorized state, the flow of information passes freely. We will describe the EAP protocol a little later on.

The 802.1x standard defines encapsulation techniques used to carry EAP packets between the client 802.1x port and access point port or switch. These ports are called PAE (Port Access Entity). The encapsulation is known as EAPoL (EAP over LAN). EAPoL indicates the beginning and the end (optional) of an authentication session with the notification messages EAPOL-START and EAPOL-LOGOFF.

In the authorized state, the port controls the duration of the session, meaning the time that we consider the client remains authenticated without asking to re-authenticate, using the reAuthPeriod variable, whose value default is 3,600 s. Typically, the access point retransmits lost EAP frames every 30 s. Meanwhile, the 802.1x client retransmits EAPOL-START frames not acknowledged every 30 s by a EAP-REQUEST IDENTITY message.

7.4.3. *Authentication procedure*

In wireless networks, the EAP protocol is used in a transparent manner between the station and the authentication server through an access point. It is encapsulated first in EAPoL frames then in the RADIUS protocol, which is routable since it is transported over IP.

Basically, the insertion of a wireless terminal in an 802.1x environment occurs as follows:

1) The station authenticates first, then gets associated with an access point, which is identified by its SSID (a chain of 32 characters or less).

2) To begin the authentication, the station broadcasts a EAPOL-START frame every 30 seconds.

3) The access point sends a REQUEST.IDENTITY-EAP message to the 802.1x client, which in turn produces a EAP-RESPONSE.IDENTITY response containing the identity (EAP-ID) of the wireless terminal.

4) From this parameter, the access point deduces the IP address of the authentication server and sends to this server the EAP-RESPONSE.IDENTITY message encapsulated in a RADIUS request. Other possibilities were implemented in the access point, as successive interrogation RADIUS servers until it finds the node corresponding to the IP address.

5) Therefore, request and response EAP messages are exchanged between the RADIUS server and the 802.1x client, the access point playing only the role of passive relay.

6) The RADIUS server indicates the success or failure of this procedure through a EAP-SUCCESS or EAP-FAILURE message. Based on this information, the port conveys in the authorized or unauthorized state.

7) At the end of the authentication process, the RADIUS ACCESS-ACCEPT message causes a transition in the state of the port to authorized. The RADIUS ACCESS-REJECT message forces the concerned port to the unauthorized state. A port retains its current status during an authentication session.

8) In cases where authentication is successful, the client and the 802.1x authentication server calculates a session key, called the Unicast Key. In the Microsoft environment, this value is a pair of keys 2x32 bytes (these attributes were defined by RFC 2548 in March 1999). The authentication server sends it to the

access point in the MS-MPPE-SEND-KEY and MS-MPPE-RECV-KEY attributes of the RADIUS Access-Accept message.

9) The access point then selects an encryption key, called a Global Key, to the security association with the 802.1x client. The latter is encrypted and signed with the session key received from the RADIUS server and then delivered to the 802.1x client in a EAPOL-KEY frame (see draft congdon-radius-8021x-29.txt, April 2003).

The authentication procedure is shown in Figure 7.5 below.

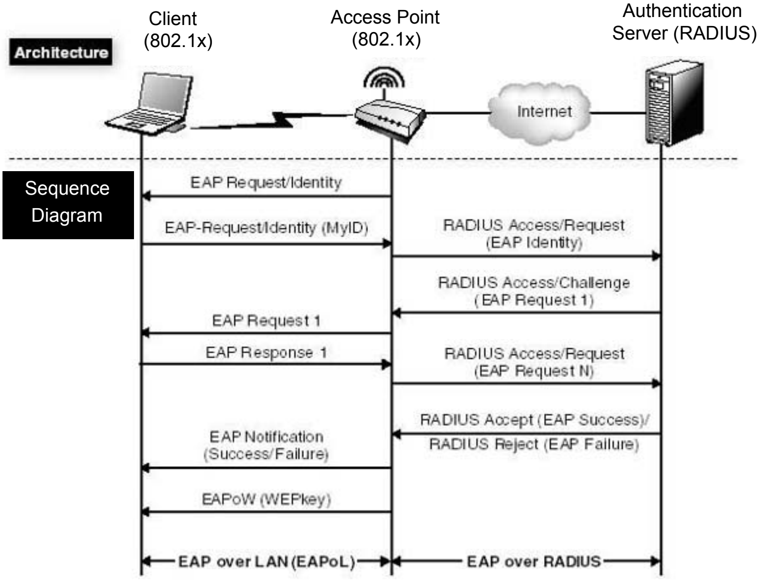

Architecture
Client (802.1x)
Access Point (802.1x)
Authentication Server (RADIUS)
Internet
Sequence Diagram
EAP Request/Identity
EAP-Request/Identity (MyID)
RADIUS Access/Request (EAP Identity)
RADIUS Access/Challenge (EAP Request 1)
EAP Request 1
EAP Response 1
RADIUS Access/Request (EAP Request N)
RADIUS Accept (EAP Success)/
RADIUS Reject (EAP Failure)
EAP Notification (Success/Failure)
EAPoW (WEPkey)
EAP over LAN (EAPoL)
EAP over RADIUS

Figure 7.5. *Authentication procedure*

7.5. Security in 802.11i

We stressed in section 7.3 the WEP weaknesses of the 802.11 standard and showed that 802.1x defines a framework for authentication, but did not specify the method for distributing keys in detail. Moreover, as the client does not participate in the calculation of the global key, there is no procedure for mutual authentication between the client and the access point that profits from the existence of a shared secret in the form of a unicast key.

The IEEE 802.11i working group developed an architecture designed to bridge these gaps. The first Industrial Committee, the Wi-Fi Alliance, formerly WECA, published on 29 April 2003 recommendation WPA (Wi-Fi Protected Access), based on a subset of the IEEE 802.11i standard. This version of WPA can be considered as a second-generation standard for the security of wireless networks. Implemented in products since the beginning of 2004, WPA has not been a great success because of its intermediate status. It is however important to note that this second generation is compatible with the Wi-Fi equipment market and there is a change in firmware to operate.

Finalized in June 2004, the 802.11i standard is a more important step since it establishes how to secure a wireless network for years to come. As we will see, this third generation standard, WPA2 stamped on products, is compatible neither with the first nor the second generation and therefore calls into question all previous investments in security, which can be relatively heavy. This non-compliance comes from the use of the AES encryption algorithm, which cannot be loaded into the firmware of the Wi-Fi cards.

The contributions of 802.11i can be classified into three categories:

– definition of multiple protocols for radio security;

– information elements allowing to choose one of them;

– a new method for distributing keys.

The standard is based on the 802.11 wireless networks using 802.1x for authentication and the calculation of a master key, called the PMK (Pairwise Master Key). In the case of ad hoc mode, this key, called the PSK (Pre-Shared Key), is distributed manually.

This section examines the mechanisms to implement security for a second and third generation. The fundamental difference between the two generations is the encryption algorithm used.

7.5.1. *The 802.11i security architecture*

As explained previously, 802.11i defines two new generations of security for Wi-Fi networks. The standard begins by defining a network with a strong security, called the RSN (Robust Security Network). We will first look at the information necessary to ensure this security and explain the retained mechanisms by the standard bodies to implement them.

7.5.1.1. *RSN (Robust Security Network)*

The authentication services and the key management of a RSN are based on the 802.1x standard. The RSN provides access control based on a strong authentication of the higher layers.

The role of the RSN is to guarantee security and mobility, integrity and confidentiality like scalability and flexibility.

Security and mobility

The security architecture allows client authentications independently of whether he is in his local network or in a foreign network. An architecture equipped with a centralized authentication server can satisfy this requirement, the client no longer has to be concerned with the access point he is associated with. Other solutions propose a distributed authentication server or even a deputy server, which can be used in certain situations, in particular in the case of connection problems with the central computer in charge of the authentication. For example, an oil platform which would have lost the satellite communication with its central location could continue to function in an autonomous way. In such cases, total coherence must be maintained regularly so that a client who would have lost his authorization can get connected to another unsynchronized site.

Integrity and confidentiality

Each 802.11i access point has an authenticator role; it shares a secret with the RADIUS server with which it communicates. This secret is used to calculate a digest HMAC-MD5 of the RADIUS packets exchanged, i.e. a field of binary characters with determined length, calculated using the packet to be sent and the shared secret. Each RADIUS packet contains a field named REQUEST AUTHENTICATOR, which is the HMAC-MD5 calculation of the packet and this secret. This field is inserted in the RADIUS packet by the RADIUS server and is checked by the access point.

In the other direction of the communication, the RADIUS server checks the attribute EAP AUTHENTICATOR present in the RADIUS packet using the EAP MESSAGE attribute. These two attributes offer the possibility of a mutual authentication per packet and preserve the communication integrity between the RADIUS server and the access point.

As explained previously in this work, it is easy enough for an attacker equipped with the adequate reception tool to listen to the traffic between the stations monitoring the radio links. The security architecture suggested by 802.11i aims at providing the guarantees of a strong confidentiality. Moreover, it defines a dynamic key distribution mechanism.

Scalability and flexibility

The security model is extensible in terms of number of users and their mobility. A user who moves from an access point to another can be quickly re-authenticated in a protected manner.

The wireless networks deployed in companies or in public places have a strong need for confidentiality. To satisfy this need, the security architecture must be flexible, in order to facilitate the administration and to take into consideration the network deployment environment.

By separating the access point or authenticator from the authentication process itself, the RSN allows access on a number of access points. Flexibility is brought by the fact that optional messages EAPOW KEY (EAP over Wireless), similar to the EAPOL KEY term used on fixed networks, can be deactivated for a particular deployment in which data confidentiality is not necessary. The 802.11i model specifies how the RSN interacts with 802.1x. Two types of protocols ensure the security at the MAC level:

– TKIP (Temporal Key Integrity Protocol);

– CCMP (Counter-mode/CBC-MAC Protocol).

A TSN (Transition Security Network) supports the previous architectures; it means pre-RSN, in particular the following mechanisms, imported from the IEEE 802.11 standard:

– Open Authentication;

– Shared Key Authentication;

– WEP (Wired Equivalent Privacy).

A RSN network must support the CCMP protocol. It can also ensure a migration of the previous WEP networks implementing the TKIP protocol. In other words, the standardization bodies, instead of directly imposing the third generation, which is incompatible with the first, propose passing by an intermediate stage, WPA, using TKIP to guarantee excellent security while waiting for the change to the third generation. Obviously, nothing prevents clients wishing to establish Wi-Fi wireless networks from directly adopting the third generation.

Figure 7.6 illustrates the various security levels of 802.1x architecture without 802.11i.

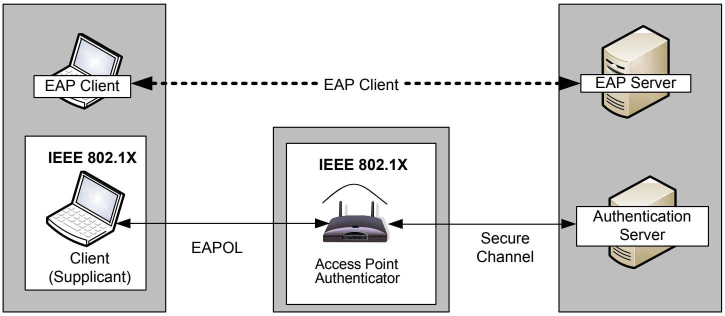


Figure 7.6. *Security levels in 802.1x architecture*

7.5.1.2. *Security relations in RSN*

The 802.11i access point and the authentication server perform a mutual authentication and establish a protected channel. The 802.11i model does not describe the methods used to conclude this operation; protocols such as RADIUS, IPsec or TLS/SSL can be implemented.

The 802.11i client and the authentication server are mutually authenticated using the EAP protocol and generating a master key, or PMK (Pairwise Master Key). The elements of this procedure are transported by the secure channel, where the cryptographic parameters must be different for each 802.11i client.

The PMK key is divided between the 802.11i client and the access point. They use a protocol with four stages, or 4-ways handshake, based on EAPOL-KEY messages in order to perform the following operations:

– confirmation of the PMK existence;

– confirmation of the PMK startup;

– calculation of the PTK (Pairwise Transient Key) starting from the PMK;

– installation of ciphering keys and the 802.11 frame integrity;

– operation confirmation of 802.11 keys.

The GTK (Group Transient Key), transmitted via EAPOL-KEY packets from the access point towards the 802.11i client, allows the client to exchange messages in broadcast mode and optionally in unicast mode.

In the case of the PSK mode, the PMK key is preinstalled between the 802.11i client and the access point.

7.5.2. *Security policy negotiation*

An access point diffuses in its beacon or probe frames data elements, called IE (Information Elements), in order to notify the 802.11i client of the following indications:

- list of supported authentication infrastructures (typically 802.1x);
- list of security protocols available (TKIP, CCMP, etc.);
- ciphering method for key group distribution (GTK).

A 802.11 station notifies its selection using a data element inserted in its request for association. This step is illustrated in Figure 7.7.

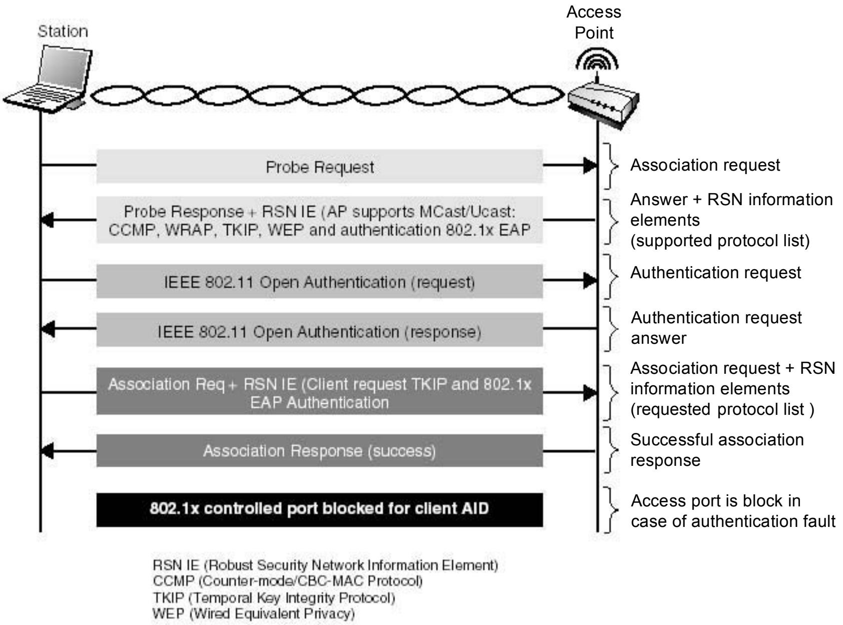

Station
Access Point
Probe Request
Probe Response + RSN IE (AP supports MCast/Ucast: CCMP, WRAP, TKIP, WEP and authentication 802.1x EAP
IEEE 802.11 Open Authentication (request)
IEEE 802.11 Open Authentication (response)
Association Req + RSN IE (Client request TKIP and 802.1x EAP Authentication
Association Response (success)
802.1x controlled port blocked for client AID
Association request
Answer + RSN information elements (supported protocol list)
Authentication request
Authentication request answer
Association request + RSN information elements (requested protocol list)
Successful association response
Access port is block in case of authentication fault
RSN IE (Robust Security Network Information Element)
CCMP (Counter-mode/CBC-MAC Protocol)
TKIP (Temporal Key Integrity Protocol)
WEP (Wired Equivalent Privacy)

Figure 7.7. *Security policy negotiation*

7.5.3. *802.11i radio security policies*

The WEP was insufficient to ensure the security of 802.11 networks, so two additional mechanisms were added to 802.11i:

– TKIP (Temporal Key Integrity Protocol), the successor of the WEP;

– CCMP (Counter-Mode/CBC-MAC), which uses the AES encryption algorithm in CCM mode and a signature MIC.

7.5.3.1. *The TKIP protocol*

The TKIP protocol implements the RC4 encryption algorithm and adds to each SDU (Service Data Units) MAC a signature of 64 bits named MIC (Message Integrity Code). The RC4 key (128 bits) is calculated from a 48 bit (transmitted bit sequence) counter transmitted clearly in each frame and a TK (Temporal Key).

The TKIP frame is detailed in Figure 7.8.

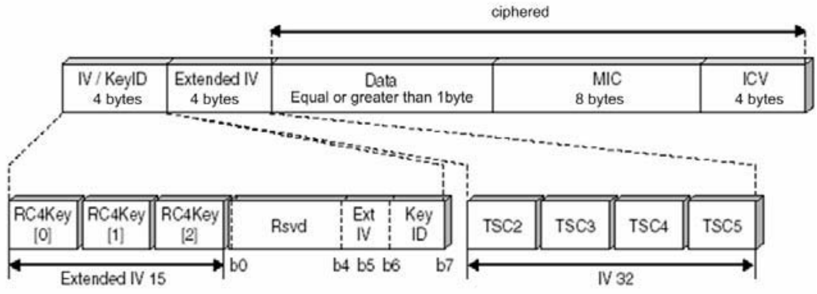


Figure 7.8. *The TKIP frame*

In the TKIP frame, the TSC (Transmitted Counter Sequence) field carries the IV32 and IV16 values for a total of 48 bits. The Rsvd field is always zero and Ext.-IV is always 1. The key ID is equal to 00.

a) TKIP ciphering

TKIP is a ciphering protocol intended to improve WEP. It generates dynamic WEP keys via periodic 802.1x re-authentications.

TKIP calculates the MIC on the source address (SA), the destination address (DA), the priority and the data, then it adds the MSDU (MAC Service Date Unit).

The receiver checks the MIC after deciphering, then reassembles the MPDU (MAC Protocol Data Unit) in MSDU. MSDUs having an invalid MIC are rejected.

Due to the fact that an attacker can compromise the MIC by observing the messages, TKIP implements counter-measures intended to limit updates to the keys. TKIP divides the MSDU in several MPDUs and assigns a TSC (TKIP Sequence Counter) to each PDU that it generates. This value is communicated to the receiver, which rejects the received MPDU in wrong order.

TKIP uses a mixing cryptographic function to calculate the WEP seed, formed by an IV extended to 128 bits and a RC4 key. This helps to cipher the PDU with the WEP. The receiver recovers the TSC from a MPDU and then uses the mixing function to re-compute the WEP seed and decipher the MPDU. The aim of they key mixing function is to avoid attacks due to the use of weak keys. TTAK (TKIP mixed Transmit Address and Key) is an intermediate key produced at the end of phase 1 of the TKIP mixing function.

7.5.3.2. *The CCMP protocol*

The CCMP protocol is founded on the AES (Advanced Encryption Algorithm). It uses the CCM operation mode, which combines the assets of the CTR (Counter Mode) mode for confidentiality and CBC-MAC (Cipher Block Chaining-Message Authentication Code) for authentication and integrity. CCM ensures the integrity of the data fields of the MSDU and also certain selected parts of the MAC header. In CCMP, all the AES treatments use a key and a block size of 128 bits.

CCM uses the same temporary key for CTR and CBC-MAC. Normally, the use of the same key for several functions introduces a security flaw. It is not the case here because the IV is different for the CTR and CBC-MAC modes. Moreover, all the intermediate values in the calculation of the CBC-MAC are random, where the collision probability is very weak. Despite everything, if there is a collision, only the ciphered MIC is affected, and no information can be deduced, not even the occurrence of the collision.

The CCM protocol is a generic mode that can be used with any encryption algorithm oriented to blocks. It employs two parameters, M and L:

– M = 8 indicates that the MIC is coded with 8 bytes.

– L = 2 indicates that the length field is 2 bytes, which is sufficient to preserve the longest possible 802.11 frame.

In addition to one fresh temporary key (TK) per session, CCM requires a random value (Nonce), which is unique for each frame. For this purpose, CCMP uses a number of packets (PN) coded on 48 bits.

CCMP ciphers the MPDU payload and encapsulates the resulting ciphered text by incrementing the PN packet number in order to obtain a fresh PN for each MPDU. The PN should not be repeated for the same temporary TK key.

The fields in the MAC header are used to build the AAD (Additional Authentication Data). CCM protects the integrity of these fields; while masking them to 0, some of them become silent. The Nonce is built from the PN, the A2 (MPDU 2 address) and the MPDU priority. It later encodes the new PN and ID key in the CCMP header of 8 bytes.

The author of the CCM treatment uses the temporary key TK, the AAD, the Nonce and the MPDU data to form the ciphered text and the MIC. The ciphered MPDU is obtained by combining the origin MAC header, the CCMP header, the ciphered data and the MIC.

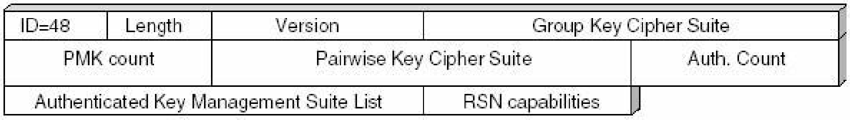
The ciphering parameters are deduced from a 48 bit (Packet Number) counter which is clearly transmitted in each frame and also from a TK key.

7.5.3.3. *The WPA (Wi-Fi Protected Access) solution*

WPA is a subset of the 802.11i standard which regroups 802.1x and TKIP with the purpose of mitigating the WEP security deficiency. The WPA standard is transitory and is already in the course of being replaced by a new 802.11i version called WPA2, which uses the AES ciphering mechanism. All the elements of these two protocols having been presented in the preceding sections, we examine here only the RSN IE frame of WPA and WPA2.

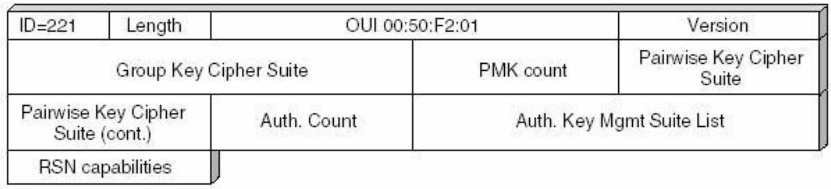
WPA and WPA2 present significant differences. In particular, the WPA default protocol is TKIP and for WPA2 is CCMP.

The RSN IE frame of WPA2 is illustrated in the Figure 7.9 and that of WPA in Figure 7.10.



ID=48	Length	Version	Group Key Cipher Suite
PMK count	Pairwise Key Cipher Suite	Auth. Count	
Authenticated Key Management Suite List	RSN capabilities		

Figure 7.9. *802.11i RSN IE frame*



<table>
<tr><td>ID=221</td><td>Length</td><td colspan="2">OUI 00:50:F2:01</td><td>Version</td></tr>
<tr><td colspan="3">Group Key Cipher Suite</td><td>PMK count</td><td>Pairwise Key Cipher Suite</td></tr>
<tr><td colspan="2">Pairwise Key Cipher Suite (cont.)</td><td>Auth. Count</td><td colspan="2">Auth. Key Mgmt Suite List</td></tr>
<tr><td colspan="2">RSN capabilities</td></tr>
</table>

Figure 7.10. *WPA RSN IE frame*

These two protocols are supposed to guarantee the security of the wireless networks for at least several years. WPA has the advantage of being able to be introduced into the firmware of 802.11 cards built before 2004. This compatibility is explained by the fact that WPA uses the same protocols as WEP. On the other hand, the AES encryption algorithm was not originally implemented in 802.11 cards, and a firmware modification is not possible. It is thus necessary to buy a new compatible WPA2 card to enter this new generation.

Given the incompatibility between the two generations, we can hope that the new generation card will integrate both encryption algorithms. Such cards would be WPA and WPA2 compatible, and they could be used in TKIP with necessary security. It would be enough to lower each year the key refresh speed, because when almost all the cards would be compatible with the AES algorithm, passing to AES ensures the security for several additional years.

7.6. Authentication in wireless networks

As we have seen on several occasions, authentication is an essential security function. This is the reason why the WPA and WPA2 protocols start with the client authentication before authorizing this last to cross the access point. The authentication protocol used comes from the IEEE 802.1x standard, which is not expressly reserved for wireless networks but is related to all network categories. We have detailed the broad outlines of this authentication technique but without detailing the low level protocols that are able to transport the authentication information in a secure way. This is the problem we examine in this section.

We start by introducing the PPP protocol and all its derivatives, then we examine the EAP (Extensible Authentication Protocol) extensions, which became the standard for authentication information transport. We also present some protocols that could play an important role in authentication.

7.6.1. *RADIUS (Remote Authentication Dial-In User Server)*

On the other side of the Atlantic, Internet Service Providers (ISPs), frequently use pools of modems installed in urban telephone exchange centers. This infrastructure, allowing cheap accesses, is called POP or Point Of Presence. Rather than duplicate and update the client accounts database in each POP, the ISP deployed a centralized architecture, ensuring a remote management of their clients and achieving the three following functional levels:

– the user is provided with a login and a password, i.e. an 802.1x supplicant in our case;

– the NAS (Network Server Access) controlling the whole pool of modems and providing the interface to the authentication server; this is analog to an 802.1x authenticator;

– the RADIUS server acting as an 802.1x authentication server. This last system provides the interface to the database managing the user account. The authentication dialogue is usually based on PAP or CHAP protocols and it is relayed by the NAS between the users and the authentication server.

The NAS is a bridge between the PAP or CHAP protocols transported by PPP and the RADIUS server. In the case of PAP, it transmits to the RADIUS server the user’s identity and his password in order to be verified. The RADIUS server indicates to the NAS if the operation succeeds or fails. The NAS also measures the time the client uses the service and transmits a invoicing request when he leaves the POP.

7.6.2. *EAP authentication procedures*

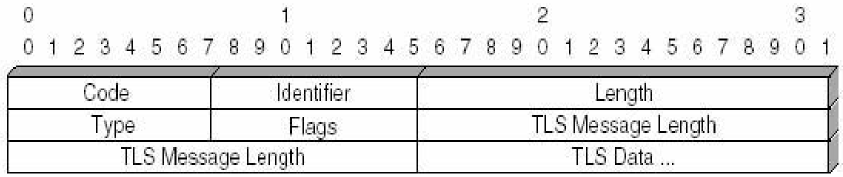
As explained previously, the EAP became the standard tunnel for authentication. We set up this tunnel to perform the authentication procedure itself. A vast range of authentication mechanisms are possible. LEAP (Extensible Lightweight Authentication Protocol) is the solution chosen by Cisco Systems for its first wireless networking equipment. FAST-EAP should be one of the standards proposed by Cisco in the future, LEAP showing some weaknesses in particular cases, such as the dictionary attack, because the passwords are not sophisticated enough. EAP/SIM and EAP/TLS are the two big standards at present. They correspond to the choices selected by the mobile network operators and by many software publishers, like Microsoft. Two additional solutions, PEAP (Protected EAP) and EAP by smart card, are pushed by Microsoft (in the former case) and by the smart card equipment suppliers (in the latter case).

7.6.2.1. *EAP-TLS (Transport Layer Security)*

EAP-TLS authentication became the best recognized authentication technique and it is considered to be one of the most solid thanks to the mutual authentication it executes. In fact, TLS is only one extension to the SSLv3 procedure, which is frequently used for application level authentications between the client and the Web server. This EAP-TLS solution was chosen by many companies. Microsoft, for example, has included a standard version in its operating system since Windows 2000. The EAP-TLS procedure is not strictly used in a wireless network environment. It is however within this framework that it reveals all its power.

Defined by the RFC 2716 of October 1999, EAP-TLS is based on a PKI infrastructure. The RADIUS server and the client of the network have certificates issued by a common Certification Authority.

The EAP-TLS packet format is illustrated in Figure 7.11.



```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
```

<table>
<tr><td>Code</td><td>Identifier</td><td colspan="2">Length</td></tr>
<tr><td>Type</td><td>Flags</td><td colspan="2">TLS Message Length</td></tr>
<tr><td colspan="2">TLS Message Length</td><td colspan="2">TLS Data ...</td></tr>
</table>

Figure 7.11. *EAP/TLS packet*

EAP-TLS uses the TLS handshake to allow the client and the server to exchange their digital certificate, which is the basis of authentication. The server presents a certificate to the client that the client validates. Optionally, the client presents his certificate to the server. The certificate can be protected on the client's side by a password, a PIN code or a smart card.

An EAP-TLS conversation between a client requiring access to the network and the access point proceeds in the following way:

– The access point sends an EAP-REQUEST/IDENTITY packet.

– The client answers using an EAP-RESPONSE/IDENTITY packet, containing the identity of the user.

– The server sends an EAP-TLS/START packet.

– The answer of the client is an EAP-RESPONSE packet containing a TLS CLIENT_HELLO HANDSHAKE message. The CLIENT_HELLO message contains the TLS version of the client, a random number and a list of encryption algorithms supported by the client.

– The server sends an EAP-REQUEST packet where data contains a SERVER_HELLO HANDSHAKE message. This message specifies the TLS version of the server, another random number, a session identifier and a CIPHERSUITE message corresponding to the selected encryption algorithm.

– The client answers by an EAP-RESPONSE packet, the data field encapsulates a TLS_CHANGE_CIPHER_SPEC message and a FINISHED_HANDSHAKE message.

Figure 7.12 illustrates the different messages sent in the authentication phase. This case represents a successful authentication between authenticator and the client.

The TLS Master Secret, or the MSK (Master Session Key), is the shared secret between the client and the server, the result of the handshake phase. The following data are derived from the MSK:

– the ciphering client key (MSK (0,31));

– the ciphering server key (MSK (32,63));

– the authentication client key for the MAC calculation client side (MSK (64,95));

– the authentication server key for the MAC calculation server side (MSK (96,127));

– two initialization vectors (IV).

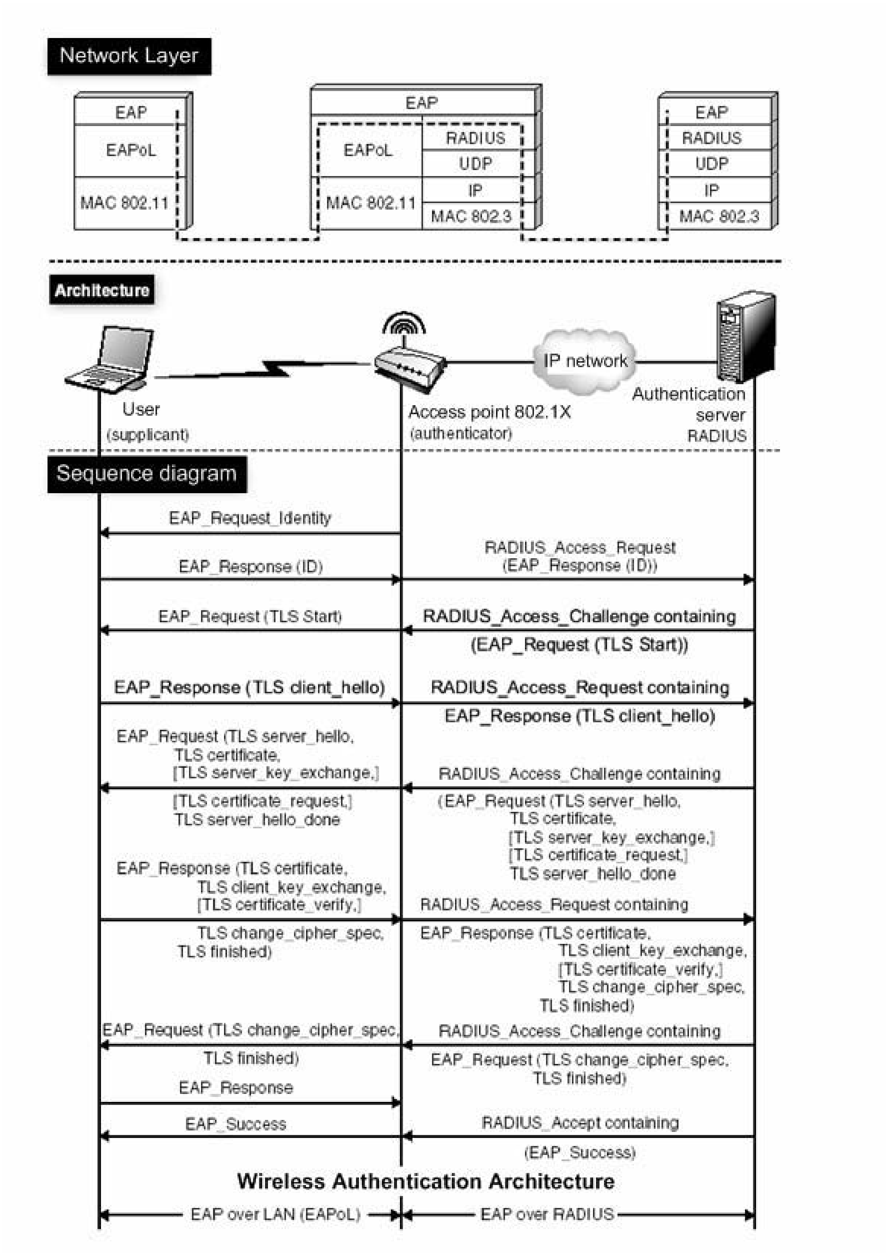

Network Layer
EAP
EAPoL
MAC 802.11
EAP
EAPoL
RADIUS
UDP
IP
MAC 802.11
MAC 802.3
EAP
RADIUS
UDP
IP
MAC 802.3
Architecture
IP network
User
(supplicant)
Access point 802.1X
(authenticator)
Authentication
server
RADIUS
Sequence diagram
EAP_Request_Identity
RADIUS_Access_Request
(EAP_Response (ID))
EAP_Response (ID)
EAP_Request (TLS Start)
RADIUS_Access_Challenge containing
(EAP_Request (TLS Start))
EAP_Response (TLS client_hello)
RADIUS_Access_Request containing
EAP_Response (TLS client_hello)
EAP_Request (TLS server_hello,
TLS certificate,
[TLS server_key_exchange,]
[TLS certificate_request,]
TLS server_hello_done
RADIUS_Access_Challenge containing
(EAP_Request (TLS server_hello,
TLS certificate,
[TLS server_key_exchange,]
[TLS certificate_request,]
TLS server_hello_done
EAP_Response (TLS certificate,
TLS client_key_exchange,
[TLS certificate_verify,]
TLS change_cipher_spec,
TLS finished)
RADIUS_Access_Request containing
EAP_Response (TLS certificate,
TLS client_key_exchange,
[TLS certificate_verify,]
TLS change_cipher_spec,
TLS finished)
EAP_Request (TLS change_cipher_spec,
TLS finished)
RADIUS_Access_Challenge containing
EAP_Request (TLS change_cipher_spec,
TLS finished)
EAP_Response
EAP_Success
RADIUS_Accept containing
(EAP_Success)
Wireless Authentication Architecture
EAP over LAN (EAPoL)
EAP over RADIUS

Figure 7.12. *EAP-TLS authentication*

The ciphering client key, also called the PMK, is transmitted to the access point via the MS-MPPE-RECV-KEY RADIUS attribute. The WEP key will be ciphered with this key then signed before being given to the client.

The authentication server can check if the client's certificate is revoked. Also, the client can check the validity of the certificate of the server. However, this checking can be done only once in the connection finishing phase. Indeed, a client initiating a level connection conversation does not have connectivity.

The TLS message transportation primarily poses a segmentation problem. The size of a TLS register is more than 16,384 bytes, but the RADIUS protocol limits its payload to 4,096 bytes. In addition, the 802.11 frames size is limited to 2,312 bytes. EAP-TLS must thus support a register segmentation mechanism. Contrary to the common TLS usage, implementing a simple server authentication, EAP-TLS uses a mutual authentication between the RADIUS server and the 802.1x client.

The use of a private key by the 802.1x client raises the critical problem of security required by its storage as well as the implementation of such a component. In the usual data-processing platforms, this security is ensured by passwords that make it possible to decipher and use the private key. The smart card constitutes a protected alternative with this method.

The use of authentication with digital certificates obliges us to have a suitable PKI infrastructure. If such an infrastructure is not deployed, the client certificates involve an important management surplus. However, EAP/TLS is natively supported on Windows platforms, where the client certificate can be stored in a smart card.

7.7. Layer 3 security mechanisms

During the previous sections, we examined the different mechanisms implemented in wireless networks, and more particularly in Wi-Fi networks, to obtain an acceptable security. We saw that in the first generation of these mechanisms, with WEP, there was little security and the second and third generations, being launched in the market, were likely to satisfy the security needs of companies. While waiting for the arrival of these new generations, many solutions were developed or taken from wired networks to mitigate the weaknesses of the first generation. These solutions can be added again to those coming from the IEEE standards to ensure a still better level of security.

Most of these mechanisms come from the treatment of IP packets and are thus at the packet level, i.e. on level 3 of the reference architecture. As we will see, several of the described mechanisms are only derived from level 3 and are in fact located either near the lower level; the frame level, or near the higher level; or at the message and application level. Some of these mechanisms were standardized by the IETF and others by the ISO (International Standardization Organization).

The distribution of secret keys in the machines which want to communicate by the PKI infrastructure mechanisms makes it possible to authenticate and cipher their communications in WPA or WPA2 systems. The passage of secret keys is performed thanks to asymmetric encryption algorithms.

The VPN ensures very good communications security that crosses not very secure networks. Wi-Fi networks that are not secure networks can greatly benefit from this solution.

The IPsec protocol is one of the most used protocols in order to guarantee data confidentiality, but its advantages are greater. IPsec is heavily used in the VPN. The IPsec competitor, SSL, intervenes on a higher level than level 3 because it was conceived for exchanges between navigators and Web services.

Firewall technologies and filters are also well adapted to wireless networks.

7.7.1. *PKI (Public Key Infrastructure)*

PKI infrastructures are at the basis of the distribution of secret keys that are now mandatory in order to seriously handle authentication and encryption. They are mandatory in EAP-TLS and PEAP authentications.

An important choice for the deployment of a PKI is the format of the digital certificate. The most commonly accepted format is the UIT-T X.509 standard. The public key is associated with a certificate containing a name, an address and other information describing the person transporting the secret key. All the certificates are signed by the third party which registers the public key of the members of the community.

To become a registered member, a person has to satisfy two conditions:

– provide the registration directory with a public key and some authentication information so that other persons can verify the signature;

– obtain the public key of the repository service so that a registered member can verify the signature of the other persons.

A signed certificate cannot be changed. The authenticity depends on the channel through which it was received.

A Certificate Authority (CA) transmits, manages and removes certificates. The public key of the CA certificate has to be trusted by all the final users. The certificates sent to the final user are called user certificates, and those sent for validation between CAs are called CA certificates.

A CA for the whole world is not appropriate. A distributed PKI architecture where CAs are authorized to certify other CAs is necessary. A CA can delegate its authority to a subordinate authority by emitting a CA certificate creating a certificate hierarchy. The ordered sequence of certificates of the last branch from the root is called path or chain of certification.

Each certificate contains the name of the issuing process, that is, the name of the certificate directly above it in the chain. In general, it is possible to have an arbitrary number of CAs on the path between two users. To obtain the public key of its corresponding node, a user must verify the certificate of each CA. This process is called validation of the certification path.

When several CAs are used, the manner in which the CAs are organized is very important for building the PKI architecture. Some PKIs are using a hierarchical model, called general hierarchy, in which the CAs certify only their son and the CA root in all the certification paths.

In top-down architectures, all users must use the highest CA as a root. This necessitates that all users obtain a copy of the public key of the highest CA before using the PKI. All the users must trust the CA root, and this is impracticable for a global PKI.

Cross-certification can help to reduce the length of the path, but at the risk of complicating the discovery of this path in the validation process.

For an external communication in a company, the interoperability of the PKI is essential. The main normalizing efforts come from RSA laboratories with PKCS (Public Key Cryptography Standards). Currently, the PKCS are real standards and are unanimously adopted, mainly for cryptographic processes and key exchanges. In parallel, the IETF produces more general standards, like the RFC PKIX (Public Key Infrastructure X.509). Some aspects stay insufficiently normalized, like the policies and practices for certification or parameters of the certificates.

7.7.2. *Level 3 VPN*

VPNs are the equivalent of a private network interconnecting the different geographically distributed sites of the same company. In other terms, the different sites of a company can be interconnected through a VPN as if the network belonged to the company. It is impossible for a client coming from another network to access the company. On the contrary, a client of the company cannot go out of this network without a very specific authorization. In summary, a VPN is a network that seems private but which is only a shared and protected telecommunication network such that the different companies are independent of each other and have the impression of possessing the network for themselves.

Operators are very interested in VPNs. They can distribute their resources between the different clients so that these clients have the impression of being alone, obtaining a reasonable response time, and having the possibility of using a strong multiplexing of network resources. This makes it possible to increase the benefits of operators since the resources are sold several times to different companies.

VPNs are private networks in which a resource allocation is realized through demand. The entrances of the VPNs are situated at different levels of the architecture, but generally at the IP level. The packet level (layer 3) being the IP level, level 3 VPNs are called VPN-IP. This VPN generation came on the market at the beginning of the new millennium. They make it possible to access to all the properties that we can find in Intranet and Extranet networks, in particular the information system of a distributed company. The IP solution allows at the same time the integration of fixed and mobile stations.

An IP-VPN is illustrated in Figure 7.13. Companies A, B, and C have IP-VPNs. Their access points are IP routers allowing the arrival and departure of packets from and to the different sites of the company.

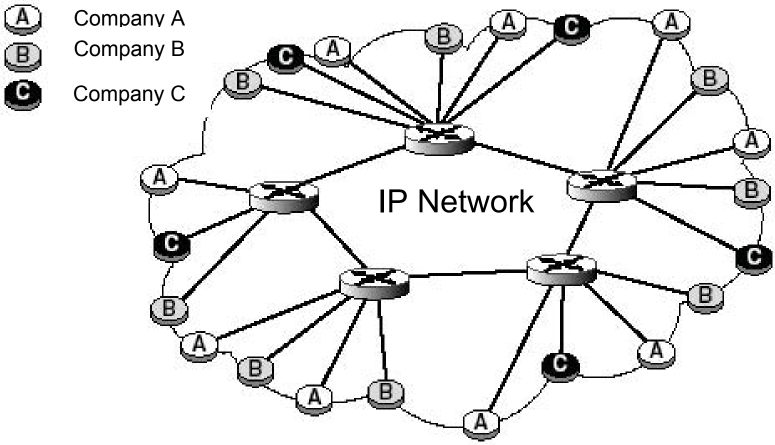


Figure 7.13. *IP-VPN*

The client of a VPN uses the IP network to go from one access point to another access point belonging to the same VPN. Quality of Service and security are taken into account by the users. As security is a very important item in these networks, the first generation of IP-VPN uses the IPsec protocol to carry the packets. This protocol makes it possible to create tunnels that can be encrypted by algorithms. The VPN access points communicate between themselves through the encrypted tunnels.

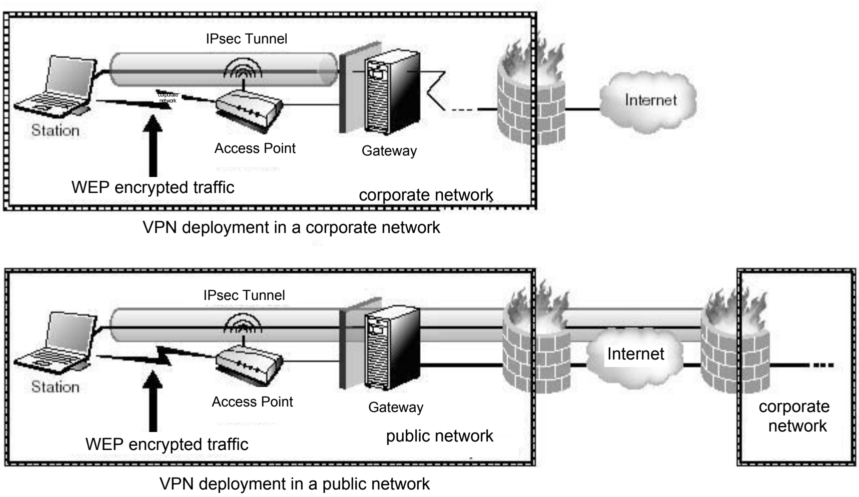
VPNs use encryption technologies to protect the IP packets going from one LAN of the company to another.

Today the majority of VPNs use IPsec or SSL protocols, normalized by the IETF. IPsec provides the following protections: confidentiality, integrity, non-repudiation, authentication and protection against traffic analysis. The header ESP (Encapsulating Security Protocol), when used, means that the data pass in transit inside an encrypted IPsec tunnel. Therefore, confidentiality is guaranteed. In the same way, the use of the AH (Authentication Header) indicates that the data are protected against all malicious modifications (data protection with integrity but not confidentiality).

The IKE (Internet Key Exchange) protocol allows the exchange of secret keys and of secure parameters before communication without user intervention. Applications and protocols operating above are protected by the IPsec protocol.

Figure 7.14 illustrates an example of a wireless network with VPNs. Using wireless terminals, users can connect themselves in a secure way to the company going through the VPN gateway. Above the WEP, wireless clients establish IPsec connections with the VPN gateway.

The use of a VPN is slightly different in a public network. The IPsec tunnel is remotely connected to the VPN gateway of the company.



VPN deployment in a corporate network

VPN deployment in a public network

Figure 7.14. *The use of a VPN in wireless networks*

The VPN gateway can use shared or digital certificate cryptographic keys for the authentication of a wireless client. Companies providing a PKI with smart cards for in order to memorize the certificate of the client can use this smart card in VPN solutions.

7.7.3. *IPsec*

The TCP/IP world makes it possible to connect millions of users requiring their communications to stay secret. Moreover, Internet has massively adopted e-commerce, in which confidentiality is necessary to take charge of the transmission of a very large numbers of banking cards, for example.

The idea developed in IP security working groups on e-commerce consists of the definition of an environment containing a set of security mechanisms. As not all communications have the same characteristics, their security does require the same algorithms. Appropriate security mechanisms have to be chosen through a security association. Each communication defines its own security association. The main items in a security association are as follows:

– authentication and cryptographic algorithms to be used;

– global or specific keys to be used;

– other parameters of the algorithm, such as synchronizing data or initializing values;

– duration of the validity of keys or associations;

– sensitivity of the provided security (secret, top secret, etc.).

The IPsec solution introduces some security mechanisms at the IP level, so that there is independence from the transport protocol. The use of IPsec properties is optional in IPv4 and mandatory in IPv6.

A security base called SAD (Security Association Database) groups together all the characteristics of the different associations by the intermediate of the communication parameters. Their use is defined in another database, the SPD (Security Policy Database). One input of the SPD database groups together all the IP addresses of the source and the destination, plus the identity of the user, the required level of security, the identification of the protocols that are used, etc.

The format of IPsec packets is illustrated in Figure 7.15. The highest part of the figure corresponds to the format of the IP packet in which a TCP fragment is encapsulated. The middle part of the figure illustrates the IPsec packet itself. The IPsec header is between the IP header and the TCP header. The lowest part of the figure shows the format of a packet going through an IP tunnel. The lowest part corresponds to an encapsulated IPsec packet so that the interior IP packet is very well protected.

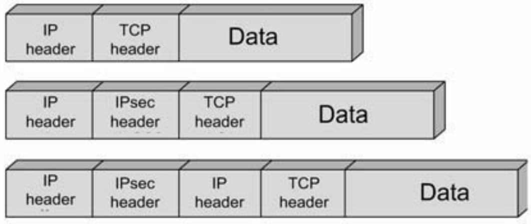


Figure 7.15. *Format of IPsec packets*

In an IPsec tunnel, all the IP packets from the same flow are encrypted. Thus, it is impossible to see either the IP addresses or the values of the supervising field of the encapsulated IP packet.

7.8. Bibliography

[1] K. AL AGHA, G. PUJOLLE, G. VIVIER, *Réseaux de mobiles et réseaux sans fil*, Eyrolles, 2001.

[2] K. CURRAN, *Wi-Fi Security*, BookSurge Publishing, 2006.

[3] H. DAVIS, R. MANSFIELD, *The Wi-Fi Experience: Everyone's Guide to 802.11b Wireless Networking*, Que, 2001.

[4] A. DORMAN, *The Essential Guide to Wireless Communications Applications*, Prentice Hall, 2002.

[5] J. EDNEY, W.A. ARBAUGH, *Real 802.11 Security: Wi-Fi Protected Access and 802.11*, Addison Wesley, 2003.

[6] R. FLICKENGER, *Building Wireless Community Networks*, O'Reilly, 2001.

[7] M.S. GAST, *802.11 Wireless Networks: The Definitive Guide*, O'Reilly, 2002.

[8] T. GEE, *Set Up Your WiFi Network*, Micro Application, 2003.

[9] J. LA ROCCA, *802.11 Demystified: Wi-Fi Made Easy*, McGraw-Hill, 2002.

[10] M. MAXIM, D. POLLINO, *Wireless Security*, Osborne McGraw-Hill, 2002.

[11] P. MÜHLETHALER, *802.11 et les réseaux sans fil*, Eyrolles, 2002.

[12] R.K. NICHOLS, P.C. LEKKAS, *Wireless Security: Models, Threats and Solutions*, McGraw-Hill, 2001.

[13] K. PAHLAVAN, P. KRISHNAMURTHY, *Principles of Wireless Networks: A Unified Approach*, Prentice Hall, 2001.

[14] J. REYNOLDS, *Going Wi-Fi: A Practical Guide to Planning and Building 802.11 Networks*, CMP Books, 2003.

[15] K. ROEDER, JR. F.D. OHRTMAN, *Wi-Fi Handbook: Building 802.11b Wireless Networks*, McGraw-Hill, 2003.

[16] C.W. SAYRE, *Complete Wireless Design*, McGraw-Hill, 2001.

[17] J.R. VACCA, *Guide to Wireless Network Security*, Springer, 2006.

[18] A. VLADIMIROV, K.V. GAVRILENKO, A.A. MIKHAILOVSKY, *Wi-Foo: The Secrets of Wireless Hacking*, Addison-Wesley, 2004.

Chapter 8

WiMAX Security

8.1. Introduction

The IEEE 802.16 standard deals with last mile network technologies. It is intended for the building of Wireless Metropolitan Area Networks (WMANs) supporting indoor or outdoor features. These are aimed at fixed, nomadic or large mobility uses (a car moving at normal speed for example). It is a flexible framework, compatible with a large range of frequencies such as 10-66 GHz or 2-11 GHz.

8.1.1. *A brief history*

The former version approved in 2001 and called IEEE 802.16-2001 [IEE 01], uses the 10-66 frequency band GHz (see Table 8.1). With such frequencies, radio devices are in the line of sight (LOS) and the link no longer works when an obstacle such as a tree or a building comes in between its two ends. For lowest frequencies, between 2 and 11 GHz, this constraint is no longer essential. In this special case, we talk about non-line of sight (NLOS). The second version of the standard published in 2004 and called IEEE 802.16-2004 [IEE 04] includes the two ranges of frequencies already quoted, and enables deployments like LOS and NLOS. The new version finalized in February 2006, IEEE-802.16e-2006 [IEE 06b], applies more particularly to the 5-6 GHz frequency band. The information coding takes into account the specific problems resulting from the speed of the users, such as the *Doppler effect*.

Chapter written by Pascal URIEN and translated by Léa URIEN.

To conclude, MIMO (multiple input, multiple output) technology uses many frequency carriers for transmission and reception, and makes the integration and deployment of WiMAX easier.

Standards	IEEE 802.16-2001	IEEE 802.16-2004	IEEE 802.16e
	December 2001	October 2004	February 2006
Frequency Band	10-66 GHz	2-11 GHz	< 6 GHz
Data Throughput	32-134 Mbits in 28 MHz channels	Up to 75 Mbits in 20 MHz channels	Up to 15 Mbits in 5 MHz channels
Modulation Technique	QPSK, 16QAM, 64QAM	OFDM 256 sub-carriers OFDMA 2,048 sub-carriers	S-OFDMA
Mobility	Fixed	Fixed, Nomad	High Mobility
Channel Bandwidth	20, 25 and 28 MHz	Variable 1.5 to 20 MHz	Similar to IEEE 802.16-2004
Cell Radius	2-5 km	7-10 km Maximum 50 km	2-5 km

Table 8.1. *Summary of the IEEE 802.16 standards*

8.1.2. *Some markets*

In a nutshell, WiMAX networks offer five types of service. The first one is the providing of wireless telephony services such as T1 in Europe (2,048 Mb/s) or E1 (2,000 Mb/s) in the USA. It is an alternative opportunity to the operators' offers dealing with cabled infrastructures. On-demand broadband enables a company to establish high-performance connections between its agencies, in order to organize videoconferences for example. This technology also provides, to areas with poor Internet accesses, high speed internet access, which is similar to DSL (*Digital Subscriber Link*) but based on radio waves. In the same way, geographical areas with a high cabling cost may also benefit from this technique. In this case it is called a *Local Radio Loop* and it delivers voice and data services. Finally, the WiMAX network is a complement to Wi-Fi hotspots. It guarantees the continuity of IP connectivity for a nomad user or driver. A subscriber may be ruled by a single *Wireless Internet Service Provider* (WISP) or can benefit from agreements between different WISPs, in order to retain these services in a transparent way. This mechanism is called roaming.

8.1.3. *Topology*

The WiMAX architecture (see Figure 8.1) is composed of base stations (BSs), including several bidirectional radio antennas, covering geographical sectors (cone-shaped) and establishing PMP (point to multi-point) links. In a given sector, the downlink frames (transmission of data to subscribers) and the upstream frames (reception of data from subscribers) are managed by a single BS.

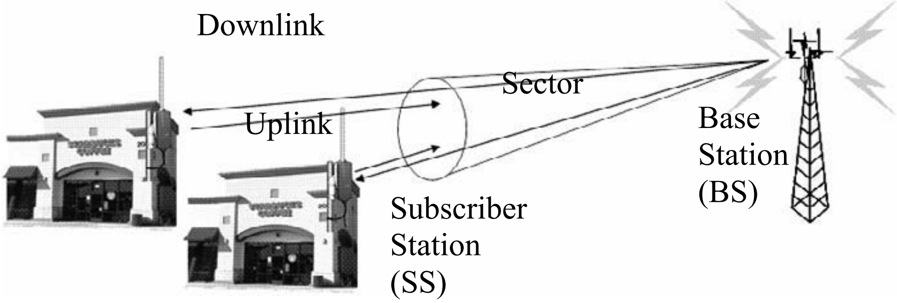


Figure 8.1. *PMP architecture*

The BS periodically transmits management frames, indicating the information structure of the link:

– downlink organization is described by the downlink map message (DL-MAP);

– upstream organization is described by the uplink map message (UL-MAP).

To be more precise, a radio link is organized in bursts. Each of them is identified by a Downlink Interval Usage Code (DIUC) or an Uplink Interval Usage Code (UIUC), which define modulation processes and other coding parameters. This scheme makes it possible to obtain data rates adapted to the observed signal-to-noise ratio between a subscriber and a BS. A communication channel comprises one or several bursts organized in several logic subsets.

The subscriber station (SS) in 802.16 or the mobile station (MS) in 802.16e analyzes the incoming frames and uses the upstream channels for different classes of services such as system administration (connection requests, Quality of Service (QoS) allocation, etc.) or data transmission (according to a *best effort* mode, for example). The management of access collisions for upstream channels is realized using several classes of algorithms.

A second alternative supports MESH networks (see Figure 8.2). A BS linked to the backbone network is called the mesh BS. The other components of the infrastructure are called the mesh SS.

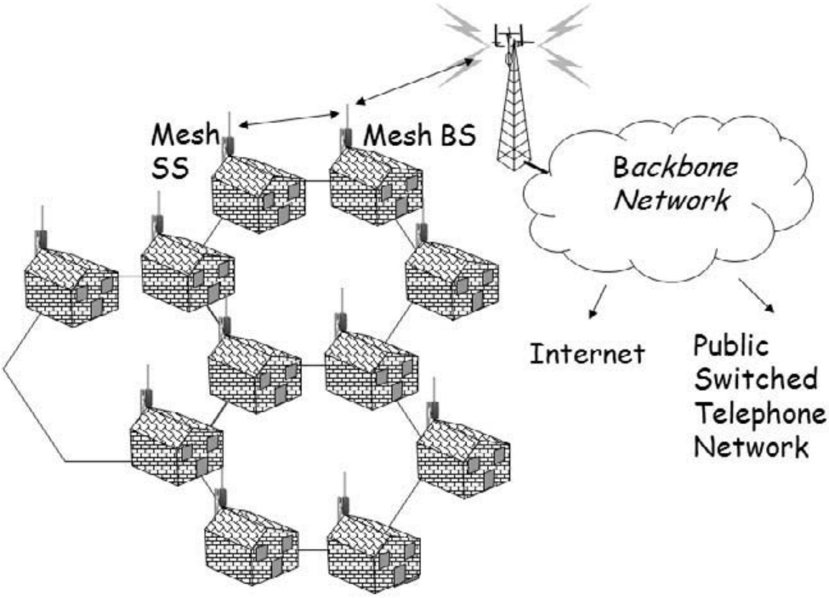


Figure 8.2. *MESH architecture*

8.1.4. *Security evolution in WiMAX standards*

WiMAX security has evolved with the targeted applications.

Even though the network architecture is different, WiMAX concepts result from the IEEE 802.14 project (cable-TV access method and physical layer specification), started in 1996, but today given up. The IEEE 802.14 standard suggested defining a MAC protocol based on the ATM infrastructure and dedicated to TV broadcasting via cables. On the one hand, the headend is connected to an operator network. On the other hand, it is connected to a group of users, who are kitted out with *cable modems* (CMs). The security of exchanges between subscribers and the headend is based on several parameters: a cookie, a cryptographic key computed via a Diffie Hellman procedure, and two random numbers generated by each entity. The MAC frames are ciphered by the DES algorithm, with a key size of 40 or 56 bits.

Across the Atlantic, the cable modem is a connection technology supported by many ISPs. There are about 10 manufacturers of such devices (you can find them on the website http://www.cable-modems.org). The dominant standard is the DOCSIS (*Data-Over-Cable Service Interface Specifications*) [DOC 05]. It is published by the *CableLabs* consortium (http://www.cablelabs.com) founded in 1998 by the US operators of cabled television. This association is represented at the IETF by the working group *IPCDN* (IP over cable data network), which defines the management information base dedicated to cable modems. The DOCSIS architecture is similar to the DSLAM infrastructure. A distribution hub (sometimes called a headend) simultaneously accesses the Public Switched Telephone Network (PSTN) and the operator's networks. This entity is also linked with many users who are kitted out with cable modems. The BPKM (*Baseline Privacy Key Management*) protocol offers two types of service: user authentication and data privacy with distribution hubs. The security is based on a public key infrastructure (PKI). Each cable modem holds a X.509 certificate and a RSA private key. The DOCSIS root CA (Certification Authority) allocates certificates to the manufacturers, who in turn deliver a certificate of conformity to their equipment. The distribution hub authenticates a cable modem thanks to its certificate associated with a private key. The frames are encrypted by a conventional DES algorithm.

The security mechanisms introduced by the IEEE 802.16-2001 standards [IEE 01] are very close to the DOCSIS standard, first from a functional point of view and also at a binary encoding level. In this standard working in LOS mode, the radio beams are caught thanks to high points such as skyscrapers or pylons. As a consequence, data security is light: a simple encryption with the DES algorithm and a small cryptographic key (56 bits). Each SS is equipped with a certificate that proves its conformity. It could be a modem in which the cable would have been replaced by radio links.

With the 2004 standard, NLOS deployments become possible due to the numerous reflections of the radio signal. This also increases hacking risks. As a result, data protection is stronger thanks to the AES encryption with a 128 bit key for example. The integrity of the frame contents is also guaranteed. However, the subscriber's authentication is based on a certificate and it is not mutual. The network authenticates its subscriber but the contrary is not true.

The large mobility introduced by the 802.16e standard involves organizing an architecture similar to the one previously defined for Wi-Fi (IEEE 802.1x) and including one of several authentication servers. In the same way, mutual authentication becomes necessary because the low price of BSs allow hackers to easily install rogue BSs.

8.2. WiMAX low layers

Conforming to the IEEE 802 LAN models, the logical architecture of a node is divided in two subsets (see Figure 8.3): the MAC (Medium Access Control) layer and the PHY (physical) layer. The originality of these blocks is to incorporate sublayers which realize the operations necessary to the services abstractly defined, that is, independently of specifications of radio technologies.

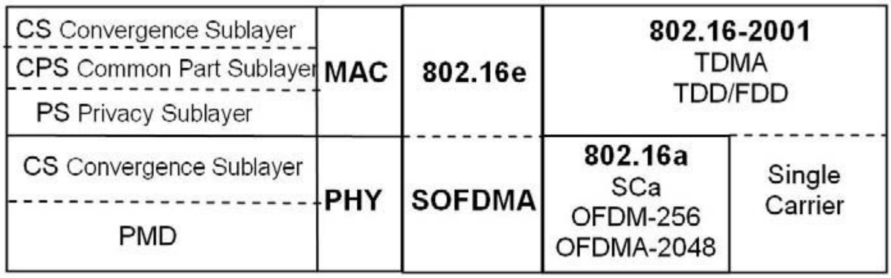


Figure 8.3. *The WiMAX layers*

8.2.1. *MAC layers*

The MAC layer is divided into three components: a convergence sub-layer, a common part sub-layer and a privacy sub-layer:

– The convergence sub-layer (CS) realizes the interface between an external network (ATM, Ethernet) and the Service Data Units (MAC-SDU) exchanged with the common part sub-layer (MAC-CPS). It rules a classification mechanism (see Figure 8.4) in charge of the QoS, in associating with each connection identifiers (CID, a number of 16 bits) and a data stream exchanged with an external network (and identified by a service flow identifier (SFID), a number of 32 bits).

– The common part sub-layer (CPS) is linked to the physical resources. It supports radio connections, enforces the QoS mechanisms and rules the access (transmission/reception) at physical level. It also exchanges SDU with others classes of CS.

– The privacy sub-layer (PS) is in charge of the authentication mechanisms and key exchanges. It also rules the encryption and the integrity of the frames.

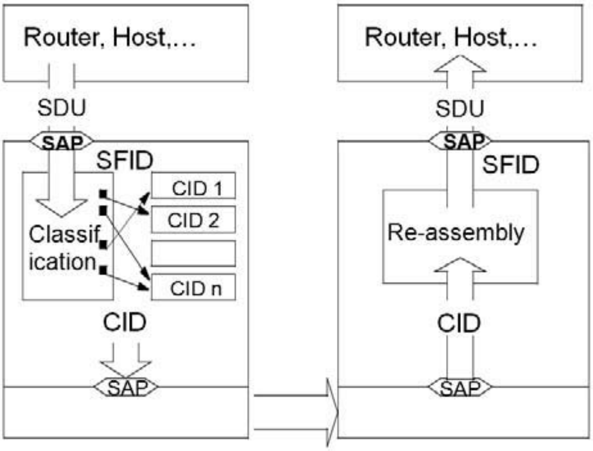


Figure 8.4. *The classification concept*

8.2.2. *The physical layer*

The physical layer (PHY) is divided into two parts: a convergence sub-layer (CS) and a physical medium dependent (PMD) sub-layer. However, when the PMD sub-layer realizes every necessary service to the MAC-CSP entity, the CS is empty. The 802.16 standards essentially introduce four types of physical layer:

– The WirelessMAN-SC PHY layer. This is a technology using a single carrier.

– The WirelessMAN-OFDM PHY layer deals with an Orthogonal Frequency Division Multiplexing scheme that includes 256 carriers. The access to the SS is controlled by a TDMA (Time Division Multiple Access) algorithm.

– The WirelessMAN-OFDMA PHY layer is based on an Orthogonal Frequency Division Multiple Access process. The multiple accesses result from the allocation of a subset of carriers to a single connection (a combination of TDMA and OFDMA processes).

– The WirelessMAN-SOFDMA (scalable OFDMA). This process works with a fast Fourier transform (FFT) whose size ranges from 128 to 2,048 samples, the interval between subcarriers having a constant value of 3.94 KHz.

8.2.3. *Connections and OSI interfaces*

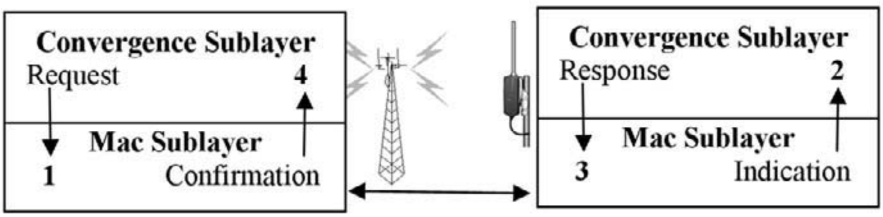


Figure 8.5. *OSI connection interface*

MAC_CREATE_CONNECTION. request	.confirmation
(scheduling service type, convergence sub-layer, service flow parameters, payload header suppression indicator, length indicator, encryption indicator, Packing on/off indicator, Fixed-length or variable-length SDU indicator, SDU length, CRC request, ARQ parameters, sequence number)	(Connection ID, response code, response message, sequence number)

Figure 8.6. *Connection procedure details*

MAC_DATA.request	MAC_DATA.indication
(Connection ID, length, data, discard-eligible flag, encryption flag)	(Connection ID, length, data, reception status)

Figure 8.7. *Procedure for data transfer*

A connection is a unidirectional link between a BS and a fixed (or mobile) station identified by a connection identifier (CID, 16 bits). There are two types of connections: transport and management. A transport connection is associated with a security association (SA) and a flow identifier (SFID). A management connection has no association security or SFID.

In accordance with the OSI model, the MAC interface bears four classes of primitives (see Figure 8.5) for each connection management procedure:

– The creation of connections, MAC_CREATE-CONNECTION.request, indication, response, confirmation. Figure 8.6 specifies the different parameters included in a connection request. We can specifically notice the scheduling service type, a CS, and the service flow parameter. The induced response associates a connection identifier (CID) with the previous attributes.

– The modification of connections, MAC_CHANGE_CONNECTION.request, indication, response, confirmation.

– The transmission/reception of data, MAC_DATA.request, indication, confirmation and response. Data packets are always associated with a CID (see Figure 8.7).

8.2.4. *MAC frame structure*

A MAC frame includes three parts (see Figure 8.8): a header, a payload and a cyclic redundancy check (CRC).

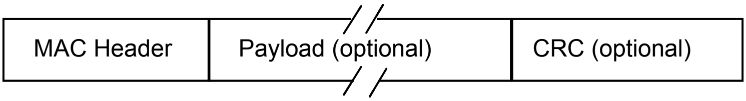


Figure 8.8. *The IEEE 802.16 MAC frame*

A header is a 48 bit structure whose type is generic or bandwidth request. Generic headers are followed by a payload and a CRC, while bandwidth request neither has a payload nor a CRC.

A generic MAC header essentially includes the following pieces of data:

– A connection identifier (CID) of the recipient. According to the CID, the payload contains data or administrative information (see section 8.2.5).

– An encryption control (EC) flag, which points out an encrypted payload.

– An encryption key control (EKC), an index associated with a traffic encryption key (TEK), ranging between 0 and 3.

– The length (coded on 11 bits) of the MAC_PDU, ranging between 0 and 2,047 bytes.

– A type field (6 bits), which points out the presence of optional headers used for many services such as the segmentation of long messages.

Management Message Type	Management Message Payload

Figure 8.9. *Structure of the IEEE 802.16 management frames*

8.2.5. *The management frames*

The management frames (see Figure 8.9) play a fundamental role during the access procedure to the WiMAX network as for the transport of authentication protocols. They are associated with specific CIDs and require a generic MAC header. A management message includes a first byte which defines its role (MngtType), that is, which defines the nature of the required operation, and a payload whose structure depends on the message type (MngtType parameter).

8.2.6. *Connection procedure of a subscriber to the WiMAX network*

A subscriber analyzes the downlink and establishes a primary management connection with the BS. This is used for operations dealing with the authentication and management of cryptographic keys. Upon success of the authentication and registration operations, the secondary management connection is established and enables the SS and BS to create transport connections thanks to *MAC-create_connection* primitives (see section 8.2.3). The basic insertion procedure into a WiMAX network is divided into ten steps described by Figure 8.10:

– *Scanning and synchronization with the downlink channel.* The reception module of the subscriber analyzes the downlink signal and synchronizes with it. This operation is realized by analyzing the features of the downlink channel periodically provided by the BS thanks to management messages DL-MAP (MngtType = 2). The MAC module of the subscriber deduces, thanks to DL-MAP, the number of bursts of the downlink channel. Then, it obtains the structure of the channels inserted into the DCD message (downlink channel descriptor, MngtType= 1).

– *Acquisition of the uplink channel parameters*. The subscriber deduces from the UL-MAP (MngtType= 3) and UCD (uplink channel descriptor, MntgType= 0) messages the structure of the transmission channels.

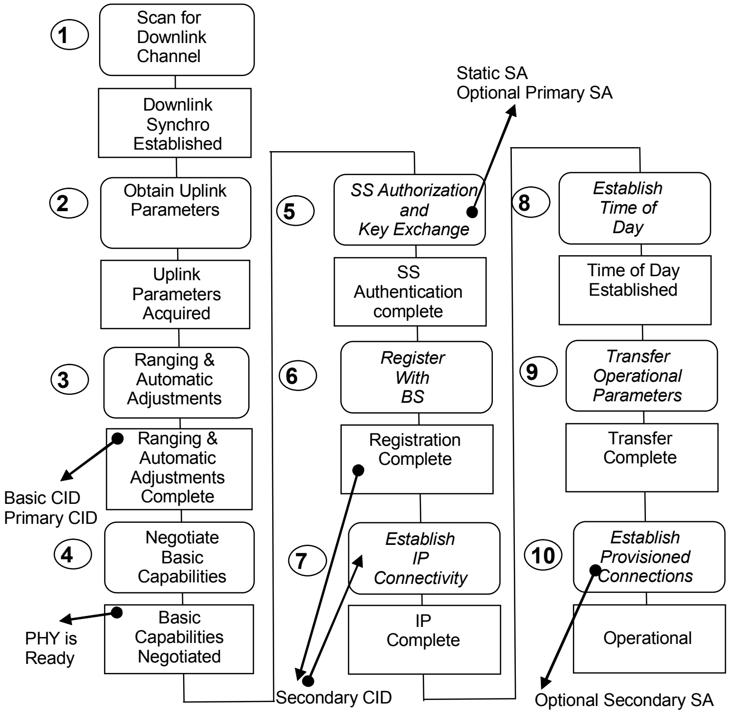

1
Scan for Downlink Channel
Downlink Synchro Established
2
Obtain Uplink Parameters
Uplink Parameters Acquired
3
Ranging & Automatic Adjustments
Ranging & Automatic Adjustments Complete
Basic CID
Primary CID
4
Negotiate Basic Capabilities
Basic Capabilities Negotiated
PHY is Ready
5
SS Authorization and Key Exchange
Static SA
Optional Primary SA
SS Authentication complete
6
Register With BS
Registration Complete
7
Establish IP Connectivity
IP Complete
Secondary CID
8
Establish Time of Day
Time of Day Established
9
Transfer Operational Parameters
Transfer Complete
10
Establish Provisioned Connections
Operational
Optional Secondary SA

Figure 8.10. *Subscriber insertion procedure*

– *Ranging and automatic adjustments*. Thanks to the Ranging Request (RNG-REQ) and Ranging Response (RNG-RSP), the subscriber adjusts his transmission power and collects additional information from the BS. The basic connection ID and the primary management CID parameters are affected to the subscriber by the BS, and notified in the RNG-RSP response.

– *Negotiation of the transmission parameters*. At the end of the calibration procedure, the subscriber informs the BS of its capacities and performs a negotiation.

– *Authorization and key exchange*. Thanks to the PKM-REQ (privacy key management request, MngtType= 9) and PKM-RESP (privacy key management response) management messages the SS is authenticated and upon success computes and collects a set of cryptographic keys. This protocol deals with the primary management CID.

– *Registration*. Thanks to this procedure, the subscriber becomes an active member of the network. The registration request (REG-REQ) and registration response (REG-RSP) messages, authenticated by a HMAC-tuple, allow to obtain a secondary management CID, more particularly used for IP services such as DHCP.

– *The establishment of the IP connectivity*. The IP version used by the subscriber is indicated in the REG-REQ message. The subscriber obtains an IP address thanks to the classical DHCP protocol (described by the RFC 2131 [IET 97]).

– *Acquisition of the date and time*. The subscriber collects these parameters thanks to the protocol defined by the RFC 868 [IET 83].

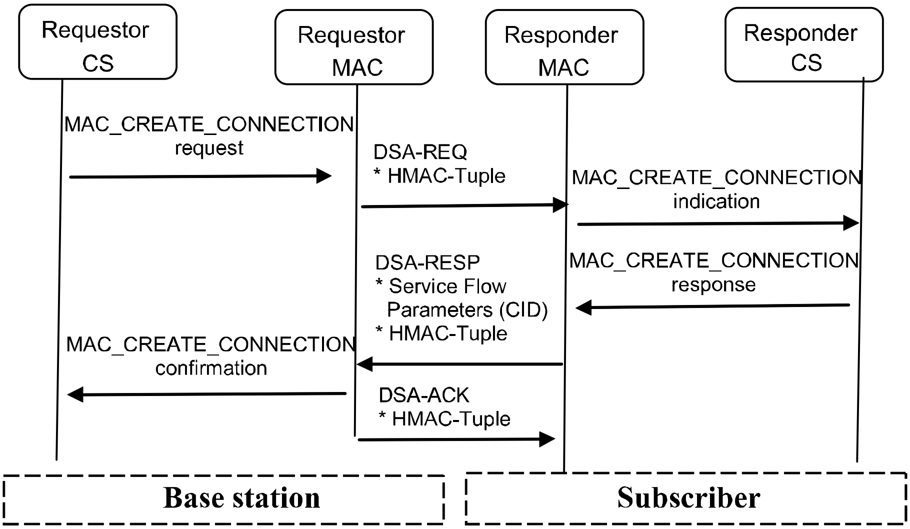

Requestor CS
Requestor MAC
Responder MAC
Responder CS
MAC_CREATE_CONNECTION request
DSA-REQ
* HMAC-Tuple
MAC_CREATE_CONNECTION indication
DSA-RESP
* Service Flow Parameters (CID)
* HMAC-Tuple
MAC_CREATE_CONNECTION response
MAC_CREATE_CONNECTION confirmation
DSA-ACK
* HMAC-Tuple
Base station
Subscriber

Figure 8.11. *Provisioned services*

– *Downloading of the operational parameters*. The subscriber obtains a configuration file thanks to the TFTP protocol (Trivial FTP, RFC 1123 and 2349 [IET 89, IET 98]).

– *Establishment of provisioned connections*. The BS delivers DSA-REQ (Dynamic Service Additional Request) messages to the client in order to establish prepaid services. These messages are acknowledged by the DSA-RESP (Dynamic Service Additional Response).

The allocation of provisioned connections by the BS is mandatory. In an optional way, the subscriber dynamically creates a connection with a particular QoS. The MAC messages (DSA-REQ, DSA-RESP, etc.) are authenticated by HMAC-Tuples and collect the necessary information, such as the CID and SFID parameters, needed to use network services.

Figure 8.11 illustrates the establishment of a provisioned connection. A connection request is translated by a DSA-REQ message authenticated by a HMAC-tuple associated with a signature key, which is computed during the authorization phase. The acceptance of this request is notified by a DSA-RESP signed by a HMAC-Tuple. This last message includes attributes specifying the quality of the service flow (SFID) and the CID inserted in a DSA-ACK frame.

8.3. Security according to 802.16-2004

The entire security of the 802.16-2004 standard relies on the PKM protocol which realizes the subscriber authentication and negotiates a set of cryptographic algorithms and their associated keys. The PKM protocol is an inheritance from the IEEE 802.14 standard (cable-TV access method and physical layer specification) and then from DOCSIS (Data-Over-Cable Service Interface Specifications). It is transported by PKM-REQ or PKM-RESP type MAC management messages (respectively requests and responses).

After this negotiation, the following MAC frames are secured between the subscriber and the BS:

– the MAC data frames are ciphered and optionally their integrity is checked;

– the integrity of MAC management frames is guaranteed using the keyed MAC algorithm (HMAC).

These security functions are based on a set of four keys – AK, KEK, TEK and HMAC – whose characteristics are summarized in Table 8.2. The AK key plays an essential role because it is the root for the calculation of the KEK and HMAC keys.

Keys	Features
Authorization Key (AK)	This key is transmitted by the BS, and encrypted thanks to the subscriber's public key. The KEK and HMAC keys are directly calculated from the AK value.
Key Encryption Key (KEK)	This key value is deduced from AK by the BS and the subscriber. It is used for the encryption and the decryption of the TEK keys.
Traffic Encryption Key (TEK)	This key is delivered by the BS to the subscriber. The key value is encrypted by the KEK key according to the algorithm negotiated during the PKM exchange. It is used for the encryption of the data frames.
HMAC key HMAC_KEY_D HMAC_KEY_U HMAC_KEY_S	The HMAC keys are deduced from the AK value. They are associated with the HMAC algorithm and allow us to authenticate uplink HMAC_KEY_U) and downlink (HMAC_KEY_D) management frames. The HMAC_KEY_S is only used for MESH infrastructures.

Table 8.2. *Summary of the symmetric keys defined in 802.16-2004*

The authentication and the cryptographic key distribution mechanisms are ruled by two state machines: the authentication state machine and the TEK distribution state machine.

8.3.1. *Authentication, authorization and key distribution*

8.3.1.1. *PKM authentication and authorization by the PKM protocol*

The PKM authentication procedure realizes the authentication of a subscriber and upon success obtains an AK delivered by the BS. It occurs after the negotiation of the basic capabilities with the BS. It is the fifth step in Figure 8.10. It exchanges three PKM messages (see Figure 8.12):

– The first message is the authorization information message that includes the X.509 (RFC 2459, [IET 99]) certificate of the SS.

– The second message is an authorization request message including the following data:

- the X.509 certificate of the subscriber (SS),

- a list of the cryptographic suites supported by the subscriber. Each cryptographic suite identifies three types of algorithms (the frame encryption algorithm, the frame MAC algorithm and the cipher algorithm dealing with the KEK and used for TEK encryption),

- the basic subscriber's CID, in other words the first CID delivered by the BS during the establishment of the primary management channel. This parameter also constitutes the primary SAID of the security association.

– The third message is the response to the *authreply* request produced by the BS. It embeds:

- the AK encrypted with the subscriber's public key according to RSAES-OAEP coding rules [PKCS 1] (see section 8.3.3.1),

- a key sequence number, pointing the current AK,

- the AK lifetime,

- a list of security associations (identified by their respective SAID) for which the subscriber may collect the TEK.

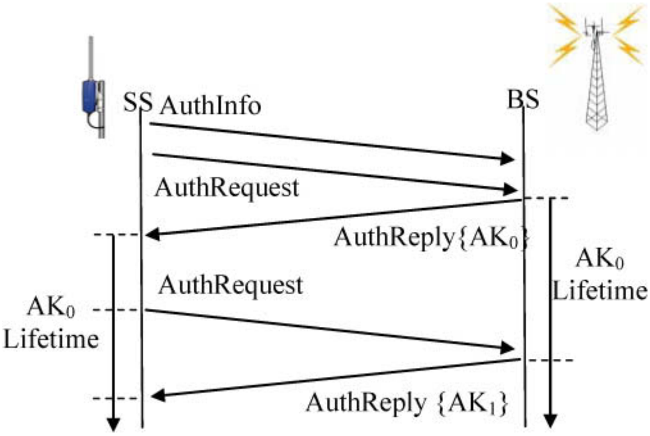


Figure 8.12. *Authorization procedure*

At the end of the AK lifetime, as the first AK is called AK_0, a new authorization session begins, and leads to a new AK_1.

8.3.1.2. *TEK key distribution procedure*

As illustrated by Figure 8.13, the distribution procedure of TEKs to the subscriber is based on two management messages as follows:

– The subscriber transmits a *keyrequest* associated with a particular security association identifier (SAID) in order to obtain a couple of TEKs (TEK_0, TEK_1) linked to a particular SAID security association.

– The BS responds using the management message, *keyreply* (or *KeyReject* in the event of problems). The *keyreply* message includes the following information:

- a key sequence number, i.e. the index of an AK,

- a SAID identifier,

- two (encrypted) TEKs; each of them is used for the encryption of both uplink and downlink frames,

- a key sequence number for each TEK,

- a key lifetime for each TEK,

- the initialization value (IV) for each TEK, because the encryption algorithms work according to the chained mode (CBC),

- a HMAC digest.

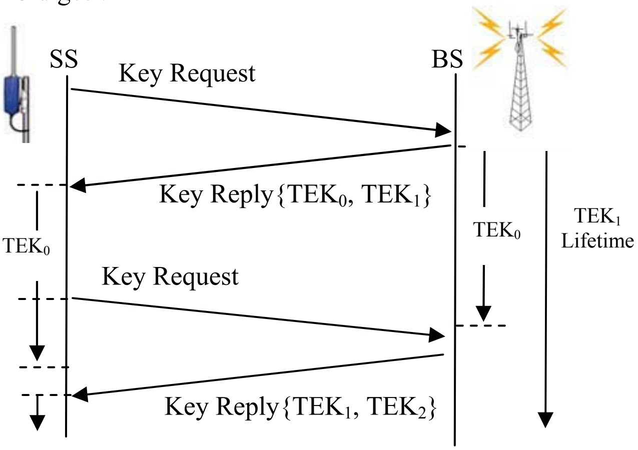


Figure 8.13. *TEK key distribution procedure*

The TEK keys are transmitted and ciphered by the BS thanks to the algorithm pointed by the SAID associated with a key sequence number (the AK index).

The CBC-IV of the downlink and uplink frames are obtained by the exclusive or (EOR) of the IV field, associated with the current TEK and with the synchronization attribute included in DL-MAP messages.

It should be noted that *keyrequest* and *keyreply* messages are authenticated and their integrity is checked thanks to the HMAC attribute calculated with the HMAC_KEY_U/D key.

When a TEK ends, a new key distribution session is initiated.

8.3.2. *Security associations*

A security association is a set of information that makes it possible to handle the authentication procedure or to secure frames exchanged over radio links. These collections of parameters include, among others, cryptographic algorithms, keys and their lifetime.

8.3.2.1. *Security association for management frames authentication*

In order to guarantee the authentication and the integrity of the data frames, the security associations shared between the subscriber and the BS include the following data:

– the subscriber‘s X.509 certificate;

– an AK key of 160 bits;

– a key sequence number;

– the AK lifetime (70 days by default);

– a KEK associated with a cryptographic algorithm for TEK encryption (triple DES for example);

– two keys of 160 bits for the downlink and uplink communications and associated with HMAC algorithms;

– a key of 160 bits, dedicated to signing operations in the MESH infrastructures.

In this chapter, the MESH infrastructure is not detailed in order to keep things concise. The MESH key is calculated from a secret value called the operator shared secret (see Figure 8.2). It is involved in the authenticity and the integrity of some management messages such as *keyrequest* (see section 8.3.1.2).

8.3.2.2. *Security associations for data coding*

This is a list of parameters which guarantee the security of the exchanges between one or several subscribers and a BS. There are three types of SA: *primary*, defined during the subscriber initialization procedure (and identified by the basic CID), *static*, attributed by the BS, and *dynamic* used for particular data services.

The security association includes the following data:

– a 16 bit identifier (SAID);

– an encryption algorithm, the standard IEEE 802.16-2001 only supports DES-CBC;

– two TEK encryption/decryption keys;

– two key sequence numbers of 2 bits for the TEK;

– lifetimes of TEK keys (30 minutes by default);

– the IVs CBC-IV (64 bits) associated with a TEK because the encryption algorithm works according to a chain mode;

– the type of security association: primary, static or dynamic.

8.3.3. ***Cryptographic elements***

The IEEE-802.16-04 [IEE 04] standard is based on several cryptographic credentials that make it possible to compute keys, but also perform frame encryption or HMAC calculations.

8.3.3.1. *Encryption and decryption of the AK*

The AK plays an essential role in the organization of security. It is encrypted with the *RSAES-OAEP* algorithm defined by the [PKC 01] standard. The RSA algorithm realizes an encryption thanks to the exponent of the private or public key. The processed number must be "big", i.e. its size must be similar in magnitude to the modulus.

For example, if the public exponent is three, and the modulus is about 1,024 bits, the magnitude of the AK^3 is about 2^{480}, which is lower than 2^{1024}. As a consequence, it is trivial to find the AK value by calculating the cubic root.

RSAES-OAEP is a way to construct a "big" number from a "small" number (such as an AK), i.e. a value whose magnitude is about 127 bytes.

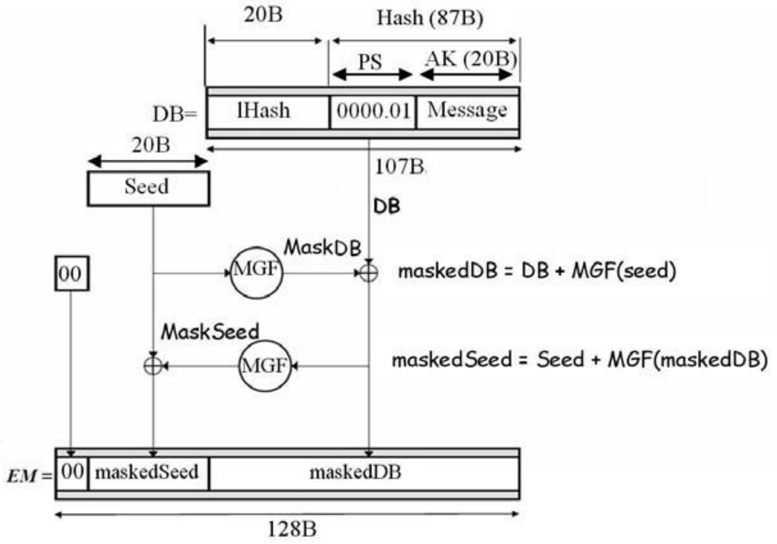


Figure 8.14. *RSAES-OAEP*

This process, in the case of a modulus of 128 bytes, is realized in accordance with Figure 8.14:

– The AK (20 bytes) is completed by a list (*PS*) of 67 bytes, all zero (00), except the last one, whose value is set to 01.

– A SHA1 digest (*lhash*) is calculated for the previous PS|AK set where the notation "|" indicates the concatenation operation. As a result a number DB = lHash| PS|AK is obtained, whose size is 107 bytes.

– A seed of 20 bytes (a random value) is used as an input value for a mask generator function (MGF), in order to produce a set of 107 bytes, the *MaskDB*. An EOR operation between MaskDB and DB leads to the *maskedDB* value.

– The MGF function is applied one more time to the maskedDB parameter and returns a maskSeed value. An EOR between maskSeed and Seed produces the 20 byte parameter maskedSeed.

– A zero byte (00) is concatenated to the maskedseed and maskedDB values.

– A RSA calculation, which deals with the subscriber's public key, is performed on the previously built number (00|maskedSeed|maskedDB).

The decryption operation comprises a first calculation using the subscriber's private key; it makes it possible to obtain the value 00|maskedSeed|maskedDB. The MGF(*maskedDB*) function makes it possible to again find the maskseed value and thus the seed value. As a result we deduce *MaskDB*= MGF(*Seed*), and then DB.

8.3.3.2. *Calculation of the KEK and HMAC keys*

The KEK and the keys associated with the HMAC algorithms are deduced from the AK thanks to the following procedures:

– KEK= *Truncate*(SHA1(K_PAD_KEK | AK),128), the value K_PAD_KEK being a fixed number of 512 bits. SHA1 returns a value of 20 bytes; as a result, KEK is equivalent to the most significant 128 bits (that is, 16 bytes) of the hash value produced by SHA1. Indeed, *Truncate*(x,n) indicates a truncation operation of the n most significant bits (left part) of a list of x bits;

– HMAC_KEY_D= SHA1(H_PAD_D | AK);

– HMAC_KEY_U= SHA1(H_PAD_U | AK);

– HMAC_KEY_S = SHA1(H_PAD_D | OperatorSharedSecret).

H_PAD_D and H_PAD_U values are fixed 512 bit numbers. HMAC_KEY_S requires for its calculation the knowledge of a secret shared with the operator ***(OperatorSharedSecret)***

8.3.4. *Crypto-suites for TEK encryption with KEK*

A crypto-suite is a set of encryption algorithms associated with TEKs whose size ranges between 64 and 128 bits. Two crypto-suites are usually used for TEK encryption by the KEK (16 bytes) and are summarized in Table 8.3.

Crypto-suites	Encryption algorithms (KEK)	Size of TEKs
Crypto-suite 01	Triple DES (3-DES)	64 bits
Crypto-suite 03	AES-128	128 bits

Table 8.3. *Examples of TEK crypto-suites*

8.3.5. *Crypto-suites for the data frames associated with the TEK*

8.3.5.1. *Crypto-suite 01, DES-CBC algorithm*

The encryption only applies to the payload of a MAC frame (see Figure 8.15). The 802.16-2001 standard works with the DES-CBC mode (using a 56 bit key and a 64 bit IV) and does not support data integrity mechanisms.

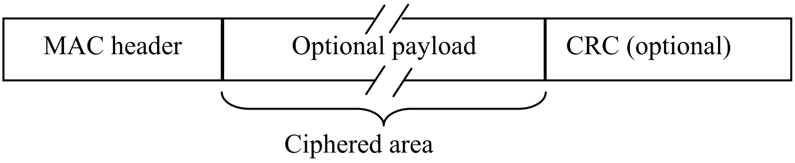


Figure 8.15. *DES encryption of a MAC data frame*

8.3.5.2. *Crypto suite 03, AES algorithm*

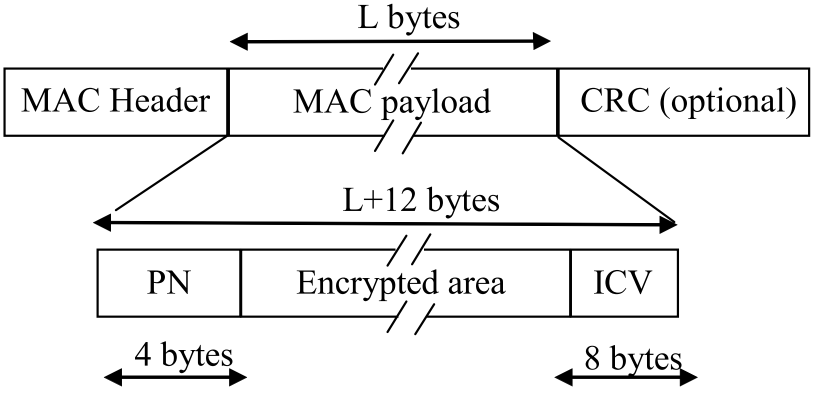


Figure 8.16. *AES-CCM encryption of a MAC frame*

Data are ciphered according to the CCM mode, associated with the AES algorithm, such as those defined by the *NIST* standard *Special Publication 800-38 C*, *FIPS-197*. As illustrated in Figure 8.16, MAC payload includes an ICV (Integrity Check Value) suffix, which is intended for integrity checking, and a PN prefix that prevents for replay issues.

8.3.6. *A brief overview of the IEEE 802.16-2004 threats*

WiMAX is a radio network with vulnerabilities similar to those of Wi-Fi. These radio infrastructures deployed by companies such as Alvarion were in accordance with the 802.16-2001 standards. Many telecommunication manufacturers actively work on the IEEE-802.16e standard, which is definitely more attractive than the 2004 version in terms of services. A first 802.16e network called WiBRO (for wireless broadband) has been set up in South Korea, and uses an authentication procedure based on the EAP-AKA protocol (RFC 4187, [IET 06]).

This section presents a brief analysis of the potential weaknesses of WiMAX. However, because of the lack of practical experience, efficient attacks are not observed or known yet.

We classify attacks in two categories: attacks at the PHY level and attacks at the MAC level.

8.3.6.1. *Attacks at the PHY level*

It is possible to create interferences to the radio signals of the uplink and downlink channels, thanks to adapted noise generators. We distinguish two types of processes, jamming and scrambling:

– Jamming generates large interferences on the uplink and/or downlink channels, and implies a denial of service on the network. As a result, radio links are no longer functional. However, it is easy to locate the geographical position of the attacker by classical goniometry techniques. As a consequence, the hacker may be sued.

– Scrambling causes interference to a small part of the uplink or downlink channels. It may decrease the QoS, causing the retransmission of particular MAC frames for example.

8.3.6.2. *Attacks at the MAC level*

At the MAC level several classes of attacks may be realized such as generation of erroneous data frames, identity theft of subscribers, or rogue BS. However, most of these weaknesses are corrected by the 802.16e standard:

– *Generation of erroneous date frames*. When a security association dealing with MAC data frames does not support integrity mechanisms, the attacker forges wrong and offensive frames. Their content, once it has been decrypted without integrity checking, will be hazardous in practice and may provoke software crash that parse the frame. This weakness only concerns the DES encryption mode, without integrity features, which is imported from the 802.16-2001 standard [IEE 01].

– *Masquerade of a subscriber*. Some management messages such as *AuthInfo* or *AuthRequest* are not authenticated (they do not include HMAC digests). As a result, an attacker forges and transmits such packets. He hopes to flood the network and, for example, creates an overload of certificate checking, which may induce a denial of authorization services. It should also be noted that a subscriber's certificate is transmitted in clear text. Consequently, it is possible to associate identity and geographical position, which gives rise to a potential privacy threat. However, this last problem is corrected with techniques such as the double session EAP, which has been introduced by the IEEE 802.16e standard.

– *Rogue BS*. The IEEE 8012.16-2001 and IEEE 802.16-2004 standards only require a simple authentication of the subscriber by the network. Thanks to the probable low cost of WiMAX technology, hackers could design and deploy rogue BSs in order to intercept the communications of network subscribers. This is called a *man-in-the-middle* attack. The first Wi-Fi infrastructures also faced this kind of problem, which has been solved by the IEEE 802.1x [IEEE 802.1x 04] standard. For this reason, the IEEE 802.16e standard requires mutual authentications between subscribers and BSs.

8.4. Security according to the IEEE-802.16e standard

The IEEE-802.16e-2006 standard [IEE 06] improves the previous version of 802.16-2004 (mutual authentication between subscribers and BSs, stronger cryptographic algorithms, authentication servers, etc.) and enables communications facilities from a car. It introduces network broadband accesses, intended for fixed or mobile applications. It also includes recommendations in order to manage handovers, that is, rapid changes of BSs. It works with frequency bands smaller than 6 GHz, whose use is subject to a license.

The network architecture (see Figure 8.17) includes mobile stations (MSs) communicating with BSs. BSs are connected to an operator backbone network, which usually includes an authentication and service authorization server (ASA). This is a database that centralizes all the data of the subscriber's subscriptions and the credentials required for their authentication.

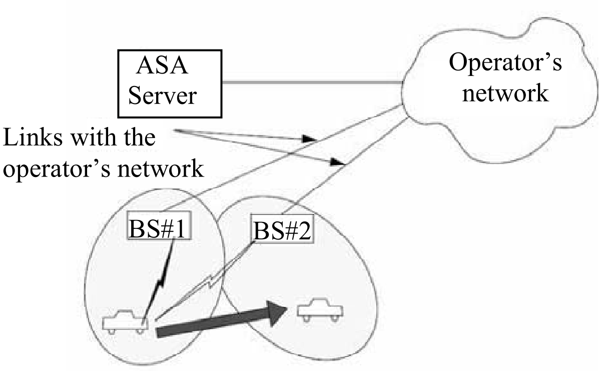


Figure 8.17. *IEEE 802.16e architecture*

The standard introduces the functional interfaces (see Figure 8.18) dealing with mobile services, without describing the underlying protocols in a precise manner. The U interface rules the services between MSs and BSs. The IB interface transports messages between BSs, which are required for the management of handover procedures.

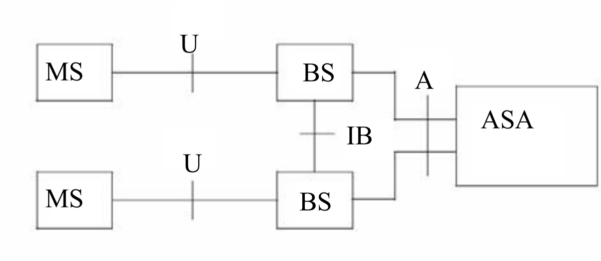


Figure 8.18. *IEEE 802.16 functional interfaces*

The standard identifies two classes of infrastructures. The first is not linked to an operator, while the second is typically ruled by a mobile phone operator. According to these constraints, but also for legacy issues with the previous versions, two types of authentication mechanisms are defined: PKM-RSA imported from IEEE 802.16-2004 and PKM-EAP enabling the reuse of the EAP, more precisely described by the RFC 3748 [IET 04].

Two versions of the privacy key management protocol are proposed. The first one, PKMvl, is compatible with environments in accordance with the IEEE 802.16-2004. The second one, PKMv2, supports new features such as:

– a mutual authentication between BSs and MSs;

– the use of mechanisms based on RSA and/or the EAP protocol;

– a modified hierarchy of keys (see section 8.4.1 for more details);

– the replacement of the HMAC-SHA1 procedure, based on the SHA1 digest (whose the cryptographic strength is questionable) with the AES-CMAC algorithm (more precisely defined by [NIS 05] and [IET 06]);

– a new encryption method, *AES-key-wrap,* addressing the transport of TEKs. This algorithm, which is recommended by the NIST, realizes AES encryption with a 128 bit key and includes an integrity check value (ICV). It increases the security of the TEK distribution process;

– the notion of pre-authentication, e.g. a protocol that makes it possible for mobiles and BSs, to share an authentication key, without a mutual authentication procedure. The 802.16e standard does not define a particular method for the calculation of AK, but note that it could rely on parameters such as the subscriber's MAC address and the BS identifier;

– the multicast broadband service (MBS). As stated in its name, it is intended for data broadcasting, e.g. multimedia features. These security mechanisms make it possible, for example, to effectively deploy *Pay TV* services.

In this section, we only describe new features introduced by the PKMv2 protocol. Note that this protocol requires the mobile to be equipped with a couple of public/private keys and a X.509 certificate. A large number of keys are exchanged or computed between mobiles and BSs. For a better understanding of this section, the reader should start with the PKMv2-RSA and PKMv2-EAP authentication procedures, and should refer to the summary in Table 8.4.

At the end of these authentication procedures, keys are shared between MSs and BSs. More precisely, and with PKMv2-EAP, a master session key (MSK) is shared between BSs and MSs. The AK is computed from this value and the *Dot16KDF* algorithm. Keys required by the MBS are deducted from a MAK (the MAS authorization key, MBS AK), distributed according to procedures that are not covered by the 802.16e standard.

8.4.1. *Hierarchy of the keys*

The summary of the keys and their calculation is given by Table 8.4 and the detailed structure of the Dot16KDF function is introduced in section 8.4.1.1.

Keys	Characteristics
Pre Primary AK Pre-PAK	This key is managed by the BS, and encrypted by the subscriber's public key, during an optional PKM-RSA process.
Primary AK PAK	This key is deduced from the pre-PAK key, thanks to the Dot16KDF function and input parameters such as the subscriber's MAC address and the BS identifier. This value is involved in the calculation of the AK.
Master Session Key MSK	This key is obtained at the end of a first EAP authentication session. It is used for the calculation the EIK and PMK keys.
EAP Integrity Key EIK	This key is calculated from the pre-PAK or the MSK. It is used for authenticating EAP messages, during the first occurrence (EIK=f(pre-PAK)) or for other occurrences (EIK=f(MSK)) of an authentication session.
Master Session Key 2 MSK2	This key is obtained from a second EAP authentication session. It is used for the calculation of the PMK2.
Pairwise Master Key PMK	This key is calculated from the MSK value. It is used for the calculation of the AK.
Pairwise Master Key 2 PMK2	This key is deducted from the MSK2 value. It is used for the calculation of the AK.
Authorization Key AK	This is obtained thanks to the Dot16KDF function with input parameters such as the PAK, PMK, PMK2, subscriber's MAC address and BS identifier.

Key Encryption Key KEK	The KEK is deduced from the AK value. It is used for the encryption of TEKs.
Traffic Encryption Key TEK	This key, is generated by the BS and is transmitted encrypted to the subscriber thanks to the KEK. It is used for the encryption of data frames.
CMAC or HMAC keys used for uplink channels C/HMAC_Key_U	In general, this key is deduced from the AK, the subscriber's MAC address and from the BS identifier. It authenticates uplink messages.
CMAC or HMAC used for downlink channels C/HMAC_Key_D	In general, this key is deduced from the AK, the subscriber's MAC address and from the BS identifier. It authenticates downlink messages.
Group Key Encryption Key GKEK	This key is generated by the BS and transmitted encrypted to the subscriber thanks to the TEK. It is used for the encryption of the GTEK
CMAC or HMAC group keys used for downlink channels C/HMAC_Key_GD	This key is calculated from the GKEK. It is used for some messages of the PKMv2 protocol.
Group Traffic Encryption Key GTEK	This key is produced in a random fashion by the BS and is transmitted to the subscribers, encrypted by GKEK. It is used for transmitting data to the members of a group.
MBS Transport Key MTK	This key is deduced from a GTEK and from a secret value MAK (MBS AK) whose distribution is not described by the standard. This value is used for broadcasting services such as Pay TV.

Table 8.4. *Summary of the symmetric keys defined in IEEE 802.16e*

8.4.1.1. *The Dot16KDF function*

The Dot16KDF (*Key Derivation Function*) algorithm is based on the CMAC or HMAC procedures according to the authentication policy.

The pseudocode introduced in Figures 8.19 and 8.20 unveils the calculation details dealing with CMAC and HMAC functions. The character "|" indicates a concatenation operation. The Truncate function (binary-value, n) realizes the extraction of the more significant (n) bits (left part) of a binary value.

```
Dot16KDF(key, astring, keylength), using the CMAC algorithm
{result = null;
Kin = Truncate (key, 128);
for (i = 0;i <= int((keylength-1)/128); i++){
result <= result | Truncate (CMAC(Kin, i | astring | keylength), 128);}
return Truncate (result, keylength);}
```

Figure 8.19. *Dot16KDF pseudocode based on the CMAC function*

```
Dot16KDF(key, astring, keylength), using the HMAC algorithm
{Kin = Truncate (key, 160);
return Truncate (SHA-1(astring | Kin), keylength);}
```

Figure 8.20. *Dot16KDF pseudocode based on the HMAC function*

8.4.1.2. *The AK and the pre-PAK, MSK, EIK, PMK and PMK2*

Four authentication scenarii are possible in the PKMv2 context: (1) a mutual RSA authentication (without EAP session), (2) a single or (3) double EAP session (without RSA pre-authentication), (4) a RSA pre-authentication and a single session EAP.

At the end of a PKMv2-RSA authentication procedure (see section 8.4.2), a pre-PAK is pushed by the BS towards the MS, encrypted with its public key.

Thanks to the Dot16KDF function, two keys are calculated, respectively PAK (160 bits) and EIK (160 bits), calculated according to the following formula:

EIK | PAK = Dot16KDF(pre-PAK, SS_MAC_Address | BSID | "EIK+PAK", 320)

in which SS_MAC_Address indicates the MAC address of the mobile and BSID the identifier of the BS.

If an RSA authentication occurs before an EAP session, the EAP messages are protected by the EIK. At the end of the PKMv2-EAP authentication session (see section 8.4.3) and in accordance with the rules of the EAP, an MSK (512 bits) is available. A pair of EIK (160 bits) and PMKs (pairwise master key, 160 bits) is built, thanks to the following relation:

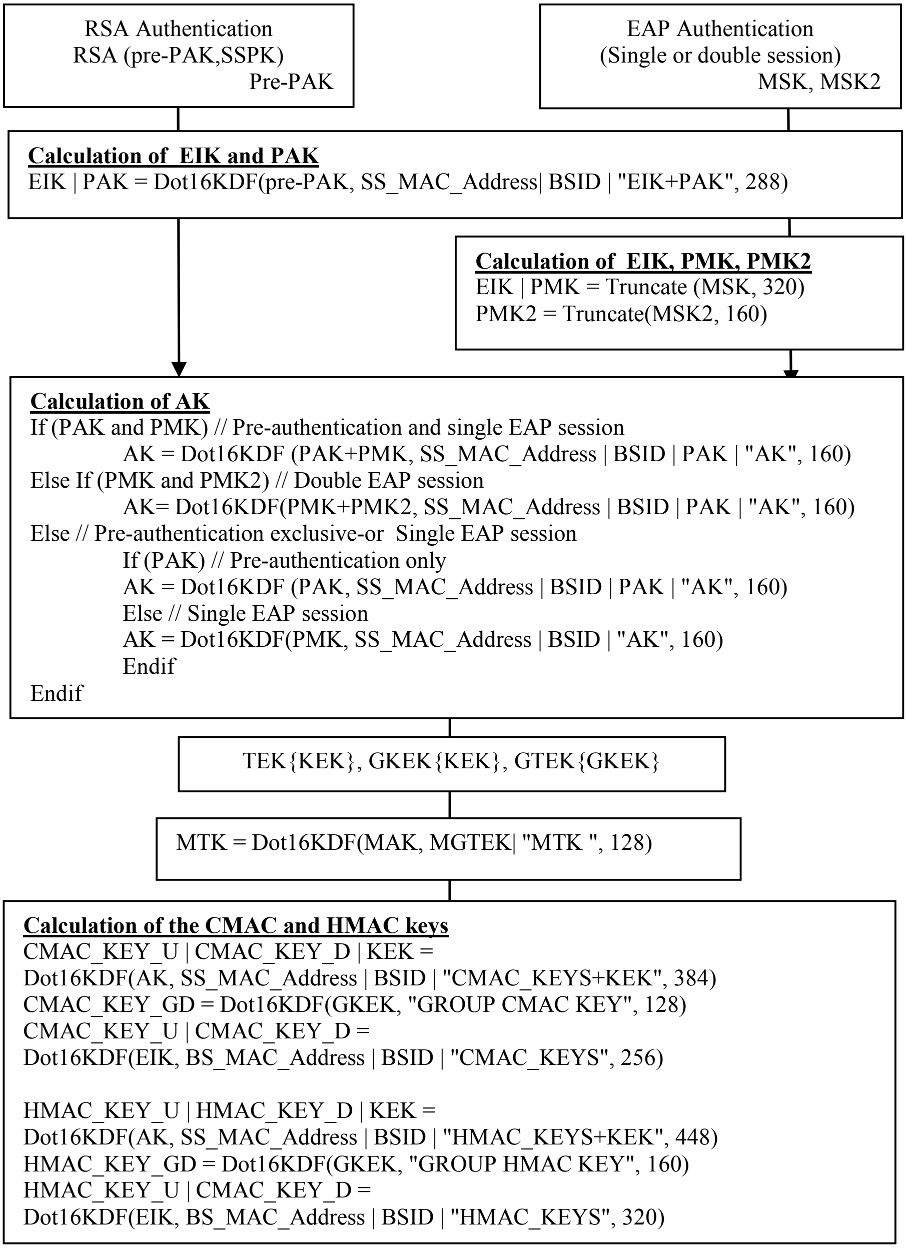

RSA Authentication
RSA (pre-PAK,SSPK)
Pre-PAK
EAP Authentication
(Single or double session)
MSK, MSK2
Calculation of EIK and PAK
EIK | PAK = Dot16KDF(pre-PAK, SS_MAC_Address| BSID | "EIK+PAK", 288)
Calculation of EIK, PMK, PMK2
EIK | PMK = Truncate (MSK, 320)
PMK2 = Truncate(MSK2, 160)
Calculation of AK
If (PAK and PMK) // Pre-authentication and single EAP session
AK = Dot16KDF (PAK+PMK, SS_MAC_Address | BSID | PAK | "AK", 160)
Else If (PMK and PMK2) // Double EAP session
AK= Dot16KDF(PMK+PMK2, SS_MAC_Address | BSID | PAK | "AK", 160)
Else // Pre-authentication exclusive-or Single EAP session
If (PAK) // Pre-authentication only
AK = Dot16KDF (PAK, SS_MAC_Address | BSID | PAK | "AK", 160)
Else // Single EAP session
AK = Dot16KDF(PMK, SS_MAC_Address | BSID | "AK", 160)
Endif
Endif
TEK{KEK}, GKEK{KEK}, GTEK{GKEK}
MTK = Dot16KDF(MAK, MGTEK| "MTK ", 128)
Calculation of the CMAC and HMAC keys
CMAC_KEY_U | CMAC_KEY_D | KEK =
Dot16KDF(AK, SS_MAC_Address | BSID | "CMAC_KEYS+KEK", 384)
CMAC_KEY_GD = Dot16KDF(GKEK, "GROUP CMAC KEY", 128)
CMAC_KEY_U | CMAC_KEY_D =
Dot16KDF(EIK, BS_MAC_Address | BSID | "CMAC_KEYS", 256)
HMAC_KEY_U | HMAC_KEY_D | KEK =
Dot16KDF(AK, SS_MAC_Address | BSID | "HMAC_KEYS+KEK", 448)
HMAC_KEY_GD = Dot16KDF(GKEK, "GROUP HMAC KEY", 160)
HMAC_KEY_U | CMAC_KEY_D =
Dot16KDF(EIK, BS_MAC_Address | BSID | "HMAC_KEYS", 320)

Figure 8.21. *Summary of key calculations in the IEEE 802.16e standard*

EIK | PMK = Truncate (MSK, 320)

When a second EAP session occurs, it produces a MSK2 (512 bits), from which a PMK2 (160 bits) is obtained according to:

PMK2 = Truncate (MSK2, 160)

According to the available keys (PAK, PMK, PMK2) the AK (160 bits) is calculated in accordance with the pseudocode described by Figure 8.21 (the sign + indicates an exclusive or operation).

8.4.1.3. *KEKs and TEKs*

The KEK (128 bits) is calculated thanks to the Dot16KDF function. It is used for example for the encryption of the TEKs and GKEKs delivered by the BS. The detail of this calculation is introduced in section 8.4.5. The TEK is generated in a random way by the BS and is afterwards encrypted by the KEK.

8.4.1.4. *GKEKs and GTEKs*

The GKEK (128 bits) is generated in a random fashion by the BS, and is then encrypted by the cryptographic algorithm associated with TEK.

The GTEK is produced in a random fashion by the BS, and is then encrypted by the GKEKs and TEKs in multicast or unicast messages.

8.4.1.5. *MTK*

The MTK is deduced from the MAK, whose distribution method is not described by the IEEE 802.16e standard:

MTK = Dot16KDF(MAK, MGTEK | "MTK", 128)

MGTEK is the current GTEK, associated with the MBS service. MTK is used for MBS traffic encryption, and thus for every type of service based on broadcasting mechanisms (Pay TV, etc.).

8.4.1.6. *MAC keys*

The keys dealing with the CMAC algorithm (CMAC_KEY_U and CMAC_KEY_D) are deduced from the following relations:

CMAC_KEY_U | CMAC_KEY_D | KEK =

Dot16KDF(AK, SS_MAC_Address | BSID | "CMAC+KEK", 384)

An additional key is used for some PKMv2 messages (such as the PKMv2 Group-Key-Update-Command):

CMAC_KEY_GD = Dot16KDF(GKEK, "GROUP CMAC KEY", 128).

The keys associated with the HMAC algorithm are deduced by the following relations:

HMAC_KEY_U | HMAC_KEY_D | KEK =

Dot16KDF(AK, SS_MAC_Address | BSID | "HMAC_KEYS+KEK", 448)

An additional key is used for some PKMv2messages (such as PKMv2 Group-Key-Update-Command):

HMAC_KEY_GD = Dot16KDF(GKEK, "GROUP HMAC KEY", 160).

The EAP messages are associated with CMAC or HMAC signatures whose keys (128 or 160 bits) are deduced from an EIK, thanks to the following procedures (respectively when the CMAC or HMAC algorithms are selected):

CMAC_KEY_U | CMAC_KEY_D =

Dot16KDF(EIK, SS_MAC_Address | BSID | "CMAC_KEYS", 256) or

HMAC_KEY_U | HMAC_KEY_D =

Dot16KDF(EIK, SS_MAC_Address | BSID | "HMAC_KEYS", 320)

The SA-TEK 3-way handshake protocol uses an identifier for the AK, called AKID (64 bits) and obtained by the following relation:

AKID = Dot16KDF(AK, AK_SN | SS_MAC_Address | BSID | "AK", 64)

the byte AK_SN being obtained by adding four zero bits to the index of the AK.

8.4.2. *Authentication with PKMv2-RSA*

The protocol PKMv2 RSA (see Figure 8.22) uses four types of messages: PKMv2 RSA-Request, PKMv2 RSA-Reply, PKMv2 RSA-Acknowledgement or PKMv2 RSA-Reject.

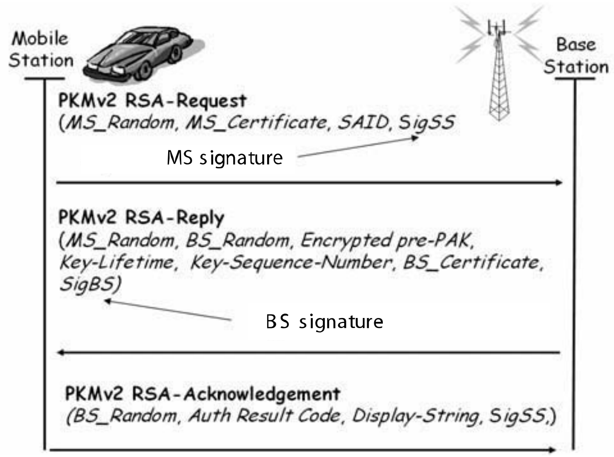

Mobile Station
Base Station
PKMv2 RSA-Request
(MS_Random, MS_Certificate, SAID, SigSS
MS signature
PKMv2 RSA-Reply
(MS_Random, BS_Random, Encrypted pre-PAK, Key-Lifetime, Key-Sequence-Number, BS_Certificate, SigBS)
BS signature
PKMv2 RSA-Acknowledgement
(BS_Random, Auth Result Code, Display-String, SigSS)

Figure 8.22. *PKMv2-RSA protocol*

The MS generates a random number (MS_Random) and inserts it into a RSA-Request message, which also transports its X.509 certificate, a security association identifier (SAID) and a RSA signature (SigSS) realized thanks to a private key.

The BS checks the signature of the received message and produces a RSA-Reply message, which includes the MS_Random number previously provided by the subscriber, a random number (BS_Random) generated by the BS, its X.509 certificate, a pre-PAK key encrypted with the public key of the mobile subscriber, and a signature (SigBS) calculated with the private BS key. The fact that the same number is returned MS_Random guarantees the subscriber that this message has been triggered by its request.

The mobile analyzes the RSA-Reply message and indicates the success of the authentication operation with an RSA-Acknowledgment message, identified by the DS-Random field, which includes the operation status (AuthResult), optional information (Display-String) and a SigSS signature.

In the event of this procedure the two entities calculate an EIK being successful (used for EAP exchange security) and a PAK.

8.4.3. *Authentication with PKMv2-EAP*

In the case of a single PKMv2-EAP session (see Figure 8.23), the packets are transported by PKMv2 Authenticated-EAP-Transfer messages (signed by an EIK), or PKMv2 EAP-Transfer if a RSA mutual authentication procedure has not previously been used (because in that case no EIK is available).

When the use of a double session has been negotiated between the two entities (MS and BS), the subscriber indicates in an optional way the beginning of the dialogue by a PKMv2 EAP-Start message including no attribute (see Figure 8.24).

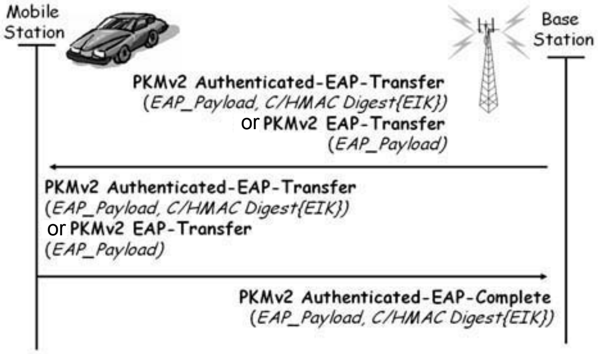

Mobile Station
Base Station
PKMv2 Authenticated-EAP-Transfer
(EAP_Payload, C/HMAC Digest{EIK})
or PKMv2 EAP-Transfer
(EAP_Payload)
PKMv2 Authenticated-EAP-Transfer
(EAP_Payload, C/HMAC Digest{EIK})
or PKMv2 EAP-Transfer
(EAP_Payload)
PKMv2 Authenticated-EAP-Complete
(EAP_Payload, C/HMAC Digest{EIK})

Figure 8.23. *A single EAP authentication*

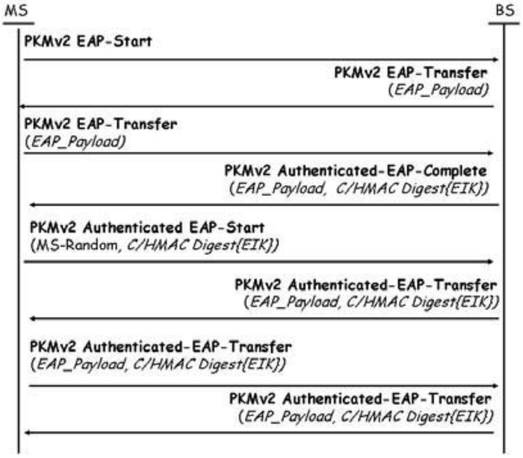

MS
BS
PKMv2 EAP-Start
PKMv2 EAP-Transfer
(EAP_Payload)
PKMv2 EAP-Transfer
(EAP_Payload)
PKMv2 Authenticated-EAP-Complete
(EAP_Payload, C/HMAC Digest{EIK})
PKMv2 Authenticated EAP-Start
(MS-Random, C/HMAC Digest{EIK})
PKMv2 Authenticated-EAP-Transfer
(EAP_Payload, C/HMAC Digest{EIK})
PKMv2 Authenticated-EAP-Transfer
(EAP_Payload, C/HMAC Digest{EIK})
PKMv2 Authenticated-EAP-Transfer
(EAP_Payload, C/HMAC Digest{EIK})

Figure 8.24. *A double EAP session*

The double session is a technique that typically uses a first authentication of the server thanks to the TLS protocol (the standardized version of the well known SSL protocol, the browser yellow padlock). It then takes advantage of the SSL secure channel previously created and performs a second EAP session, protected from hackers by SSL mechanisms, enforcing the privacy and the integrity of the exchanged data.

The first session uses PKMv2 EAP-Transfer messages without HMAC or CMAC. The EAP-Success indication, pointing out the end of this dialogue, is transmitted by the BS thanks to the PKMv2 EAP-Complete message, which includes a signature dealing with the EIK. The mobile checks the validity of this message and calculates a PMK and a new EIK value.

The second session begins with a PKMv2 EAP-Start message signed, thanks to HMAC or CMAC algorithms, by the EIK. Then, the EAP packets are transported in PKMv2 Authenticated-EAP-Transfer messages protected by HMAC or CMAC digests (associated with the EIK).

In the event of the second authentication being successful the MS and the BS generate an AK and then start a SA-TEK 3-way handshake procedure.

After the first occurrence of a double EAP session, the following process, called re-authentication, uses security mechanisms which are somewhat different.

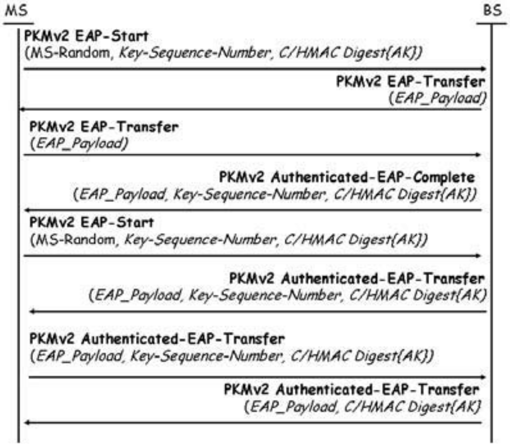

MS
BS
PKMv2 EAP-Start
(MS-Random, Key-Sequence-Number, C/HMAC Digest{AK})
PKMv2 EAP-Transfer
(EAP_Payload)
PKMv2 EAP-Transfer
(EAP_Payload)
PKMv2 Authenticated-EAP-Complete
(EAP_Payload, Key-Sequence-Number, C/HMAC Digest{AK})
PKMv2 EAP-Start
(MS-Random, Key-Sequence-Number, C/HMAC Digest{AK})
PKMv2 Authenticated-EAP-Transfer
(EAP_Payload, Key-Sequence-Number, C/HMAC Digest{AK)
PKMv2 Authenticated-EAP-Transfer
(EAP_Payload, Key-Sequence-Number, C/HMAC Digest{AK})
PKMv2 Authenticated-EAP-Transfer
(EAP_Payload, C/HMAC Digest{AK}

Figure 8.25. *Re-authentication*

A re-authentication dialogue occurs when an AK is already available. It means that the previously double EAP session has successfully ended.

The first session starts with a PKMv2 EAP-Start message (see Figure 8.25) signed by the CMAC_KEY_U or HMAC_KEY_U key, deduced from the AK

(H/CMAC_KEY_U{AK}). Then, the EAP packets are transported by the EAP-Transfer messages. The EAP-Success indication, which marks the end of this session, is transmitted to the BS thanks to PKMv2 EAP-Complete which includes a signature deduced from the AK key (H/CMAC_KEY_U{AK}).

The second session begins with a PKMv2 EAP-Start message signed, thanks to the HMAC or CMAC algorithm, by the H/CMAC_KEY_U{AK} key. Then, the EAP packets are transported by the PKMv2 Authenticated-EAP-Transfer messages provided with HMAC or CMAC digests (linked to the standard AK).

Upon success of the second EAP session, the MS and the BS generate a new AK and start a SA-TEK 3-way handshake procedure.

8.4.4. *SA-TEK 3-way handshake*

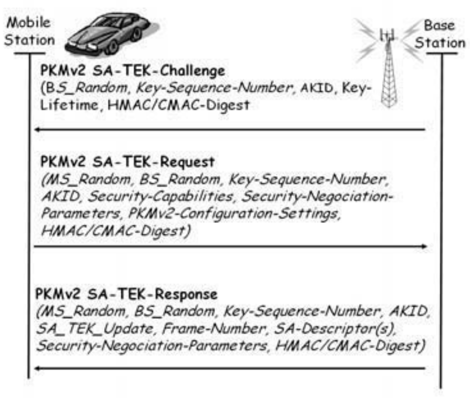

Mobile Station
Base Station
PKMv2 SA-TEK-Challenge
(BS_Random, Key-Sequence-Number, AKID, Key-Lifetime, HMAC/CMAC-Digest
PKMv2 SA-TEK-Request
(MS_Random, BS_Random, Key-Sequence-Number, AKID, Security-Capabilities, Security-Negociation-Parameters, PKMv2-Configuration-Settings, HMAC/CMAC-Digest)
PKMv2 SA-TEK-Response
(MS_Random, BS_Random, Key-Sequence-Number, AKID, SA_TEK_Update, Frame-Number, SA-Descriptor(s), Security-Negociation-Parameters, HMAC/CMAC-Digest)

Figure 8.26. *The SA-TEK 3-way handshake*

This protocol (see Figure 3.26) manages handovers. Before its execution, the MS and the BS must share a common AK and others keys needed for HMAC or CMAC calculations.

In the first instance of the TEK 3-way handshake or during a re-authentication procedure, the BS generates a random number (BS_Random) and transmits a PKMv2 SA-TEK-Challenge, which includes the BS_Random value, an AKID identifier of the session (AK) and a HMAC/CMAC signature.

AKID is the current AK identifier (associated with the Sequence-Key-Number attribute) or the identifier of a new AK instance in the case of re-authentication. The standard details the calculations required to update this value. The keys used by HMAC/CMAC algorithms are deduced from the AK.

The mobile answers this request with a PKMv2 SA-TEK message signed by the keys deduced from the AK, identified by its AKID. This message also includes the list of the security capabilities supported by the subscriber.

The BS delivers the last PKMv2 SA-TEK-Response message which comprises, in the case of a handover, a SA_TEK_Update attribute including, for each security association, the TEK, GKEK and GTEK values, the first and second items being encrypted by KEK and the last one by GKEK.

8.4.5. *TEK distribution procedure*

The TEKs are allocated in a similar way to the IEEE-802.16-2004 standard, and the distribution is based on the PKMv2 Key-Request, PKMv2 Key-Reply and PKMv2 Key-Reject (see Figure 8.27) messages. The TEK-Parameters attribute includes the TEKs and GKEKs encrypted by the KEK.

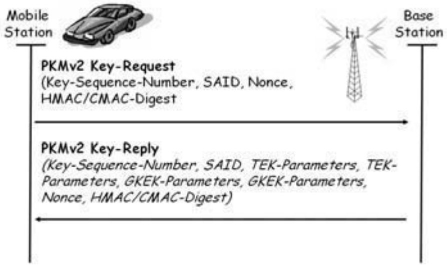

Mobile Station
Base Station
PKMv2 Key-Request
(Key-Sequence-Number, SAID, Nonce, HMAC/CMAC-Digest
PKMv2 Key-Reply
(Key-Sequence-Number, SAID, TEK-Parameters, TEK-Parameters, GKEK-Parameters, GKEK-Parameters, Nonce, HMAC/CMAC-Digest)

Figure 8.27. *TEK distribution*

8.4.6. *(Optional) GTEK updating algorithm*

First, the GTEK is transmitted thanks to the Key-Request and Key-Reply messages. The optional updating is realized by the PKMv2 Group-Key-Update messages (see Figure 8.28), associated with a HMAC or CMAC digest, dealing either with the H/CMAC_KEY_U or H/CMAC_KEY_D keys, according to the value of the Key-Push-Modes attribute.

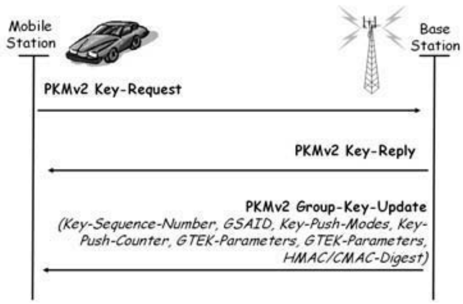

Mobile Station
Base Station
PKMv2 Key-Request
PKMv2 Key-Reply
PKMv2 Group-Key-Update
(Key-Sequence-Number, GSAID, Key-Push-Modes, Key-Push-Counter, GTEK-Parameters, GTEK-Parameters, HMAC/CMAC-Digest)

Figure 8.28. *GTEK updating*

8.4.7. *Security association*

There are three classes of security association linked to several types of connections: unicast, multicast and MBS:

– SA unicast is associated with a unicast connection. It includes the following data: a SAID (16 bits), the KEK computed from AK, two encryption keys TEK_0 and TEK_1 (128 bits) and their lifetime, the packets numbers PN_0 and PN_1 and the reception counters RxPN0 and RxPN1 (of 32 bits) used for the frame encryption;

– SA multicast is associated with a multicast connection (or group security association) and uses the GKEKs and GTEKs;

– SA MBS is associated with a MBS and includes three keys: MAK (160 bits), MGTEK (128 bits) and MTK (128 bits).

8.4.8. *Data encryption algorithms*

The list of data encryption algorithms (and their identifier) is as follows: no encryption (0), DES in CBC mode with a 56 bit key (l), AES in CCM mode with a 128 bit key (2), AES in CBC mode with a 128 bit key (3), AES in CTR mode with a 128 bit key (128).

8.4.9. *Algorithms associated with the TEKs*

The list of the algorithms associated with the TEK key (and the list of their identifier) is as follows: Reserved (0) in EDE mode with a 128 bit key, RSA with a

1,024 bit key (2), AES in ECB mode with a 128 bit key (3), AES in Key-wrap mode, with a 128 bit key (4).

8.4.10. *Summary*

The subscriber obtains two types of CID at the end of the ranging procedure, the *Basic CID* used for the transport of PKM messages and the *Primary CID* associated with messages managing the establishment of connections. A third CID (the *Secondary CID*), delivered by the BS at the end of registration procedure, is used for services such as the allocation of the IP address (DHCP).

At the end of the authorization and key exchange process, the subscriber obtains an AK (and also many signing keys) and its index, the Key-Sequence-Number.

Three types of data streams (unicast, multicast, MBS) are protected thanks to the security associations (identified by the SAID index), which hold the encryption keys and also the cryptographic algorithms used.

These security associations are updated by key distribution procedures authenticated due to C/HMAC keys. These signing keys are identified by parameters such as Basic CID, Key-Sequence-Number and SAID.

8.5. The role of the smart card in WiMAX infrastructures

In GSM and UMTS networks, operators identify their customers and manage their subscriptions using SIM or USIM smart cards. These security modules mostly store a subscriber's identity (the IMSI), and perform user authentication with a symmetric cryptographic algorithm (A3/A8 or milenage), associated with a secret key, which is totally executed in the protected and trusted computing space of the security module.

WiMAX infrastructures and, more particularly, Wi-Mobile use licensed frequency bands. As a result, and unlike the Wi-Fi networks, their deployment will not be free, but will be controlled by operators. More particularly, it should offer a high security level, in order to avoid massive fraud and to guarantee income.

However, no security module is currently defined by the IEEE 802.16 standard. The goal of this section is to introduce some services that could be provided by a smart card, especially dedicated to the IEEE 802.16 standards, in a similar way to the GSM network.

First, the IEEE 802.16 standards use on the subscriber's side a X.509 certificate and a private RSA key. An AK or Pre-PAK is generated by the BS and encrypted with the subscriber's public key.

As a result, an IEEE 802.16-2004 smart card may, for example, store the subscriber's X.509 certificate and realize the decryption of AKs (802.16-2004) in a trusted computing device (see Figure 8.29). The AK is never exported from the card, which calculates the values KEK, HMAC_KEY_D, HMAC_KEY_U, and HMAC_KEY_S.

Briefly, the card manages the authorization security association. We also point out that the subscriber's identity is clearly shown by a X.509 certificate.

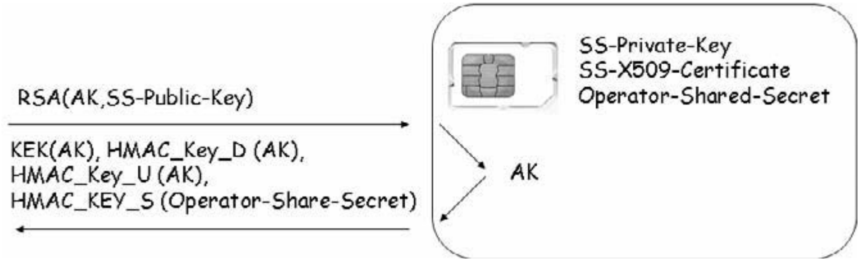


Figure 8.29. *Services protected by a smart card for IEEE 802.16-2004*

The smart card design for IEEE 802.16e is more complex. At first, the subscriber identification and authorization procedure is based on a X.509 certificate and/or on other parameters associated with the EAP, for example, an IMSI (imported from GSM) used by the EAP-SIM protocol [IET 06].

In the case of the PKMv2-RSA protocol (see Figure 8.30), the card decrypts the Pre-PAK key, encrypted by the BS as a result of the subscriber's public key. It also calculates the SigSS signatures inserted in the PKMv2-RSA-Request and PKMv2-RSA-Acknowledgment messages (see section 8.4.2). It calculates the EIK_{RSA} and PAK parameters from the pre-PAK value. The EIK_{RSA} key is exported, in order to enable the terminal to check and calculate HMAC and CMAC digests dealing with keys deduced from the EIK.

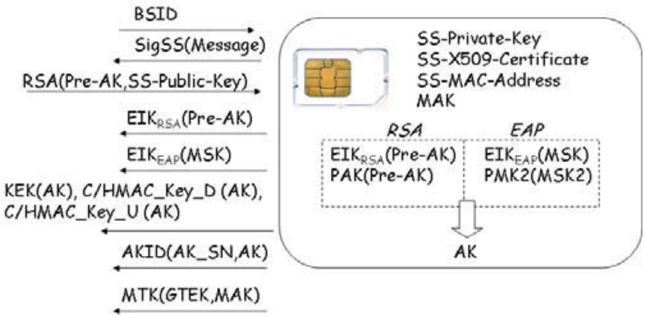


Figure 8.30. *Services protected by smart card for IEEE 802.16e*

When the EAP protocol is used with a single or double session, the smart card manages the EAP messages in an autonomous way. More information on this technology, usually called the EAP card (see Figure 8.31), can be found in [LCN 03] and [PUJ 04].

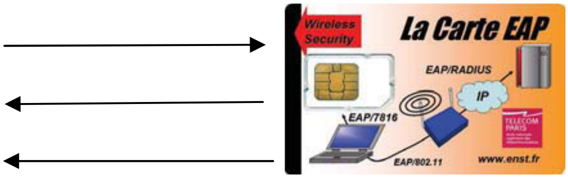


Figure 8.31. *The EAP card*

The IEEE 802.16e smart card produces, at the end of the first session, a pair of EIK_{EAP} and PMK keys. At the end of an optional second instance, it calculates a PKMv2 value.

The EIK_{EAP} is exported in order to enable the terminal to check and calculate HMAC or CMAC digest values, dealing with keys deduced from the EIK. Using secret values PAK, PMK and PMK2, the smart card calculates and exports five values H/CMAC_KEY_U, H/CMAC_KEY_D, and KEK deduced from the AK. It calculates and exports AKID values, thanks to the AK, which is securely stored. The MTK (deduced from the MAK secret value) is calculated according to the current GTEK value.

8.6. Conclusion

In this chapter, we attempt to present a clear explanation of the security mechanisms defined for emerging WiMAX networks. We notice the existence of many options that are based on PKI infrastructures or, on the contrary, that deal with architectures using symmetric cryptographic keys similar to the actual GSM or UMTS cellular networks.

The deployment of WiMAX will probably establish a de facto standard. However, it is likely that security modules, and smart cards more precisely, will be required in order to guarantee trusted and profitable services for this new wireless network generation.

8.7. Glossary

3-DES	Triple Digital Encryption Standard
AES	Advanced Encryption Standard
AK	Authorization Key
AKID	AK IDentifier
ASA	Authentication and Service Authorization
BS	BS
BSID	BS IDentification
CBC	Cipher Block Chaining
CBC-MAC	Cipher Block Chaining Message Authentication Code
CCM	CTR mode with CBC-MAC
CID	Connection IDentifier
CMAC	Cipher-based Message Authentication Code
CPS	Common Part Sub-layer
CS	Convergence Sub-layer
CTR	CounTeR mode encryption

DCD	Downlink Channel Descriptor
DES	Digital Encryption Standard
DL-MAP	Downlink Map
EAP	Extensible Authentication Protocol
ECB	Electronic Code Book
EDE	Encrypt Decrypt Encrypt
EIK	EAP Integrity Key
FDD	Frequency Division Duplexing
FFT	Fast Fourier Transform
GKEK	Group Key Encryption Key
GMH	Generic MAC Header
GTEK	Group Traffic Encryption Key
HMAC	Hashed Message Authentication Code
HO	HandOver
KEK	Key Encryption Key
LOS	Line Of Sight
MAC	Medium Access Control
MAK	MBS AK
MBS	Multicast and Broadcast Services
MIMO	Multiple Input, Multiple Output
MS	Mobile Station
MSK	Master Session Key
MTK	MBS Transport Key
NLOS	Non-Line Of Sight
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiplexing Access
PAK	Primary Authorization Key

PHY	PHYsical layer
PKM	Privacy Key Management
PMD	Physical Medium Dependant
PMK	Pairwise Master Key
Pre-PAK	Pre-Primary AK
PS	Privacy Sub-layer
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
SAID	Security Association Identifier
SC	Single Carrier
SFID	Service Flow IDentifier
SN	Sequence Number
SOFDMA	Scalable Orthogonal Frequency Division Multiplexing Access
SS	Subscriber Station
TDD	Time Division Duplexing
TEK	Traffic Encryption Key
TLV	Type Length Value
UL-MAP	Uplink Map
UCD	Uplink Channel Descriptor

8.8. Bibliography

[DOC 05] Data-Over-Cable Service Interface Specifications, *Baseline Privacy Plus Interface Specification*, DOCSIS 1.1, 2005.

[IEE 01] Institute of Electrical and Electronics Engineers, *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access System*, IEEE 802.16 Std 2001.

[IEE 04] Institute of Electrical and Electronics Engineers, *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access System*, IEEE 802.16 Std 2004.

[IEE 04a] Institute of Electrical and Electronics Engineers, *IEEE Standard for Local and Metropolitan Area Networks Part 11: Port-Based Network Access Control*, IEEE 802.1x Std 2004

[IEE 06] Institute of Electrical and Electronics Engineers, *IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operation in Licensed Bands and Corrigendum 1*, 2006.

[IET 83] Internet Engineering Task Force, IETF, *RFC 868, Time Protocol*, May, 1983.

[IET 89] The Internet Engineering Task Force, IETF, *RFC 1123, Requirements for Internet Hosts – Application and Support,* October 1989.

[IET 97] Internet Engineering Task Force, IETF, *RFC 2104, HMAC: Keyed-Hashing for Message Authentication,* February 1997.

[IET 97a] Internet Engineering Task Force, IETF, *RFC 2131, Dynamic Host Configuration Protocol*, March 1997.

[IET 98] Internet Engineering Task Force, IETF, *RFC 2349, TFTP Timeout Interval and Transfer Size Options*, May 1993.

[IET 99] Internet Engineering Task Force, IETF, *RFC 2459, Internet X.509 Public Key Infrastructure Certificate and CRL Profile Time Protocol*, January 1999.

[IET 99a] Internet Engineering Task Force, IETF, *RFC 2716, PPP EAP TLS Authentication Protocol*, October 1999.

[IET 04] Internet Engineering Task Force, IETF, *RC 3748, Extensible Authentication Protocol (EAP)*, March 2004.

[IET 06] Internet Engineering Task Force, IETF, *RFC 4186, Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM),* January 2006.

[IET 06b] Internet Engineering Task Force, IETF, *RFC 4187, Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)*, January 2006.

[IET 06c] Internet Engineering Task Force, IETF, *RFC 4493, The AES-CMAC Algorithm,* June 2006.

[LCN 03] Urien P., “Les réseaux sans fil 802.11”, in Michel Riguidel (Ed.) *La sécurité à l’ère du numérique*, Les cahiers du numérique vol. 4 – no. 3-4/2003, Hermes, 2003.

[NIS 05] NIST, Special Publication 800-38B Draft, *Recommendation for Block Cipher Modes of Operation: The CMAC Method for Authentication*, March, 2005.

[PKCS 02] RSA Laboratories, *PKCS #1*, *RSA Cryptography Standard*, June 2002.

[PUJ 04] Pujolle G., Loutrel M., Urien P., Borras P., Plateau D., *Sécurité Wi-Fi*, Eyrolles, 2004.

Chapter 9

Security in Mobile Telecommunication Networks

9.1. Introduction

Circuit-switched telecommunication networks were created at a time when there was a strong monopoly granted to government-owned corporations. Depending on the nation, the network operator was either a government-controlled company under government monopoly or a private company under a government-granted monopoly. The principal objective was to guarantee the fulfillment of public service duties, i.e., the establishment of telecommunications over a national territory. The access to such a network structure was granted with an analog landline for which the user identifier, his localization and his billing address were identical. Transnational internetworking required for worldwide telecommunications was based on mutual agreements based on an operator's reputation. The first generation of cellular networks largely followed such a principle.

At the eve of digital cellular networks such as GSM and with the potential security pitfall being located at the radio access, telecommunication companies therefore required formal user authentication. However, no authentication provision from telecommunication operators with respect to subscribers was judged necessary as they were unique operators in their own national territories and the cost of the equipment required to masquerade as a national telecommunication operator was considered to be prohibitive.

Chapter written by Jérôme HÄRRI and Christian BONNET.

A major drift in the vision of the security of telecommunication networks occurred with telecommunication deregulation laws. Such laws indeed allowed, and in special cases even forced (anti-trust legislation in the USA), the emergence of the Competitive Local Exchange Carrier (CLERC) or Mobile Virtual Network Operator (MVNO) for the first time renting the infrastructures of the national or local operator. Such a new competitive environment also triggered new challenges: with new offsprings in the telecommunications arena, reputation-based securitization was no longer sufficient. Moreover, in the case of virtual operators renting other operators' infrastructures, billing also became crucial.

Signaling System #7 (SS7), also referred to as Common Channel Signaling System 7 (CCSS7) in North America, is the worldwide signaling system used by most of the public switched telephone network (PSTN). SS7 is packet-switched and physically separated from the PSTN in order to increase the circuit switching time and to avoid fraud. Thanks to the monopoly position held by national telecommunication operators, their tightly controlled access to the SS7 was their only security measure to fight fraud, and for many years such a measure was considered to be totally sufficient. However, with the deregulation and the interconnection of telecommunication networks, such centralized access control was no longer acceptable.

With deregulation, telecommunication operators also started proposing new value-added services in order to distinguish themselves from their competitors. These kinds of new service were initially based on the concept of *Intelligent Network* (IN) platforms able to manage the interoperability and heterogenity of access technologies. Some particular value-added services (telephone number portability, toll-free calls, prepaid calling) triggered a debate on the securitization of data transported by such INs.

A further evolution occurred with the interconnection of telecommunication networks and the Internet and its multiple consequences on CLERCs or MVNOs. One of the most prominent evolutions was the landline and therefore SS7 local loop unbundling (LLU) for the access to telecommunication networks. Such action allowed cellular operators to offer mobile access to the Internet and conversely Internet providers to offer services like sending text messages (SMS) to cell phones. New access techniques from the Internet also managed to mitigate known security loopholes in telecommunication networks.

Since then, a profitable mutual collaboration has appeared between landline, mobile and Internet Service Providers, the former two benefiting from a higher communication throughput with the latter taking advantage of a local connection to its services. The Internet service that directly competed with landline and mobile operators was the transmission of voice packets, also called Voice-over-IP (VoIP).

This new application and many multimedia-related applications will be discussed in detail in Chapter 11.

In this chapter, our goal is to illustrate the security mechanisms employed in mobile telecommunication networks and to emphasize the potential security breaches and related solutions guaranteeing operators and customers a secured and efficient data and infrastructure management. However, our objective is not to provide an exhaustive list of all possible known or probable security attacks but instead to address various attack classes and methods that could illegally exploit security breaches of mobile telecommunication networks.

9.2. Signaling

Signaling in telecommunication networks has always been a problem for smooth network functionality. It is through robust signaling that calls are correctly routed to the correct destination and are specific subscribed services are billed to the correct person. The corollary is that signaling has already, since its beginnings, been a target for acts of sabotages from groups of persons aiming to illegally benefit from a telecommunications network or more dramatically seeking to hijack or to totally shatter it. It is therefore crucial to develop robust signaling protocols and if need be to identify its flaws and correct them.

In autumn 1997, the complete telecommunication network of the island of Puerto Rico was sabotaged by its wires being physically cut. With the convergence of the information and communication worlds, particularly with the interconnection of various telecommunication networks, it is now possible to conduct a similar attack by remotely tampering with the signaling network. We are going to identify and describe in this section different opportunistic attacks benefiting from various security breaches in the protocol stack of Signaling System #7.

Even though signaling issues are not specific to mobile telecommunication networks, SS7 or any other more complex signaling system being the kingpin for correct and efficient network functionality, securing signaling systems is a critical aspect in the security of mobile telecommunication networks.

9.2.1. *Signaling System 7 (SS7)*

The major signaling protocol used in telecommunication networks is the SS7 also referred to as *Common Channel Signaling System 7* (CCSS7) in North America. It is used in public switched telephone networks, cellular networks and even in their interconnection with IP networks. SS7 along with the functioning of its protocol stack is available worldwide and thus is subject to a large community of potential attackers targeting its security breaches. SS7 allows core network devices such as switches or customer databases to speak to each other.

The SS7 architecture consists of an overlay broadband packet-switched network managing the signaling of a physically separated underlay data or voice network. SS7 may therefore be used independently by packet- or circuit-switched networks. As illustrated by Figure 9.1, this network is composed of an interconnection of three categories of signaling nodes, commonly called SS7 nodes:

– *Service Switching Point (SSP)*: this is a telephone exchange that is either origin or destination of a call and is the first equipment to respond to a dialed number. An SSP sends signaling in order to configure, manage and release the network circuits required for a call. It may also send a request to a SCP in order to obtain information related to an incoming call.

– *Service Control Point (SCP)*: this is a standard component used to control the offered value-added telephone services (toll-free, prepaid calls, roaming) by communicating with a Service Data Point (SDP) holding the required billing, databases and telephone directories. A SCP and the related SDP represent the intelligence behind a particular service and are at the core of the IN approach.

– *Signal Transfer Point (STP)*: this is an intermediate router relaying SS7 signaling messages to a Signaling End-Point (SEP) such as a SSP or a SCP based on the routing information contained in a SS7 message frame. A STP also acts as a firewall and monitors SS7 messages received from external telecommunication networks.

Each nation has a specific addressing plan for its SS7 nodes. Worldwide interconnection is based on particular SS7 nodes that also possess an international address and therefore act as gateways.

The SS7 network being critical to the correct functioning of call management, SCPs and STPs are deployed in pairs in physically different locations in order to mitigate the consequences of component failures. Resilience regarding attacks or failures are based on the confidentiality of the detailed network structure and not on security measures on the SS7 protocol stack itself.

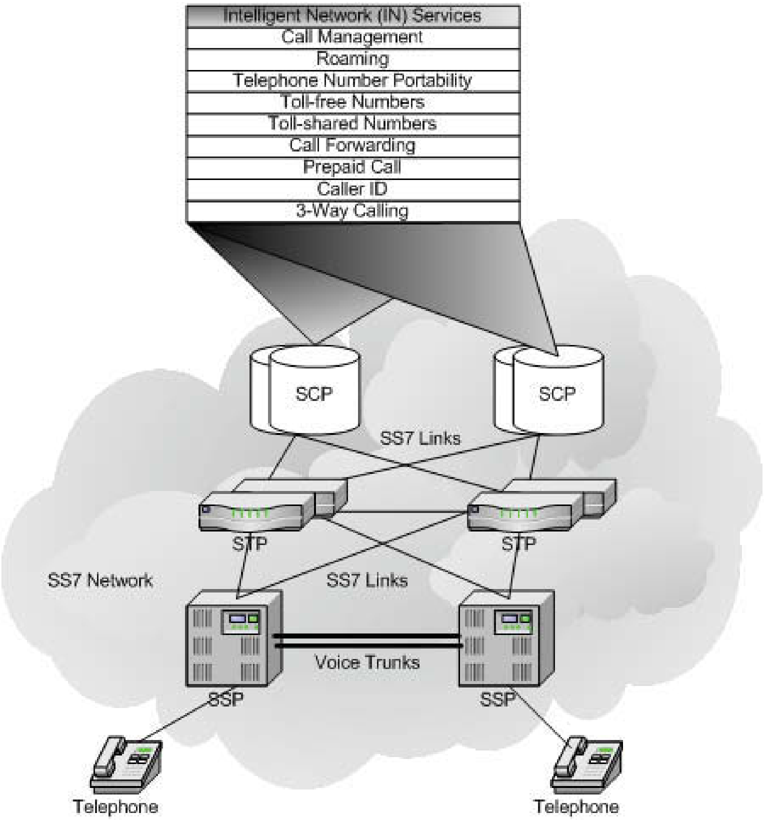


Figure 9.1. *SS7 architecture*

SS7 relates to signaling exchanges between the core components of a telecommunication network. Signaling between a subscriber terminal and the SSP, referred to as the subscriber line, is based on a different access signaling specification. In the case of a digital terminal (*ISDN – Integrated Services Digital Network*), a specific protocol for communication establishment is triggered with the SSP. It is actually on this principle that the GSM network built its access protocol between a mobile terminal and the infrastructure network.

The relative resilience of SS7 with respect to possible security breaches was strongly correlated to the former low external penetration of proprietary telecommunication networks. However, with deregulation, new emerging players entered the market, such as alternative or mobile virtual telecommunication operators. For a modest fee, the regulation authority authorized them to connect to a SS7 network or to interconnect their small SS7 network with a larger one in order to increase the diversity of their offered services. As illustrated in Figure 9.2, what was earlier a weakly interconnected SS7 cloud now became a strongly interconnected and interlinked network of SS7 networks. The bottom line is that SS7 networks are no longer trustworthy without any additional security measures. Access control to SS7 networks therefore rose to high priority in order to avoid fraudulent access or reconfiguration of SS7 nodes.

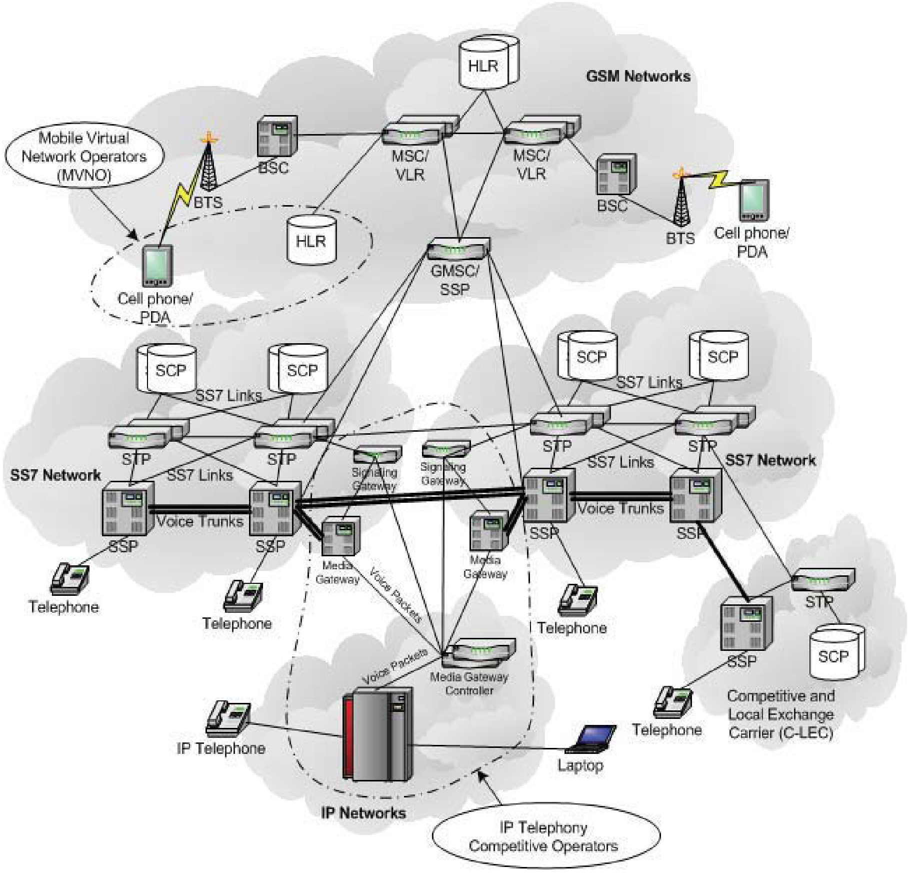


Figure 9.2. *Interconnection of telecommunication networks*

9.2.2. *SS7 protocol stack*

The SS7 protocol stack is composed of four layers. The first three are in charge of establishing point-to-point transfers while the fourth represents the application part of SS7. Figure 9.3 illustrates this 4-layer protocol stack.

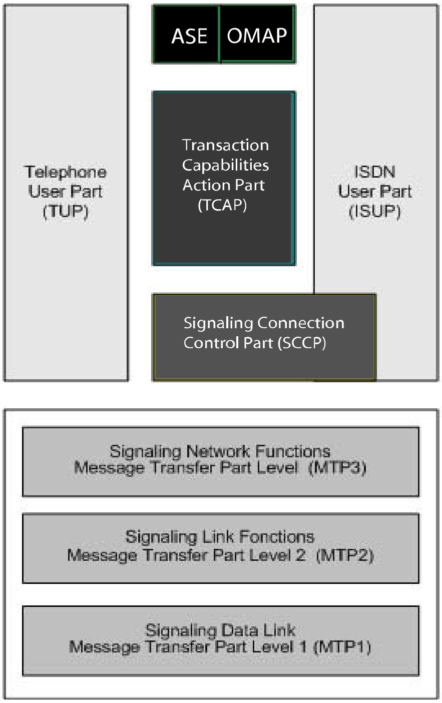

ASE
OMAP
Transaction Capabilities Action Part (TCAP)
Telephone User Part (TUP)
ISDN User Part (ISUP)
Signaling Connection Control Part (SCCP)
Signaling Network Functions Message Transfer Part Level (MTP3)
Signaling Link Fonctions Message Transfer Part Level 2 (MTP2)
Signaling Data Link Message Transfer Part Level 1 (MTP1)

Figure 9.3. *SS7 protocol stack*

The SS7 protocol stack consists of seven main functionalities:

– *Signaling Data Link (MTP1)*: this is the SS7 physical layer responsible of the interconnection of SSPs.

– *Signaling Link Functions (MTP2)*: this is the link layer managing reliable transmissions (error detection, sequence checking).

– *Signaling Network Functions (MTP3)*: this layer routes signaling messages point-to-point through the SS7 network to a requested endpoint. It is also responsible for network management as it is in charge of homogenous traffic allocations or link redirections if the availability of a MTP2 datalink changes.

– *Signaling Connection Control Part* (*SCCP*): this extends the MTP layer by including advanced facilities such as global title translations (toll-free numbers or calling numbers for prepaid cards) into SSP addresses and guarantees the transport of connectionless or connection-oriented services. Unlike MTP, SCCP establishes end-to-end connections.

– *Transaction Capabilities Application Part (TCAP*): this enables multiple and concurrent data exchanges between various applications through SS7 using the connectionless version of SCCP. Transactions between SCP and STP are based on TCAP. In GSM networks, the MAP messages that are exchanged between infrastructures and databases are also routed with TCAP.

– *Telephone User Part (TUP)*: this defines international signaling functions in order to establish communication. It does not allow the establishment of data links.

– *ISDN User Part (ISUP)*: this configures, manages and releases voice or data circuits between SSPs. Despite its confusing name, ISUP is now used for ISDN and non-ISDN calls.

9.2.3. *Vulnerability of SS7 networks*

As previously mentioned, the largest vulnerability of SS7 networks is the lack of any form of access control. Anybody who is able to generate SS7 messages and inject them into a SS7 network may potentially disturb the operator's services. For example, the perturbation of an operator forwarding call service could create chaos in the operator call routing. In Figure 9.4 we illustrate three possible potential security intrusions in a SS7 network.

First, users owning ISDN telephones may introduce messages in SS7 networks via the user access interface. For example, *Attack #1* may usurp the identity of the source telephone and then feed the network with malicious packets.

A second vulnerability comes from the convergence of telecommunication networks and the Internet. Internet Service Providers (ISP) or Data Local Exchange Carriers (DLEC) commonly rent chunks of landline telecommunication networks and conversely SS7 networks are interconnected thanks to the Internet. As a direct consequence, SS7 networks are vulnerable to Internet security breaches and consequently may in turn disturb the Internet. It becomes possible for *Attack #2* to break into a SS7 network and any other interconnected SS7-based networks.

Finally, alternative or virtual network operators may be less protected than proprietary telecommunication networks and could therefore be the source of a malicious intrusion if their networks are compromised (*Attack #3*).

It can also be mentioned that the access of a SS7 network by GSM or GPRS mobile networks operators, virtual or not, is considered to be secure. However, deregulation in mobile telecommunication also brought its share of vulnerabilities, leading to more potential security breaches.

Another significant breach comes from telephone number portability between operators. This service authorizes customers to change operators while still keeping their original calling number, a service typically handled by accessing a SCP node. Unfortunately, once a SCP (the brain of a SS7 network) or more generally any SS7 component has been corrupted by a malicious customer or network, it becomes virtually impossible to counter its attacks.

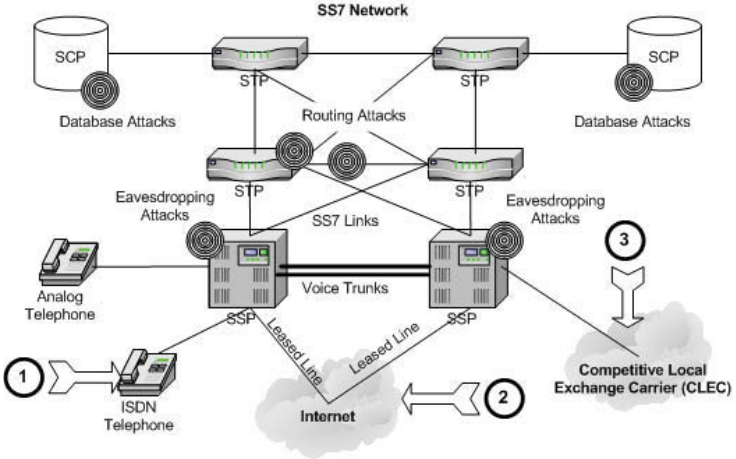


Figure 9.4. *Examples of potential SS7 attacks*

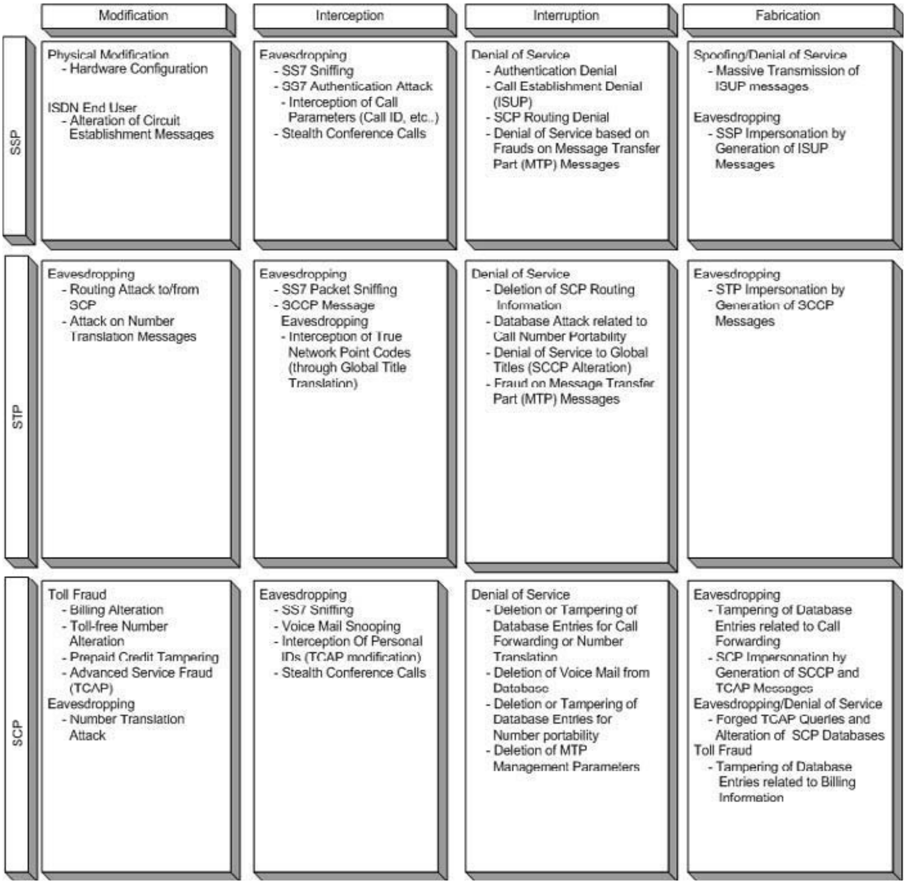
9.2.4. *Possible attacks on SS7 networks*

Security flaws addressed in the previous section are the source of a plethora of possible attacks classified as tampering, interception, disruption and deception. Their targets are all three SS7 node types, as illustrated in Figure 9.4:

– *SSP attacks*: SSP nodes belong to the edge of the SS7 network and may therefore act as attack gateways. They are particularly subject to eavesdropping of voice communication or sniffing of data transmissions as a user's complete traffic is relayed by its dedicated SSP. Other types of attacks simply benefit from the lack of access control on the ISDN user interface, as the ISDN explicitly allows direct access to the SS7 network by a simple digital telephone.

– *STP attacks*: STP nodes being the routing core of any SS7-based telecommunication network, attacks disrupting a correct delivery of signaling packets may potentially be very harmful. For example, a SSP is at least directly connected to one STP and compromising it would allow the monitoring of traffic from and to this SSP. It is also possible to remotely compromise a STP to create a transparent bogus STP to filter traffic, making it possible to eavesdrop on selected conversations.

– *SCP attacks*: SCP nodes store sensitive information for the signaling plane and are therefore critical. As an example, let us consider toll-free numbers. Such a number typically hides the physical number effectively dialed to the user and the billing information is maintained separately. However, the real number is used at the signaling phase for a correct call establishment. If a malicious user manages to access a SCP and replace it with another number, it will be possible for him to call for free. Similar tampering attacks are also possible with billing data, number portability or information related to a user's phone line. It is therefore possible to steal a customer's identity and call on the spoofed customer account.



	Modification	Interception	Interruption	Fabrication
SSP	Physical Modification - Hardware Configuration ISDN End User - Alteration of Circuit Establishment Messages	Eavesdropping - SS7 Sniffing - SS7 Authentication Attack - Interception of Call Parameters (Call ID, etc..) - Stealth Conference Calls	Denial of Service - Authentication Denial - Call Establishment Denial (ISUP) - SCP Routing Denial - Denial of Service based on Frauds on Message Transfer Part (MTP) Messages	Spoofing/Denial of Service - Massive Transmission of ISUP messages Eavesdropping - SSP Impersonation by Generation of ISUP Messages
STP	Eavesdropping - Routing Attack to/from SCP - Attack on Number Translation Messages	Eavesdropping - SS7 Packet Sniffing - SCCP Message Eavesdropping - Interception of True Network Point Codes (through Global Title Translation)	Denial of Service - Deletion of SCP Routing Information - Database Attack related to Call Number Portability - Denial of Service to Global Titles (SCCP Alteration) - Fraud on Message Transfer Part (MTP) Messages	Eavesdropping - STP Impersonation by Generation of SCCP Messages
SCP	Toll Fraud - Billing Alteration - Toll-free Number Alteration - Prepaid Credit Tampering - Advanced Service Fraud (TCAP) Eavesdropping - Number Translation Attack	Eavesdropping - SS7 Sniffing - Voice Mail Snooping - Interception Of Personal IDs (TCAP modification) - Stealth Conference Calls	Denial of Service - Deletion or Tampering of Database Entries for Call Forwarding or Number Translation - Deletion of Voice Mail from Database - Deletion or Tampering of Database Entries for Number portability - Deletion of MTP Management Parameters	Eavesdropping - Tampering of Database Entries related to Call Forwarding - SCP Impersonation by Generation of SCCP and TCAP Messages Eavesdropping/Denial of Service - Forged TCAP Queries and Alteration of SCP Databases Toll Fraud - Tampering of Database Entries related to Billing Information

Figure 9.5. *Taxonomy of SS7 attacks*

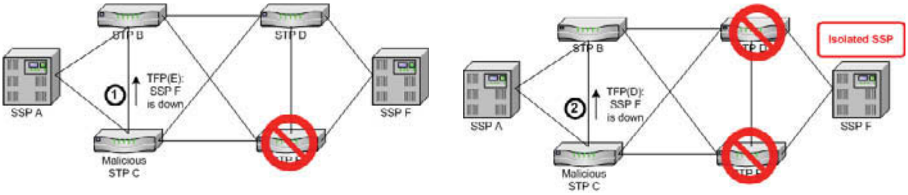
9.2.5. *Securing SS7*

In order to attempt to secure SS7, network operators have to control access to the SS7 network as well as the behavior of the respective SS7 nodes. As SS7 has not been designed with malicious actions in mind, it is therefore very hard (perhaps impossible) to totally protect SS7 against any attack forms. An intermediate solution is to work on mitigating attacks with a strong disruption potential.

Among the various possible attacks listed in Figure 9.5, those on the MTP layer form the minority. However, they have a capacity to totally block a SS7 network. Since the initial reports illustrating these problems, operators and equipment providers worked on patches. Telcordia's *Gateway Screening* [TEL 01] checks MTP3 message headers. This system is able to more thoroughly verify SS7 signaling messages and ensure that the origin and destination point codes are legitimate. However, this approach is not able to control the message content situated higher on the SS7 protocol stack. Tekelec's *EAGLE STP Gateway Screening (GWS)* [TEK 01] is similar to Telcordia's but provides a control on the message content at the MTP and SCCP layers. Another more ambitious solution is Verizon's *SS7 Security Gateway Keeper* [VER 02] which also includes a proper sequencing, syntax and content screening.

Principally, all proposed solutions are nothing more than firewalls for SS7 networks, suffering from the very well-known limitations behind this type of security. The lack of authentication and integrity checks in SS7 networks remains a major factor behind malicious attacks such as the isolation of an access point or even of an entire network, or the re-routing of legitimate signaling packets to malicious nodes. We illustrate in Figure 9.6 a two phase attack example initially diverting traffic and then isolating a SSP. This attack benefits from the lack of any integrity check in TFP (TransFer Prohibited) messages that are employed to notify network equipment when a particular link has failed. If an intruder impersonates a STP (in our example, the STP D and E), it manages to send forged FTP messages to disturb the network.

A solution called MTPSec [SEN 05] has been proposed to establish link-by-link MTP secure channels, and possibly also securing the interconnection links between two SS7 networks. Inspired from the strength of IPsec, MTPSec proposes to secure communications on MTP links by authenticating the originator and checking the integrity of MTP messages to reduce potential routing disruptions in SS7 networks. Augmented by a firewall, this approach could not only reduce malicious access but also malicious behavior.



SS7 routing alteration

Isolation of a SS7 node

Figure 9.6. *Attacks on SS7 routing*

Other proprietary or academic solutions have been developed to detect attacks on SS7 networks or to protect against intrusions [LOR 01]. In particular, it has been notably proposed to create an application-layer control system called a *Security Application Part (SecAP)* [SEI 98] which would coordinate various distributed control platforms in a SS7 network such as a firewall on all STPs or efficient access control on each SSP and SCP.

9.3. Security in the GSM

In 1982, the working group called in French “Groupe Spécial Mobile” (GSM) was created by the *European Conference of Postal and Telecommunications Administrations (CEPT)*. Its objective was the creation of a digital standard for 2G mobile telecommunication. This standard was been developed by the European Telecommunications Standards Institute (ETSI) in the 900 MHz and 1,800 MHz frequency bands. The GSM standard, later renamed in English as Global System for Mobile communications (GSM), benefited from worldwide success which allowed it to be used across Europe, Africa, the Middle East and Asia. Having not initially believed in this system, North American countries adapted it much later in the 850 MHz and 1,900 MHz frequency bands as the standard frequency bands were already allocated. 20 years after its creation, the GSM technology covers 100% of world nations and exceeds 2 billion users (including the recent extensions GPRS and UMTS, resp. 2.5G and 3G). The GSM network has been specifically created for voice communications and similarly to PSTN it is circuit-switched oriented. In order to join a GSM network, a potential customer has the choice of taking a subscription or buying a prepaid GSM calling card.

9.3.1. *GSM architecture*

A Public Land Mobile Network (PLMN) is a wireless communication system providing telecommunication services to mobile subscribers. The GSM network is the most popular example of a PLMN and each non-virtual GSM operator owns one. With deregulation, an increasing number of virtual operators leasing the structures of a PLMN appeared. In order to also offer communication capabilities outside a PLMN, the GSM is also connected to at least one PSTN. For more information on the GSM network, see [GSM 01].

As illustrated in Figure 9.7, a GSM-PLMN network is composed of 4 major entities:

– The *Mobile Station (MS)* is usually a mobile telephone but more generally any device equipped with an adequate antenna and a SIM (*Subscriber Identity Module*) card may be considered to be a mobile station.

– The *Base Station Subsystem (BSS)* is composed of a network of radio relays called *Base Transceiver Stations (BTSs)* and of concentrators called *Base Station Controllers (BSCs).* BSCs are usually considered as the intelligence behind BTSs and typically have 10 s to 100 s of BTS under their control. BTSs transmit and receive signals from and to mobile stations and are the only radio interface of the total GSM system, as any other communication form between BTS and BSC as well as with the Core Network (CN) is performed by digital landline communication based on SS7 signaling.

– The *Network Switching Subsystem (NSS)* is in charge of a correct routing of voice calls between two GSM subscribers or to a PSTN. It is composed of specialized interlined switches called *Mobile Switching Centers (MSCs)* that have several BSCs under their responsibility. Each MSC is associated with a specialized database called a *Visitor Location Register (VLR)* managing subscribers' information that is within the radio range covered by the MSC (or more precisely the summed radio coverage of all BTSs of the MSCs and BSCs). In order to manage subscribers' information for the complete PLMN, a unique database called the *Home Location Register (HLR)* also exists even though copies are appropriately located in the PLMN to mitigate the risks of failure. The HLR also contains the *Authentication Center (AuC)* which is in charge of the proper subscriber identification. The complementary information to the AuC is located into a subscriber's SIM card. Local copies of the required information from the HLR are transferred to the respective VLRs in order to speed up the handling time of MSC requests. Finally, a specific MSC called a *Gateway MSC (GMSC)* is located at the entry point of a PLMN and therefore acts as a gateway with PSTNs.

– The *Operation and Maintenance Center (OMC)* is in charge of monitoring the correct functionality of all GSM systems and taking appropriate maintenance actions when necessary.

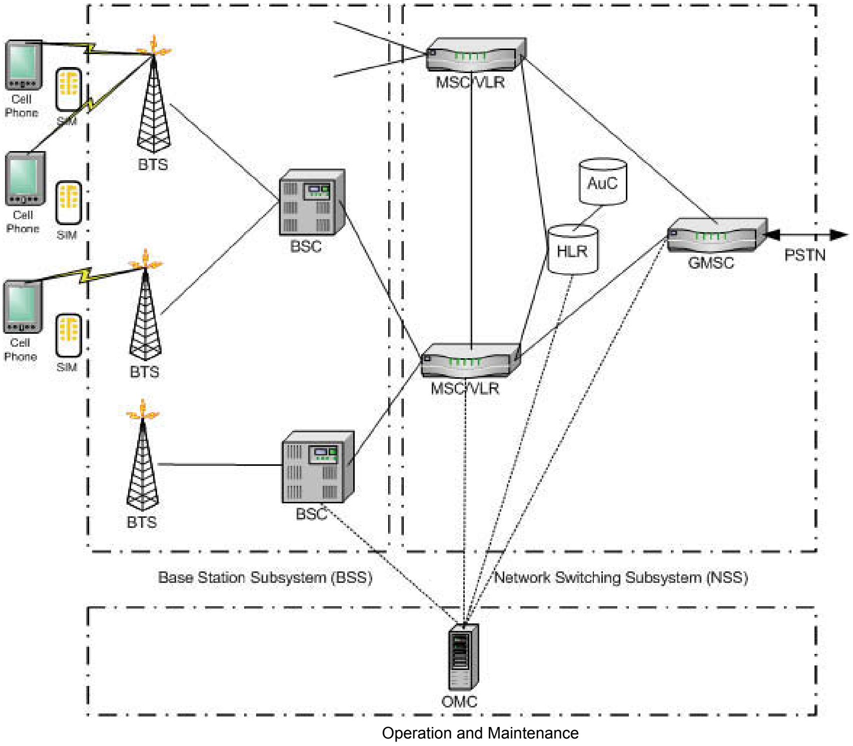

Cell Phone
SIM
BTS
Cell Phone
SIM
Cell Phone
SIM
BTS
BTS
BSC
BSC
MSC/VLR
MSC/VLR
AuC
HLR
GMSC
PSTN
Base Station Subsystem (BSS)
Network Switching Subsystem (NSS)
OMC
Operation and Maintenance

Figure 9.7. *GSM architecture*

9.3.1.1. *GSM and the SS7*

As previously mentioned, signaling and circuit switching in the NSS is based on SS7. However, GSM having specific needs, particularly in terms of roaming capabilities and access to IN platforms, two new protocols have been created for a smoother interaction between SS7 and GSM. These are illustrated in Figure 9.8:

– *Mobile Application Part (MAP)*: this is a protocol that provides application-layer communication capabilities between various NSS nodes. It is notably in charge of SMS (Text Messages) routing or of mobility and roaming management. It therefore needs to exchange information between VLR/HLRs belonging to different mobile communication networks (proprietary or virtual).

– *Customized Applications for Mobile network Enhanced Logic (CAMEL)*: this allows the interconnection of mobile networks with IN platforms and guarantees subscribers access to services such as Prepaid Call, Call Number Portability or

Location-based Applications. The specific protocol used is the CAMEL Application Part (CAP).

These two protocols (MAP/CAP) are actually also part of the GPRS, UMTS and IMS networks.

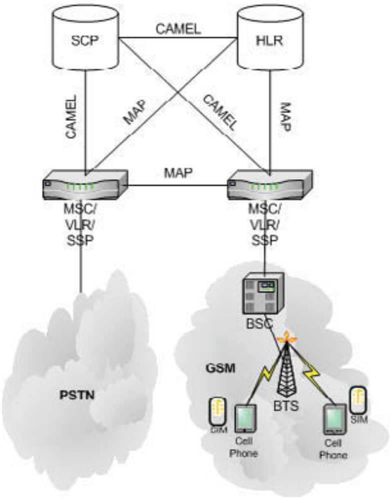


Figure 9.8. *MAC/CAP architecture*

9.3.2. *Security mechanisms in GSM*

Most security protections provided by the GSM are located at the BSS and limited to access control and radio encryption. The emergence of application-layer security mechanisms securing the messages exchanged between SIM cards and an application server is more recent. As far as the GSM is concerned, however, NSS is considered safe. In this section, we will therefore limit our investigations to radio link and access control security.

GSM security is composed of three classes of protection:

– *subscriber identity protection*. For privacy issues, transmitting a subscriber identity in plain on a radio link must be avoided;

– *network access control* by means of SIM cards. The major functionality of the SIM is to securely hold and manage confidential information to allow the GSM network to formally identify a subscriber's identity. Figure 9.9 illustrates the information contained in such a SIM and that is required for a proper operation of a GSM network;

– *radio communication encryption* between a MN and the BTS. Eavesdropping on radio communication being significantly easier than landline communication, it is absolutely vital to protect the radio link.

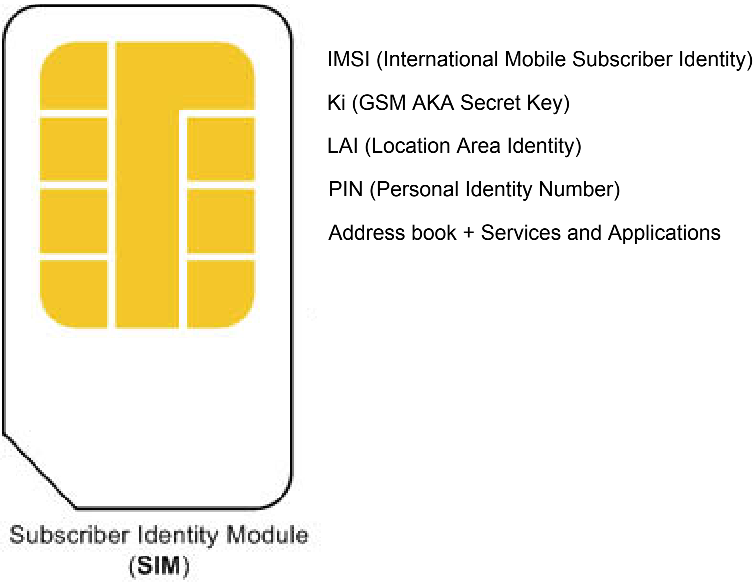

IMSI (International Mobile Subscriber Identity)
Ki (GSM AKA Secret Key)
LAI (Location Area Identity)
PIN (Personal Identity Number)
Address book + Services and Applications
Subscriber Identity Module
(SIM)

Figure 9.9. *Data contained in a SIM card*

9.3.2.1. *Subscriber identity protection*

The objective of this function is to avoid revealing which subscriber uses which network resource by simply eavesdropping on signaling on the radio link. A primary objective is to ensure data and signaling confidentiality, but a secondary objective is also to prevent the localization and the tracking of a specific MN. In concrete terms, this means the *International Mobile Subscriber Number (IMSI)* contained in the SIM card and in the HLR shall never be transmitted in plain text.

Instead, the system uses a temporary subscriber number (TMSI) on the radio link. The TMSI only holds temporary and local validity, meaning that only the fusion between the TMSI and the Local Area Identifier (LAI) may reveal the IMSI. The association between the IMSI and the TMSI is kept safe by the VLR that is accordingly in charge of creating a new TMSI when roaming outside a local area.

Figure 9.10 illustrates the exchange mechanism in order for a MN to be able to obtain a new TMSI. The identity of the subscriber is therefore protected by two methods. The first one is by only transmitting an old TMSI on an unencrypted radio link and the second is by encrypting the new TMSI. There is however a security flaw behind this procedure, as we will see below.

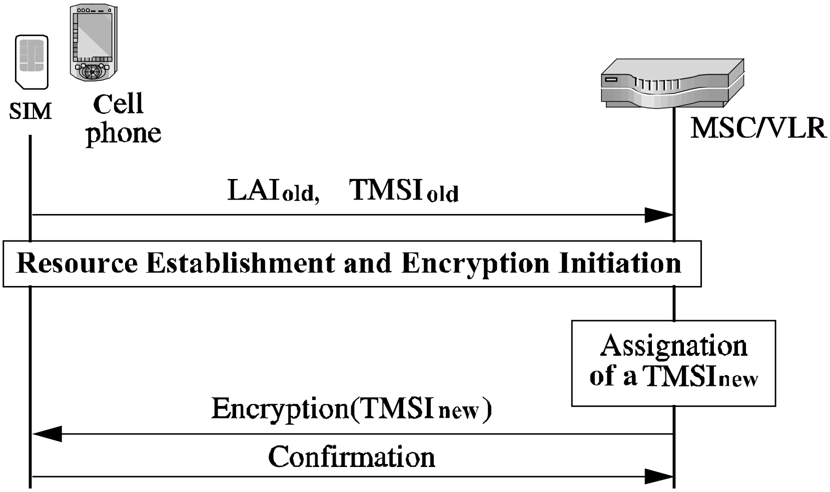


Figure 9.10. *Subscriber's identity protection in GSM*

9.3.2.2. *Access control*

When a new subscriber is added to the network, a secret AK (K_i) is also added along with the IMSI in order to check on its identity. All security mechanisms are based on this secret key, which shall never be either transmitted or compromised. This key is kept safe by the original network in the AuC and by the subscriber in its SIM card.

A symmetric security mechanism used by the GSM authentication process was chosen due to the limited capabilities of chip cards in the early 1990s. As illustrated in Figure 9.11, the authentication process is executed by the SIM card processor based on a cipher called A3, which independently computes an authentication response SRES from a random number (RAND) and the secret key K_i. This response is sent to the GSM network and if it is similar to the expected response SRES from the network, the subscriber is authenticated. At each application of the A3 cipher, the RAND value is changed in order to avoid replay attacks even if eavesdropping occurs.

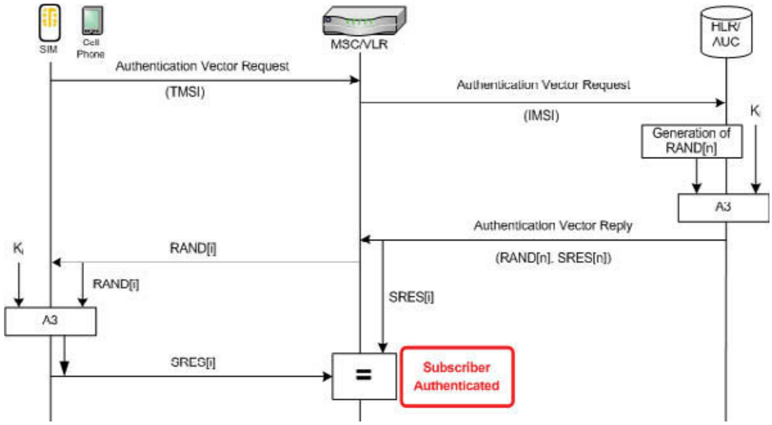


Figure 9.11. *GSM AKA*

The 2-tuple (RAND,SRES) is generated in the AuC by the A3 cipher each time an authentication is required. In order to speed up the process, the AuC may proactively calculate 2-tuple vectors (RAND_i, SRES_i).

Several A3 cipher algorithms exist and each GSM operator may use a different one. As the authentication vector is always provided by the origin network, authentication in roaming mode is guaranteed.

9.3.2.3. *Encryption of radio communication*

The encryption of radio communication is a specific feature of GSM networks that clearly differentiates them from 1G analog and ISDN networks. Only triggered on the express demand of a BSC, the encryption is performed on physical layer transmissions after channel coding and interleaving but before modulation, a particular feature that may add redundancy to the encrypted messaged and thus ease the cryptanalysis.

A cryptography key K_c is generated by the GSM network and a MN from the secret key K_i and a random number (RAND) using the A8 cipher located in the SIM card and at the AuC (see Figure 9.12). This key is then used along with the A5 cipher by the MN and the BTS to encrypt radio communication. There are several versions of the A5 cipher (A5/1, A5/2, A5/3). The A5/1 has been historically limited to Europe, while the A5/2 is a deliberately weakened version of the A5/1 for worldwide use in specific countries. The A5/3 cipher is actually the KASUMI cipher that is also implemented by the UMTS, but with a limited key length of 64 bits and simplified entry parameters in the GSM version. The network calculates the key K_c

in the AoC and generates the 3-tuple (RAND, SRES, K_c) that may then be used on demand.

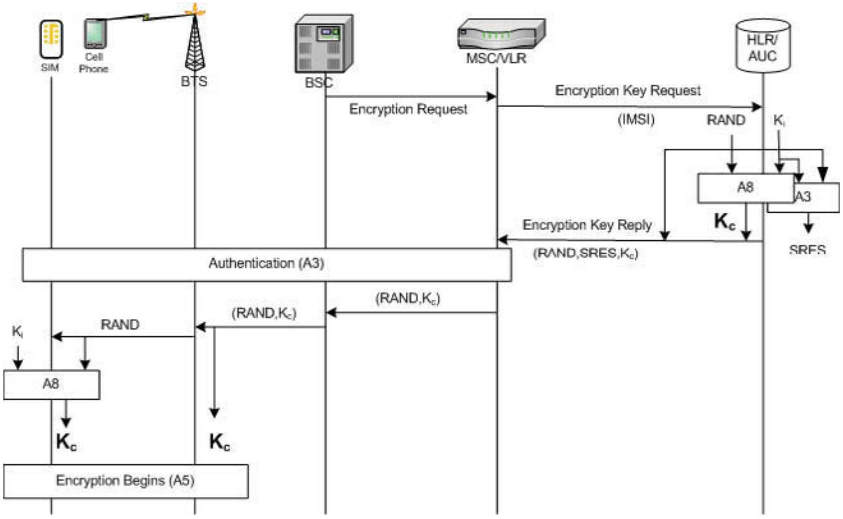


Figure 9.12. *Generation of the GSM encryption key*

Radio communication encryption is actually only performed between a MN and a BTS and is used to secure subscribers and signaling data. As with any symmetric cipher, encryption and decryption are based on the same key K_c and A5 cipher. Figure 9.13 illustrates the establishment of the GSM encryption between a MN and a BTS.

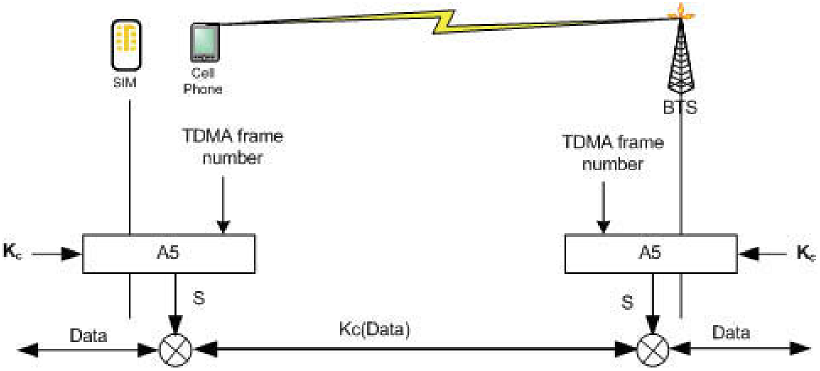


Figure 9.13. *GSM encryption and the A5 cipher*

9.3.3. *Security flaws in GSM radio access*

Security mechanisms employed by a GSM network provide a very high level of protection for the system and for subscribers. Several billion 2-tuples (RAND, SRES) would for example be required in order to defeat the A3 cipher. However, there is no system with a provable 100% reliability. Most flaws actually come from confusing situations where the system needs to transmit sensitive information in plain text. Also, despite encrypted radio communication on the BSS side, SS7-based landline communication on the NSS side is not encrypted. The issue is therefore again to secure signaling communication, which is not a GSM-specific problem. In the following text we will give some examples of security flaws and their consequences on possible attacks.

The first flaw comes from the subscriber authentication and is illustrated in Figure 9.14. As previously mentioned, the system uses a temporary identifier (TMSI) such that it never has to reveal the real identifier (IMSI). However, if the TMSI is lost or when the current VLR cannot recognize it due to a possible failure, the IMSI is nevertheless transmitted in plain text. The IMSI cannot be encrypted with the A5 cipher considering that the system will certainly not transmit any random sequence (RAND) if it does not recognize the subscriber.

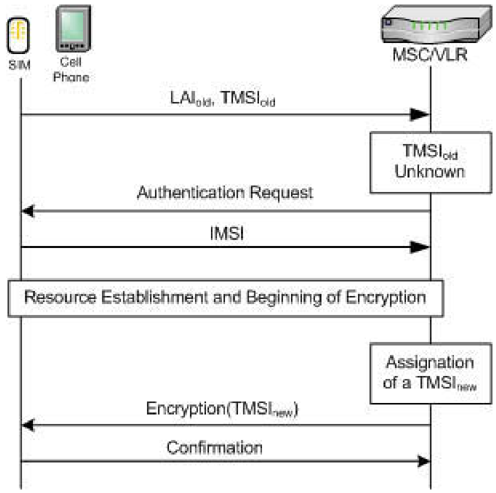

SIM
Cell Phone
MSC/VLR
LAI$_{old}$, TMSI$_{old}$
TMSI$_{old}$ Unknown
Authentication Request
IMSI
Resource Establishment and Beginning of Encryption
Assignation of a TMSI$_{new}$
Encryption(TMSI$_{new}$)
Confirmation

Figure 9.14. *Unknown TMSI and plain text IMSI transmission*

This flaw may be exploited by using forged BTSs and BSCs. Indeed, these relays could permanently reject a subscriber until it transmits its IMST in plain text. This type of attack is in principle not common in GSM networks and could be fought by a mutual subscriber-BSS authentication that has unfortunately not been considered for the GSM network. It has been added to UMTS security though. The reason behind this omission is that GSM networks were considered to be reliable while new players in UMTS could not be.

Illegally obtaining a secret key K_i is not a trivial task as the key itself is also encrypted on the SIM card. Moreover, the key never leaves the SIM card (not even for the MN) or the AoC. Yet, several breaches have been identified. The underlying A3 or A8 cipher may be independently and arbitrarily chosen by GSM operators. In practice, the COMP128 protocol that is specified in the GSM standard (but never published) has been widely used. In 1998, Briceno, Goldberg and Wagner reverse engineered COMP128 and then launched cryptanalysis studies that let them identify a breach in the generation of the secret key K_i [ISA 98]. Although corrected by COMP128-2, it has been proven that it was theoretically possible to clone a SIM card, although this technique required decent equipment and approximately 8 hours. More recently, however, Rao, Rohatgi, Scherzer and Tinguely obtained this key in less than a minute [RAO 02].

The original GSM encryption cipher is A5/1 whose use was limited to Europe. Following the worldwide success of the GSM, a reliable A5/2 version was proposed. Similarly to COMP128, neither A5/1 nor A5/2 were published but were reverse engineered in 1999. Biryukov, Shamir and Wagner [BIR 00] notably illustrated the weakness of the A5/1 and A5/2 encryption ciphers by successfully obtaining the cipher key K_c using a simple PC as a computing center. In 2003, Barkan, Biham and Keller [BAR 03] described a set of attacks on the A5/1, A5/2 and A5/3 ciphers and even on the GPRS that theoretically made it possible to obtain the secret cipher key K_c and to decrypt conversations in real time. Operators slowly migrate to more secure versions of GSM security mechanisms but are delayed by the requirement to also replace subscribers SIM cards.

Another flaw comes from SIM card cloning. If an attacker succeeds in cloning a SIM card and then turns a MN on, the network will detect two mobile devices with the same identifiers at the same time and will close the subscription, therefore impeding identity thefts. However, such attacks will go undetected if the attacker is only interested in eavesdropping on radio communications. Indeed, the intruder has access to the secret key K_i, receives the RAND and may therefore generate the encryption key K_c and passively decrypt communications between the cloned MN and the attached BTS. Some solutions have been proposed in order to mitigate such attacks, notably by injecting copy protections into SIM cards and rendering them unclonable.

Like the authentication flaw previously described, a simpler way of eavesdropping on subscriber conversations is also based on forged BTSs and BSCs.

Although MNs have to provide authentication credentials to operators, not a single form of authentication exists for the operator to the MN. If an attacker manages to guarantee that its BTS always has a stronger signal than any other legitimate BTS in its vicinity, MNs will attach to it. Even though it cannot generate the encryption key K_c and thus cannot decrypt communications, it may still simply deactivate radio encryption. It can therefore freely eavesdrop on any communication transmitted to its forged BTS. The intruder may also pass on or receive conversations that will be charged to the legitimate subscriber's account.

Also, transmissions only being encrypted between a MN and a BTS and the NSS being based on the unsecured SS7, if an intruder manages to access an operator's signaling, it will also be able to eavesdrop on any transmission.

In conclusion, major GSM security flaws find their origin in the lack of any form of mutual authentication, in the possible yet unfortunate plain text transmission of secrets and in cryptanalytic weaknesses of the A3, A5 and A8 ciphers. These flaws have been identified by the 3GPP community and provisions have been added to the UMTS standard.

9.3.4. *Security flaws in GSM signaling*

As previously described, the MAP/CAP protocols are used by GSM signaling for all interactions between various network elements. Based on SS7, there is no specific security mechanism for these protocols, the perturbation of which may yet be the source of significant troubles for a cellular network. For example, exchanging information between two different HLR/VLRs that belong to different networks could be an intrusion vector. We could already observe in section 0 the complexity of the SS7 lower layers. Application-layer security therefore seems more appropriate to that case. Although the 3GPP already foresees a progressive replacement of the SS7 protocol stack by IP, a transitional phase where MAP nodes based on SS7 will communicate with their counterparts based on IP is to be expected.

It has therefore been decided to provide application-layer security based on *MAPSec* [MAP 05]. This approach offers three levels of protection. Each MAPSec message consists of a MAP header and a MAP secured body. In any protection level, the header is sent in plain text in the network. The key agreement mechanism is relatively heavy and is controlled by a Key Administration Center (KAC) in each network which is also governed by roaming agreements between operators. KACs communicate with each others using an IP interface and negotiate security keys based on the Internet Key Exchange (IKE) protocol. 3GPP recommends using the EAS (Rijndael) cipher to generate encryption keys and to check the integrity of received messages.

The MAPSec architecture provides *authentication*, *integrity*, *replay protection* and finally *data encryption*.

9.3.4.1. *MAPSec protection 0*

Protection level 0 does not include any security provision. It is identical to plain MAP.

9.3.4.2. *MAPSec protection 1*

Protection level 1 includes *authentication* and *integrity* provisions which are provided by an MAP and a session key f7.

9.3.4.3. *MAPSec protection 2*

Protection level 2 extends level 2 by adding *privacy* on the message body, which is performed by encrypting the message body with a cryptographic key f6.

Security associations are executed between two networks and remain valid for a predetermined lifetime. Their distribution in MAP nodes is controlled by the KAC. Finally, MAP nodes must be altered by a set of operations called *Secure Transport* to support the encapsulation of MAP components by MAPSec. Figure 9.15 illustrates the MAPSec protection mechanism.

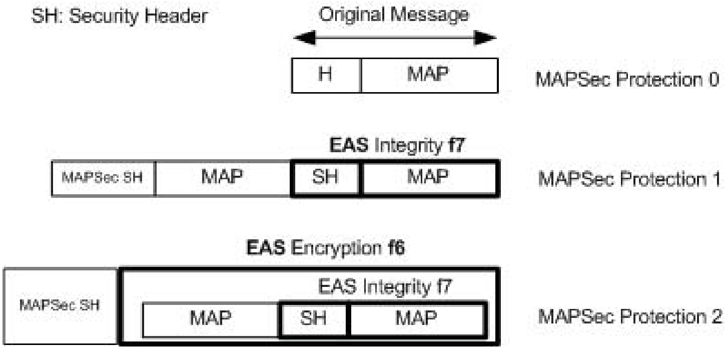


Figure 9.15. *MAPSec packet format*

AS a result of MAPSec, a significant security breach in the NSS could be corrected. Figure 9.16 outlines the macroscopic architecture of MAPSec.

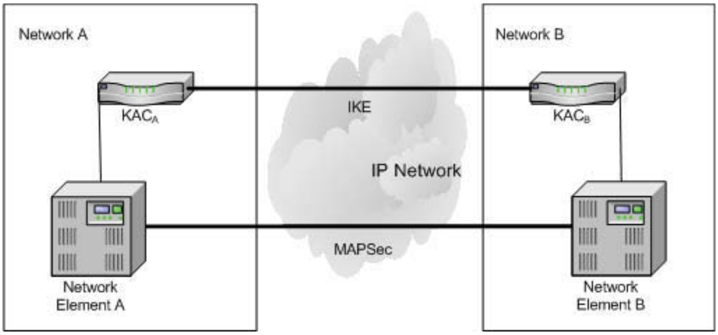


Figure 9.16. *Network interconnection based on MAPSec*

9.4. GPRS security

General Packet Radio Service (GPRS) is a mobile telecommunication standard derived from the GSM that theoretically promises a higher data throughput for sporadic traffic. While GSM is a 2G network, GPRS is in practice often described as a 2.5G network as it is technologically located between the GSM and the UMTS. The GPRS extends GSM by adding best-effort packet-switched communications for low latency data transmissions.

9.4.1. *GPRS architecture*

Unlike GSM, GPRS is able to provide a packet-based IP connectivity to a MN and also proposes a higher throughput by allocating radio resources as a function of the volume of information to be transferred. From an architectural point of view, a GPRS network exists in parallel with a GSM network benefiting from the later for voice communication but using its own infrastructure for data communication. The GPRS adds two new entities:

– *Serving GPRS Support Node (SGSN)*: this manages the attachments of the MN in the service zone and acts as a transit interface for packets on their way to a GGSN. The link between a SGSN and a GGSN is based on the IP, but user traffic is encapsulated in a proprietary protocol called the *GTP (GPRS Tunneling Protocol)*. Concerning security, the SGSN has the same role as a BSC as it is in charge of authentication, integrity and communication authorization.

– *Gateway GPRS Support Node (GGSN)*: this acts as an interconnection gateway between an operator's packet-oriented network and IP networks. The GGSN also runs a firewall in order to control access to its network, collects traffic statistics for billing and manages session and routing information. Last but not least, it provides an IP address to a MN that remains valid for the duration of an attachment.

Three main interfaces may be emphasized in a GPRS network:

– Gp: interface between an internal SGSN and an external GGSN through a border gateway (mostly a firewall);

– Gi: interface between a mobile operator and an external network (Internet or private networks) through a GGSN;

– Gn: interface between GGSNs and SGSNs of a same operator.

Figure 9.17 illustrates the various elements of a GPRS network and their interconnections.

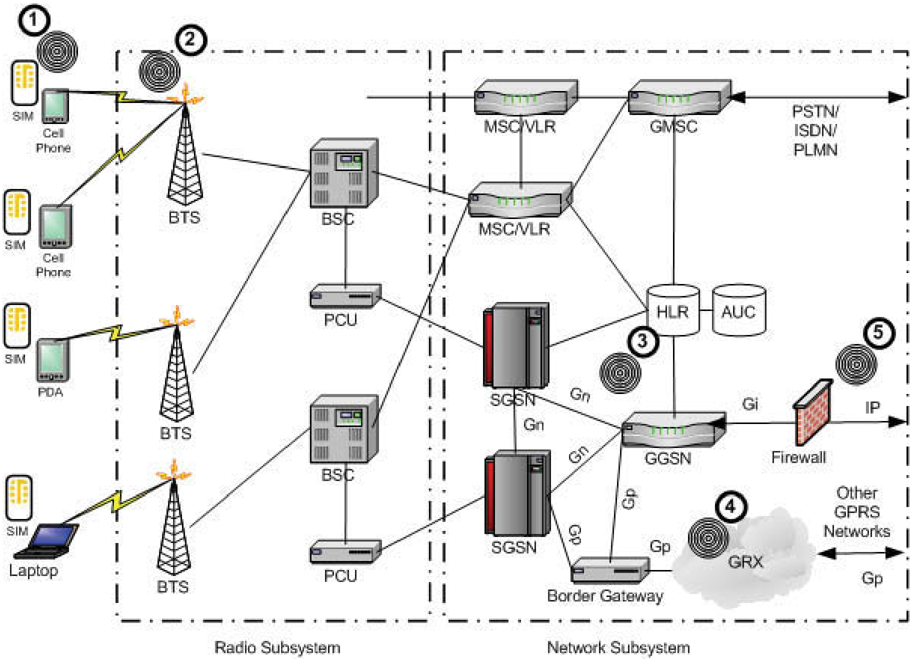


Figure 9.17. *GPRS network architecture*

9.4.2. GPRS security mechanisms

Due to the application-oriented GPRS development, its security shall be analyzed at the structural level as well as at the application level. Structural security may be separated into three parts: GPRS radio subsystem access control, GPRS session access control and GPRS network subsystem access control. We will illustrate below the various security mechanisms of the GPRS. We suggest that the interested readers refer to the GPRS 3GPP standards [GPR 02, PDN 05, GEA 03] for further details relating to these mechanisms.

9.4.2.1. *GPRS radio subsystem access control*

The large majority of GPRS security mechanisms are identical to those of the GSM, notably authentication and access control. The novelty comes from packet-oriented security instead of call-oriented security. This clearly impacts on the encryption that is now performed at the protocol level instead of the physical layer.

9.4.2.1.1. GPRS subscriber authentication

The GPRS subscriber authentication process is similar to that of the GSM. The major difference is that the authentication is not handled by a BSC but by a SGSN and uses a different and independent random number GPRS-RAND. Accordingly, the GPRS network provides a distinct challenge reply (GPRS-SRES) and a GPRS encryption key (GPRS-K_c) from the GSM network.

9.4.2.1.2. GPRS data encryption

GPRS data and signaling encryption is based on the GPRS A5 cipher, more commonly known as the *GPRS Encryption Algorithm (GEA)* in order to be clearly distinguished from its GSM alter ego. However, the encryption itself however varies from that of the GSM in several respects. First, the encryption is not only done between a MN and a BTS as for GSM but up to the SGSN. The GPRS-K_c key is therefore separately stored from the GSM K_c key. Then, unlike GSM, GPRS does not encrypt the physical channel itself but a logical channel at the LLC (Logical Link Control) layer as GPRS traffic is multiplexed on the exact same radio resource as that of the GSM. The encryption itself is therefore done at a higher protocol layer. Finally, similarly to the GSM, several versions of the GEA cipher exist (GEA1, GEA2, GEA3), where GEA3 is also a simplified version of the KASUMI cipher from the UMTS AKA, but are all strengthened by two new parameters:

– *GPRS-K_c*: specific GPRS encryption key;

– *$Frame^{LLC}$* TDMA LLC frame number;

– *Direction*: direction of transmission (from a MN to a SGSN or vice versa).

The objective of these two new parameters is to allow a separate encryption as a function of the packet's position on the TDMA frame and of the logical transmit direction. Figure 9.18 depicts the GEA3-based encryption establishment process.

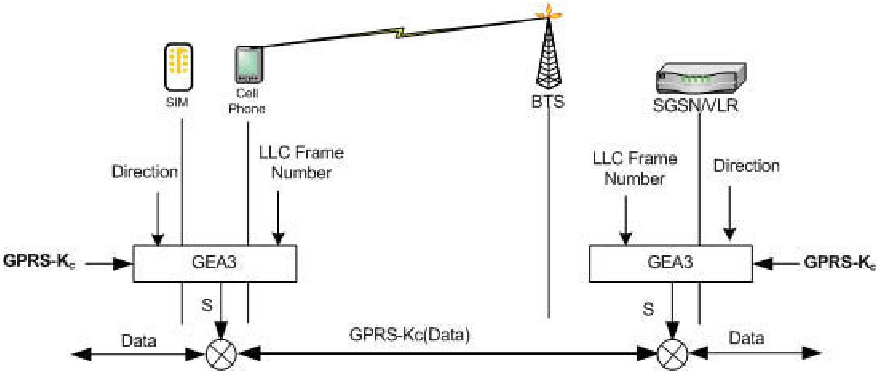


Figure 9.18. *GPRS encryption*

9.4.2.2. *GPRS session access control*

Unlike GSM, where a successful radio access also guarantees access to services (it is not necessary to re-authenticate to send a SMS) the GPRS must establish a logical connection to make a MN reachable for a particular GPRS service. A specific mechanism called a *PDP (Packet Data Protocol) context* has been created in order to establish a logical link between a MN, a SGSN and a GGSN, possibly making the MN visible to external data networks (such as Internet or private networks). A PDP context allocates an IP address to the MN and defines the routing, billing, security and QoS contexts provided by the GPRS to the MN for a particular service. By analogy with mobile IP, a GGSN is similar to a Home Agent (HA) while a SGSN is similar to a Foreign Agent (FA) within a PDP context.

A PDP context may be established after a *GPRS Attach* (a similar yet more complex process than a GSM registration). The PDP is in charge of managing GPRS sessions, establishing logical tunnels between a MN and a GGSN for IP services and updating routing information contained in the GGSN while roaming.

A PDP context may be:

– *static*: in this case, a public and static IP address is allocated either by the operator or by an Internet Service Provider (ISP);

– *dynamic*: in this case, a public dynamic IP address is allocated to a successful PDP context establishment.

Moreover, a PDP context enable transparent or non-transparent Internet access:

– *transparent IP access*: a static or dynamic IP address is allocated by the GPRS operator and its DHCP server. The subscriber does not need to authenticate once again during the PDP context establishment. No security or privacy provisions are either provided by the GPRS network subsystem or by any other visited IP networks. As a consequence, depending of the desired services, a second authentication may be requested at the application level between an ISP or any other private network and the GPRS subscriber (for example: IPsec VPN);

– *non-transparent IP access*: a static or dynamic IP address is allocated by an ISP or any other visited private IP network. A GPRS subscriber must therefore authenticate again to the GGSN during the PDP context establishment. The GGSN requests the subscriber's identification and access rights to a RADIUS server maintained by the ISP or any other private IP network and then only obtains an IP address from the DHCP. The level of security for the communications between the GPRS operator and the ISP or any other private IP network depends on mutual agreements.

Figure 9.19 illustrates a *non-transparent PDP context* establishment schema following a *GPRS Attach*. For reasons of clarity, only a part of the exchanged messages are depicted.

9.4.2.3. *GPRS network access control*

In order to manage roaming between two different GPRS networks without using the Internet as a man-in-the-middle, the 3GPP standard created a *GPRS Roaming Exchange (GRX)*. This is a secured IP network interlinking GPRS operators on the Gp interface and is used to transfer roaming traffic, roaming subscriber information or DNS-related information. A GRX is never connected to the Internet. Using a GRX, *visited SGSNs* are directly connected to the *anchor GGSN* on the home network. In order to control the in/out flows on their network through the Gp interface, GPRS operators also included a *Border Gateway (BGW)* as a gateway between two GPRS networks. Messages are routed using the IP-based *GPRS Tunneling Protocol (GTP)* that unfortunately does not include any kind of security provision and is therefore one possible vector of attack on a GPRS network.

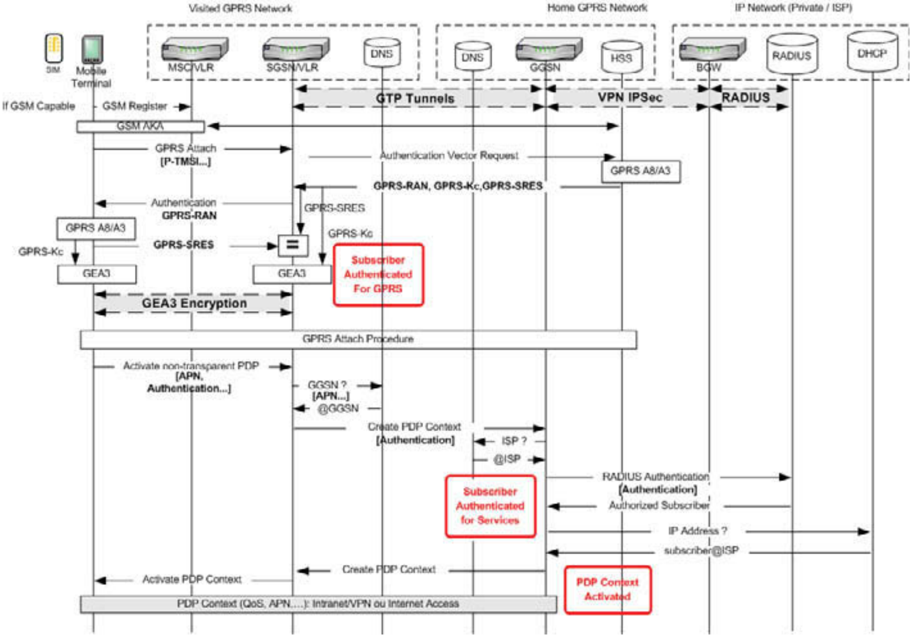


Figure 9.19. *PDP context establishment*

The second access point to a GPRS network is the GGSN itself through the Gi interface. It is the only network element that protects a GPRS network from the Internet and controls the network access using a firewall.

9.4.3. *Exploiting GPRS security flaws*

In Figure 9.17 we depicted five weak points of a GPRS network with respect to attacks:

– the mobile terminal or the SIM card;

– the GPRS radio link;

– the GPRS internal infrastructure (Gn interface);

– the interconnection of GPRS operators (Gp interface);

– the connection to the Internet (Gi interface).

We now describe various possible attacks on a GPRS network. For a more detailed description, see [XEN 06].

9.4.3.1. *Attacks on the mobile terminal or the SIM card*

Although having gone through significant improvements following similar attacks on the GSM network, this part is no less sensitive to spoofing and compromisation.

Authentication algorithms on the SIM card being identical to those of the GSM, attacks similar to those described in section 9.3.3 may unfortunately still be initiated.

A new vector of attack on the GPRS network has its roots in mobile terminals interacting with computer systems and also, through GPRS, with the Internet. We may therefore imagine attacks from computer viruses or worms that are very common on the Internet. GPRS mobile terminals are also no more secure as it is also based on an exploitation system that may also be compromised.

A virus could notably compromise a GPRS service in order to stealthily send traffic to specific potentially illegal destinations. GPRS operators charge communication based on the amount of transferred megabytes, so an attack would lead to non-negligible financial consequences.

9.4.3.2. *GPRS radio link attacks*

Although GEA3 encryption is much more developed than GEA1/2, it is still a target for eavesdropping and interception. GEA3 is indeed based on the KASUMI cipher but is deliberately simplified in order to be run on resource-limited GSM/GPRS mobile telephones. The length of the encryption key K_c is for example limited to 64 bits and therefore provides a weak confidentiality. Barkan, Biham and Keller [BAR 03] notably described theoretical attacks on GEA including GEA3.

9.4.3.3. *GPRS internal infrastructure attacks (Gn interface)*

Like the GSM network, the GPRS core network infrastructure is very vulnerable. Partly based on SS7, it unfortunately inherits all its weaknesses. The IP-based GTP protocol being unsecured, eavesdropping or interception of messages exchanged between SGSNs and GGSNs is conceivable. An intruder may also initiate denial-of-service (DoS) attacks on the signaling or may try to obtain information from a HLR or the AuC. It is therefore recommended to always use protocols such as MAPSec or IPsec on the Gn interface.

9.4.3.4. *Attacks on the interconnection between GPRS operators (Gp interface)*

The critical GPRS structural security is the GGSN. It is indeed the only network element to protect the PLMN from the IP world and is the only entry point on the GPRS network for external IP networks. Inheriting from the NSS internal signaling weaknesses, authentication is not required between SGSNs and GGSNs. A compromised GGSN could however have a dramatic impact on a GPRS network.

A possible exploitation of this security flaw may be performed by using GTP or GRX. GTP is an IP protocol used to handle roaming in GPRS networks and does not include any security provision. It is used at the SGSN to create, release or extend a GPRS session to any legitimate subscriber roaming from another SGSN. However, formal authentication of the roaming SGSN to the home SGSN is not required and consequently opens the gate to serious GTP attacks (illustrated in Figure 9.20).

By compromising a SGSN to act as a legitimate foreign SGSN, an intruder may send GTP packets to a GGSN and in turn compromise its services. For instance, it is possible to intercept legitimate GTP packets and request a subscriber disconnection. It is therefore recommended to always use secured protocols such as IPsec on the Gp interface.

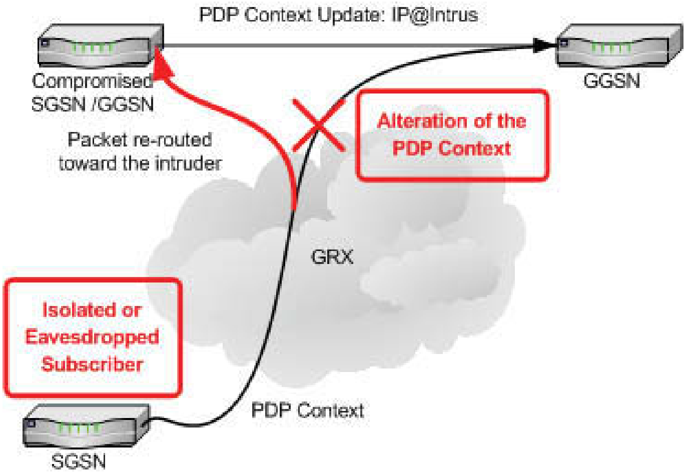

PDP Context Update: IP@Intrus
Compromised SGSN /GGSN
GGSN
Packet re-routed toward the intruder
Alteration of the PDP Context
GRX
Isolated or Eavesdropped Subscriber
PDP Context
SGSN

Figure 9.20. *GTP-based attack example*

9.4.3.5. *Attacks on the Internet connection (Gi interface)*

GPRS operators may not only be targets of attacks from inside their networks but also from outside. The Gi interface connects a GPRS network to the Internet and therefore exposes it to multiple classes of Internet-specific attacks such as worms and other viruses whose objectives are usually a denial of service.

Another form of potential attack is spam. Subscribers are charged based on the amount of megabytes transferred on the GPRS network. Not considering the overload created on the GPRS NSS itself, a bulk of spam e-mails would have a non-negligible impact on a subscriber's monthly bill.

In order to fight such threats, GPRS operators are protected by firewalls. However, their configuration remains very complex considering they must not only analyze IP traffic but also GPRS network security policies such as the monitoring of session initiations established from outside the home GPRS network.

Other forms of attacks are obviously possible depending on the employed interface. However, they all benefit from similar flaws such as the lack of formal authentication between network elements, or even between different networks, and the weaknesses of GTP. We can also mention the increasing popularity of DoS attacks that are inspired by IP networks. In [JUN 04], Bavosa described the major security weaknesses in the GPRS Network SubSystem (NSS) and proposed recommendations to correct them.

Figure 9.21 graphically summarizes the various structural attacks on a GPRS network.

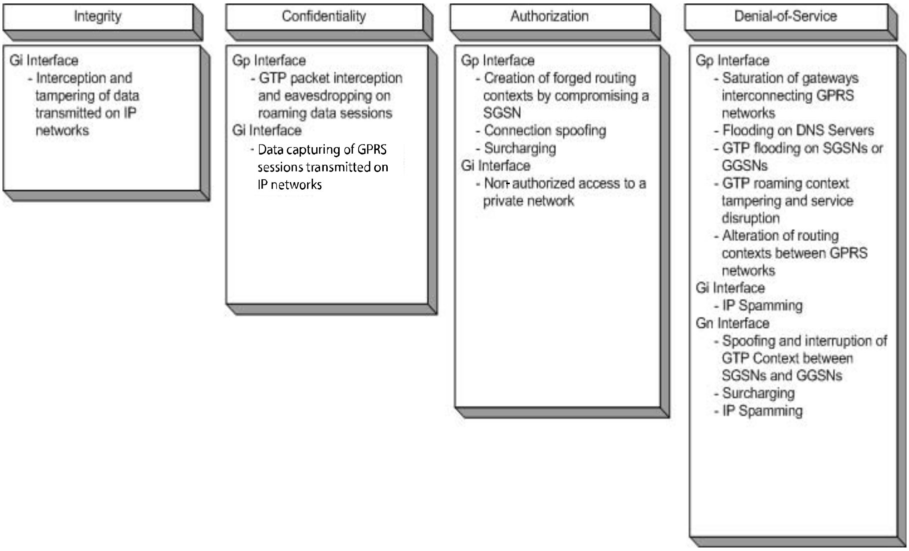

Integrity
Gi Interface
- Interception and tampering of data transmitted on IP networks
Confidentiality
Gp Interface
- GTP packet interception and eavesdropping on roaming data sessions
Gi Interface
- Data capturing of GPRS sessions transmitted on IP networks
Authorization
Gp Interface
- Creation of forged routing contexts by compromising a SGSN
- Connection spoofing
- Surcharging
Gi Interface
- Non-authorized access to a private network
Denial-of-Service
Gp Interface
- Saturation of gateways interconnecting GPRS networks
- Flooding on DNS Servers
- GTP flooding on SGSNs or GGSNs
- GTP roaming context tampering and service disruption
- Alteration of routing contexts between GPRS networks
Gi Interface
- IP Spamming
Gn Interface
- Spoofing and interruption of GTP Context between SGSNs and GGSNs
- Surcharging
- IP Spamming

Figure 9.21. *Taxonomy of GPRS attacks*

9.4.4. Application security

Two application security protocols have been proposed for the GPRS. The first is the *Wireless Application Protocol (WAP)* maintained by the WAP Forum and the second is the *i-Mode* owned by NTT DoCoMo. We will describe here WAP security mechanisms. For a description of i-Mode security mechanisms, we refer the interested reader to [WAL 02].

WAP technology has been created to enable mobile terminals to obtain Internet contents regardless of the type of mobile terminal or its internal display or processing capabilities. The objective is therefore to build a standard that defines procedures for mobile terminals to access Internet services independently of the employed transmission technologies. The WAP also defines a structuring document language derived from HTML called *WML (Wireless Markup Language)*.

The GPRS provided a way to obtain Internet services while roaming. However, two restrictions still exist on the content effectively provided by the WAP:

– the GPRS network has a limited transmission capability;

– the mobile terminal has limited display and processing capabilities.

The WAP therefore proposes to define a standard describing protocols adapted to such networks and terminals.

The classical access methodology is to request a service on a Web server from a browser on the mobile terminal. Considering the limited resources (processor, size, screen) of mobile terminals, the idea is to simplify HTML and to rely on binary transmissions on the radio interface, which actually means to create an interface and a gateway between the subscriber and the Internet that would adapt services to the target network. As each network or operator has different capacities or transmission policies, this interface must be located at the GPRS operator.

Consequently, the services provided by the WAP are all located at a GPRS operator which controls their quality. Depending on its policy, the access to services may be restricted, such a limited access to e-mails or restrictions on HTTP.

WAP architecture is built on the four elements depicted in Figure 9.22. The WAP milestone is the *WAP Gateway* that has the responsibility of making the transition between the WAP and the Web formats and also of transferring secured contents between the WAP and Web worlds.

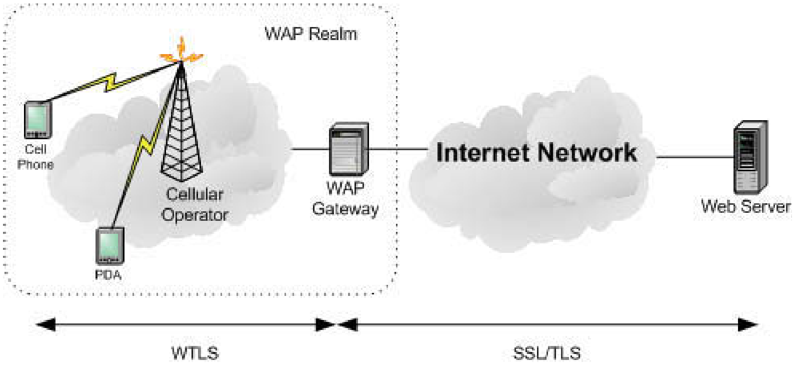


Figure 9.22. *WAP architecture*

In the Internet world, application security is handled by the SSL protocol (to be replaced by TLS in the future). The WAP realm uses a lightweight version of TLS called WTLS (Wireless Transport Layer Security) that contains all secured TLS mechanisms such as the key agreement, signature, symmetric encryption and hash function. The WAP gateway therefore has the responsibility of making the transition between WTLS and SSL or TLS. We must also mention that WTLS has recently been replaced by a more complex end-to-end security specification in WAP 2.0.

WTLS provides the following security mechanisms:

– *confidentiality*: symmetric encryption based on DES, 3DES, RC5, or IDEA;

– *Authentication and Key Agreement (AKA)*: RSA, Diffie-Hellman or elliptic curve Diffie-Hellman certificates;

– *integrity*: HMAC digests based on MD5 or SHA-1.

Although WTLS has been inspired by TLS, it contains several non-negligible cryptographic flaws inherited from simplified hypothesis on TLS [WTL 01]:

– *Hash Keys Truncation*: in order to reduce transmission costs, the HMAC messages used to check the integrity of application messages may be truncated. For example, SHA-40 as defined in WTLS employs SHA-1 to obtain a 40 byte HMAC digest, but in practice, only the first 5 bytes are considered.

– *Man-in-the-Middle*: the WAP gateway must encrypt and decrypt each WAP message. If it becomes compromised, it may act as a legitimate "man-in-the-middle".

– *Oracle*: messages sent in plain text may be a source of information on message encryption.

– *Interoperability*: in order to guarantee WAP availability on any mobile terminal even with the sparsest resources, it is possible to transmit messages in plain text or using a simplified and weak cipher.

For more detailed information relating to WTLS weaknesses, Sengodan, Smith and Abou-Rizk [SEN 00] illustrated the security differences between WTLS and TLS. Saarinen also described in [WAP 99] examples of theoretical attacks on the WTLS protocol.

However, WAP 2.0 proposed improvements such as an end-to-end TLS, HTTP or TCP profiles on the wireless interface and the suppression of the encryption/decryption processes at the WAP gateway.

9.5. 3G security

UMTS is one of the 3G mobile communication technologies. The objectives of UMTS are numerous and provide advantages relating to both voice and data communication. As this technology is based on a larger frequency band, a higher number of calls may be simultaneously serviced. Moreover, its throughput for data communication has been significantly increased. UMTS should theoretically mitigate the current quasi-constant saturation of existing GSM networks and offer higher quality services. In particular, the maximum throughput, which is theoretically five times higher, opens the door to multimedia applications.

9.5.1. *UMTS infrastructure*

The UMTS network has been compelled to guarantee a total interoperability with GSM/GPRS networks. Its infrastructure therefore includes GSM/GPRS-specific and UMTS-specific functionalities, as illustrated in Figure 9.23.

The UMTS mostly reuses GSM and GPRS entities for voice calls or data transmissions. The major difference is located at the protocol layer for each interface and with respect to the radio technology. The following two nodes replace those of the GSM/GPRS:

– node B: replaces the BTS;

– the Radio Network Controller (RNC) replaces the BSC.

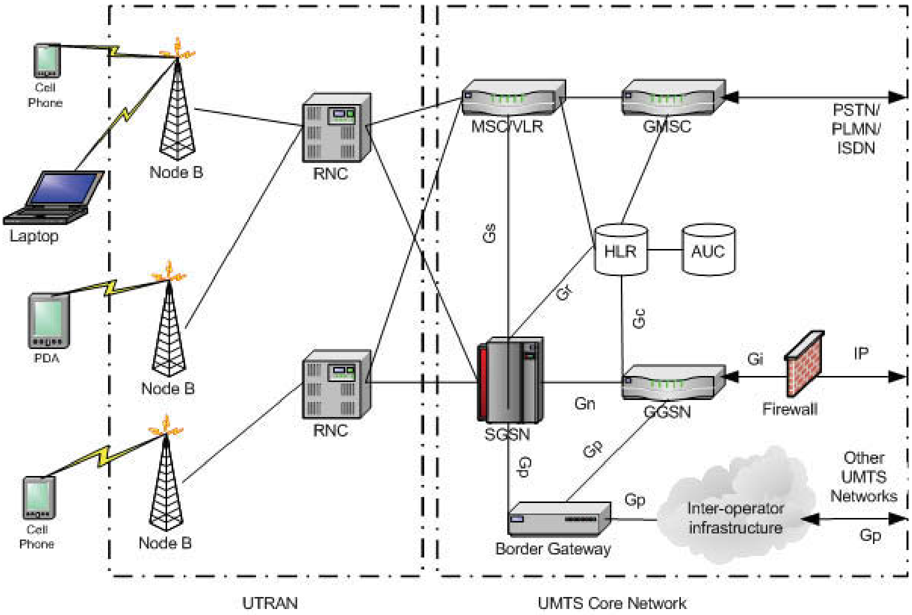


Figure 9.23. *UMTS architecture*

9.5.2. *UMTS security*

The so-called 3G security systems define a higher security management for UMTS networks. New security provisions have been added such as the detection of rogue base stations, the strict control on the context for the transmission of secret keys, network mutual evaluation and identification, longer encryption keys, data integrity and subscriber identity protections. Moreover, a more powerful chip containing an elaborated *Universal Subscriber Identity Module* (USIM) replaces the GSM SIM card.

The novelty in 3G telephony mostly comes from the heterogenity of telecommunication operators. We not only face the interconnection of new cellular telephone operators but also the interconnection of new kinds of communication operators such as Wi-Fi networks, corporate networks, PSTN and any kinds of competitive operators. Such configuration requires robust security management at the signaling and data planes in the UMTS core network. The innovation behind the UMTS is also on the radio part. Given that mobile terminals benefit from increased resources, it is now possible to use more powerful security mechanisms such as TLS or IPsec. A mutual authentication process has been added to the UMTS standard in order to solve some security flaws inherited from the GSM.

UMTS security is composed of five protection categories:

– *network access security*: mutual authentication between a MN and a UMTS network to mitigate attacks on the RSS;

– *network domain security*: protection of the signaling in the operator’s NSS;

– *user domain security*: protection of the access to UMTS terminals;

– *application security*: secured data exchanges between UMTS terminals and UMTS networks at the application layer;

– *visibility*: visibility of the various security measures and the dependency of particular network services on specific security measures.

We graphically illustrate in Figure 9.24 the various security measures of the UMTS specification.

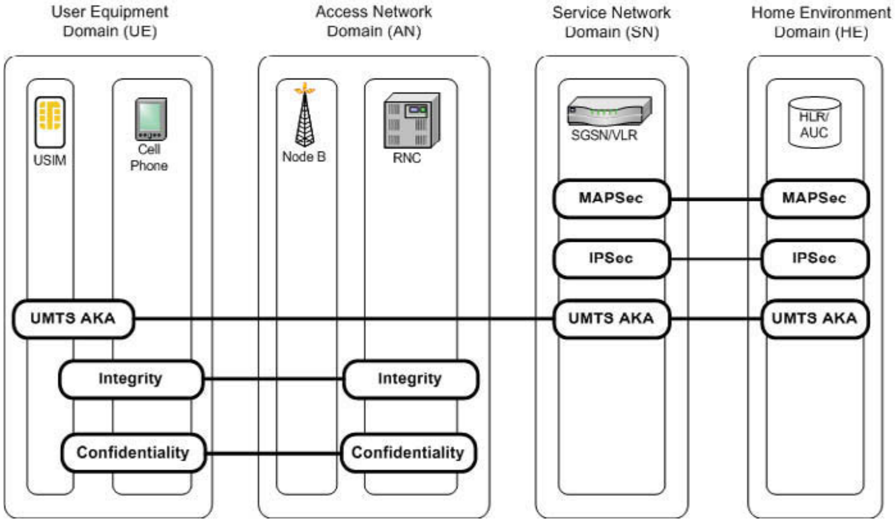
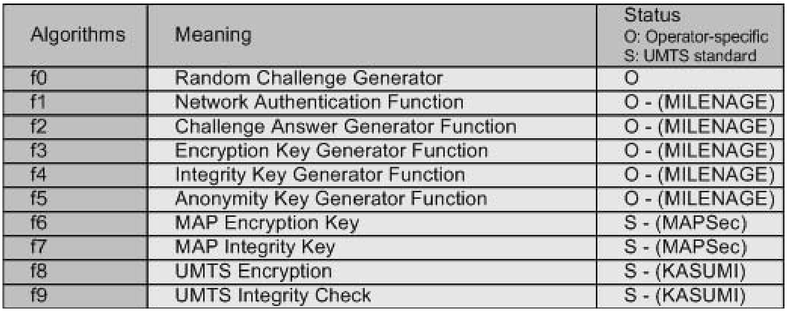


Figure 9.24. *UMTS security architecture*

Figure 9.25 establishes the list of UMTS security algorithms.



Algorithms	Meaning	Status O: Operator-specific S: UMTS standard
f0	Random Challenge Generator	O
f1	Network Authentication Function	O - (MILENAGE)
f2	Challenge Answer Generator Function	O - (MILENAGE)
f3	Encryption Key Generator Function	O - (MILENAGE)
f4	Integrity Key Generator Function	O - (MILENAGE)
f5	Anonymity Key Generator Function	O - (MILENAGE)
f6	MAP Encryption Key	S - (MAPSec)
f7	MAP Integrity Key	S - (MAPSec)
f8	UMTS Encryption	S - (KASUMI)
f9	UMTS Integrity Check	S - (KASUMI)

Figure 9.25. *UMTS security algorithms*

9.5.2.1. *Secured UMTS network access*

The new UMTS AKA mechanism is depicted in Figure 9.26. Although partially similar to GSM, a significant difference must be noted. Instead of using a 3-tuple (RAND, SRES, K_c), the *UMTS AKA* requires a 5-tuple (RAND, SRES, CK, IK, AUTN) based on the *MILENAGE* algorithm and including two new parameters:

– *AUTN*: network identification token added to let subscribers identify the network they are trying to connect to. This token contains three fields:

- *Authenticated Management Field (AMF)*: defines the operator-specific operations such as the definition of the required algorithm(s) or a key's lifetime,

- *Sequence Number (SQN'=SQN* ***XOR*** *AK)*: defined by the AuC in the home network, it is protected by an *Anonymity Key (AK)* as it could provide the identity and the position of a subscriber. The sequence number is used to mitigate *replay attacks*,

- *Message Authentication Code (MAC-A)*: by comparing it with the value computed by the USIM, the mobile terminal can authenticate the network's identity;

– *IK* (integrity key): used to control the integrity of each transmitted message at the signaling and data planes.

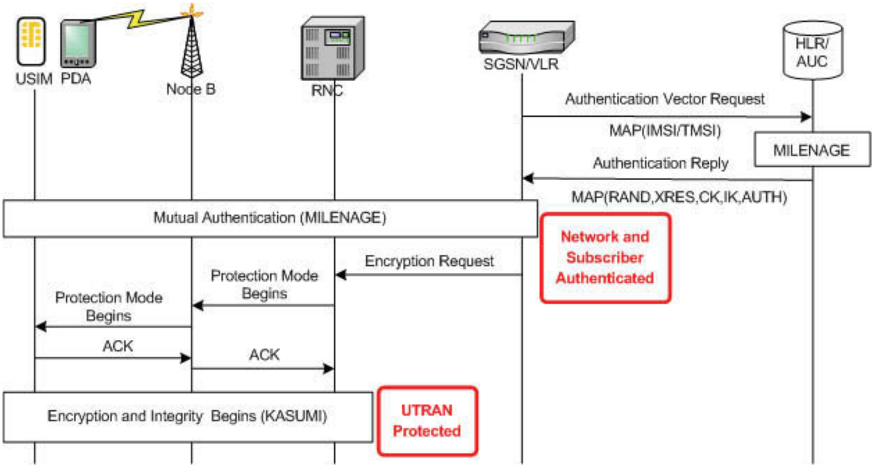


Figure 9.26. *UMTS AKA and encryption*

The following schema (Figure 9.27) is an example of the UMTS mutual authentication process. For reasons of clarity, we chose not to include the complex mechanisms to obtain the AUTH field at the AuC or the XMAC-A at the USIM.

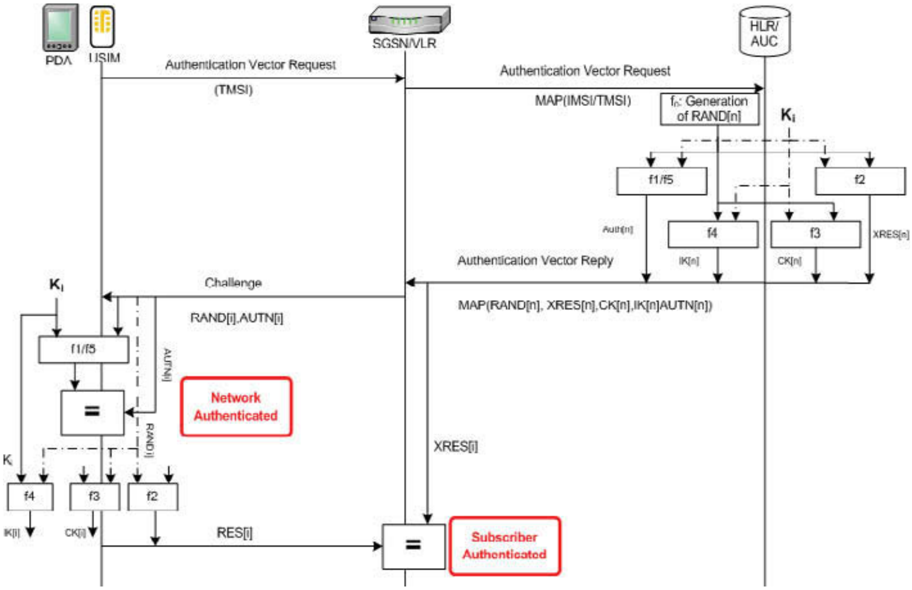


Figure 9.27. *UMTS AKA*

We also preferred to separately illustrate the MILENAGE algorithm at the AuC and the USIM in Figure 9.28 and Figure 9.29 respectively.

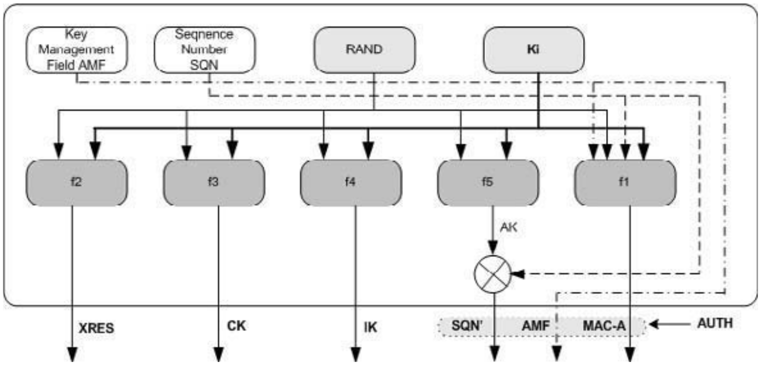


Figure 9.28. *MILENAGE at the AuC*

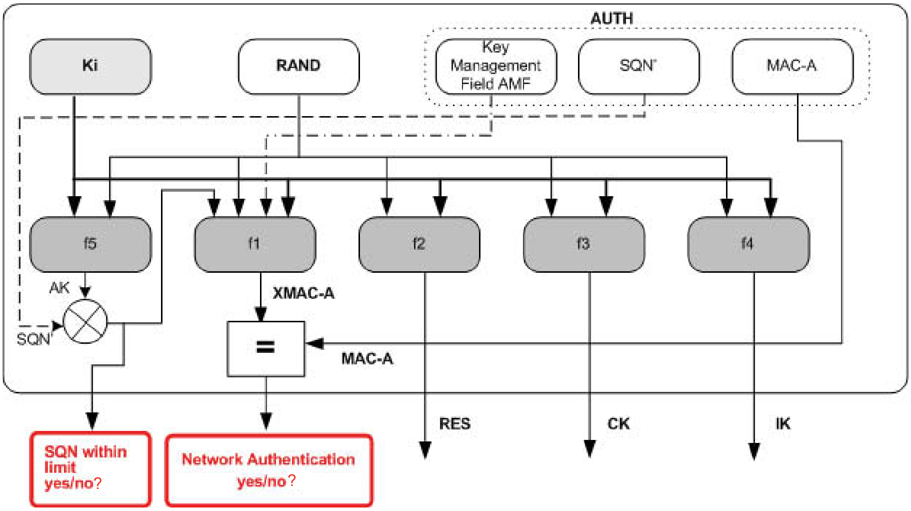


Figure 9.29. *MILENAGE at the USIM*

Another significant measure implemented by UMTS is end-to-end integrity and encryption management. The dual mechanism is based on the complex KASUMI cipher algorithm (see Figure 9.30). Moreover, in order to mitigate Oracle-based attacks, some control messages are at least protected against integrity breaches and even encrypted. Figure 9.26 illustrates the registration process with an encryption request.

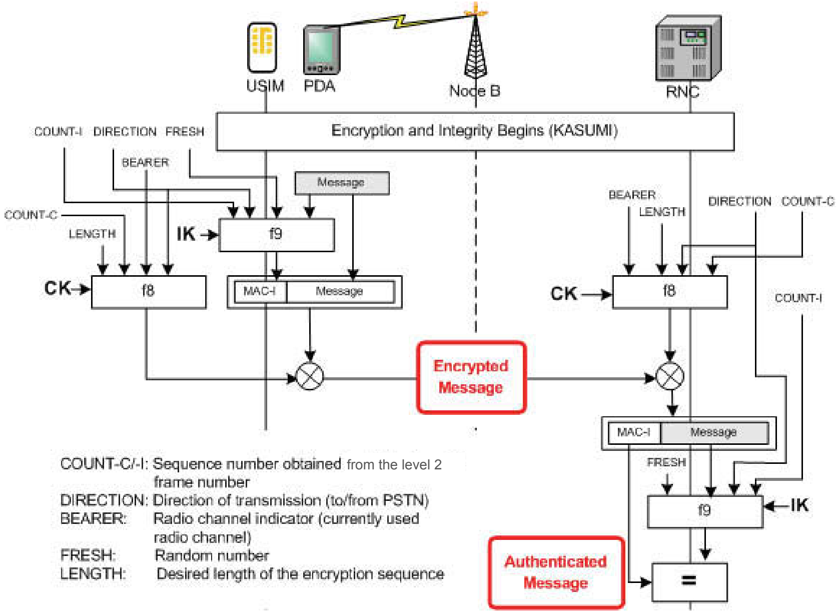

USIM
PDA
Node B
RNC
Encryption and Integrity Begins (KASUMI)
COUNT-I
DIRECTION
FRESH
BEARER
Message
COUNT-C
LENGTH
IK
f9
CK
f8
MAC-I
Message
Encrypted Message
BEARER
LENGTH
DIRECTION
COUNT-C
CK
f8
COUNT-I
MAC-I
Message
FRESH
f9
IK
Authenticated Message
=
COUNT-C/-I: Sequence number obtained from the level 2 frame number
DIRECTION: Direction of transmission (to/from PSTN)
BEARER: Radio channel indicator (currently used radio channel)
FRESH: Random number
LENGTH: Desired length of the encryption sequence

Figure 9.30. *The KASUMI algorithm*

By adding a mutual identification and more robust encryption and integrity protections, the UMTS AKA corrects a major security breach in the network access to GSM/GPRS.

9.5.2.2. *UMTS network subsystem access security*

The clear objective when securing the network subsystem is actually to secure signaling in an operator's core network and between different operators. The proposed security system secures all messages transmitted on SS7 as well as on IP networks. It accordingly requires two types of security protocols, MAPSec and IPsec:

– *Mobile Application Part Security (MAPSec)* [MAP 05]: as previously described in sections 9.2.5 and 9.3.4, securing SS7 may only be performed at the application layer. See section 9.3.4 for a description of MAPSec.

– *Network-layer IP Security (NDS/IP)* [NDS 06]: all IP-based interfaces at the NSS use IPsec-ESP with plain text headers. Moreover, IPsec-ESP is configured for a tunnel mode to secure IP messages between IP gateways. The distribution and key exchange is done by IKE. Figure 9.31 illustrates NDS/IP security.

By providing two methods to secure and authenticate the origin of any message transmitted on SS7 or IP, UMTS corrects a second major security breach illustrated in GSM and GPRS networks.

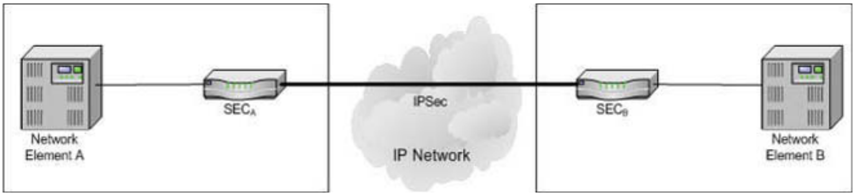


Figure 9.31. *Network interconnection based on IPsec*

To conclude this section, after having learned from the identified security flaws of GSM/GPRS networks and benefiting from increased capabilities of UMTS mobile terminals as well as the power of IP key distribution and management, 3GPP defined security rules that, if correctly applied, should efficiently protect operators and subscribers against malicious intrusions.

9.6. Network interconnection

The increasing use of packet-switched networks for real-time voice communications based on Voice-over-IP (VoIP) triggered an increased demand for access to SS7-based IN platforms, consequently requiring the interconnection of SS7 networks with the Internet or other data networks. Until recently, SS7 interconnections were limited for security reasons. Yet, in the light of this increasing demand, such a policy could be reconsidered. Indeed, IN services have already been successfully extended to cellular networks. Now, Local Exchange Carriers (LEC), competitive or not, as well as ISPs also request an access to the SS7 network of PSTNs. For instance, Internet Telephony Providers (ITP) would like to propose IN services such as number portability or free IP calls based on VoIP.

SS7 networks provide a very high stability and resilience but also contain connectivity and security issues. Data networks offer a simplified connectivity at the cost of reduced reliability. The interconnection of both worlds could be beneficial by providing increased access to SS7 networks and a better resilience to data networks. It could however generate stability and security issues on SS7 networks.

In order to connect the Internet to landline networks, it is necessary to make SS7 and IP networks transparently inter-operable. Several working groups have accordingly been created and have proposed four major standards: *H.323, SIP,* *MGCP* and *Megaco.* We give below a brief summary of these protocols and refer the interested reader to Chapter 11 for more details.

9.6.1. *H.323*

H.323 is a standard developed by the IUT and defines a multimedia communication protocol used for packet-switched networks.

9.6.2. *SIP*

The Session Initiation Protocol (SIP) is another standard, developed this time by the Internet Engineering Task Force (IETF). It is actually a signaling protocol managing video calls, telephony and instant messages where at least one participant belongs to a packet-switched network.

9.6.3. *Megaco*

The Megaco protocol, also called H.248, provides external control and management capabilities for data communications through a Media Gateway (MG) and is complementary to H.323 and SIP. Media Gateway Controllers (MGCs) are connected to and control MGs using H.248, whereas they communicate with each others using SIP or H.323.

9.7. Conclusion

The loss of the national telecommunication operators' monopoly and the interconnection of various types of networks (landline, mobile, IP) created or accelerated the collapse of reputation-based structural security that was historically established between different players in the telecommunication world. New solutions had to be created in order to guarantee the security of infrastructures, the confidentiality of subscriber information and the control of transmitted contents.

Several steps were necessary to reach this objective. First, with the development of cellular networks, the radio link between networks and subscribers had to be secured. Several algorithms were developed in order to guarantee data integrity, and subscribers' authentication and security. However, no security provision was devised to protect the network itself.

Next, security flaws on SS7 signaling and in the interconnection of different SS7 networks had to be addressed. Several patches were developed which only postponed the general abandonment of SS7. The appearance of IP in the world of mobile networks made powerful security protocols such as IPsec available and accelerated the appearance of new signaling protocols such as SIP that will eventually replace SS7.

However, the interconnection of different networks was merely the first step towards a larger vision of future mobile telecommunication networks. With Competitive Local Exchange Carriers (CLERC) or Mobile Virtual Network Operators (MVNO) came the notion of service and their interconnections between various networks and service providers. Through this, new security constraints arose that as a consequence made it necessary to provide end-to-end instead of hop-based secured services. Telecommunication networks therefore lost their influence with respect to the provided or transported services. For example, a user who would like to call might not care about the network effectively used (pure PSTN, VoIP, etc.) as long as the QoS remained the same.

The need for transparent or non-transparent access to services and their routing through an interconnection of networks is a very present and controversial topic. The outcome remains uncertain at this time despite the current tendency to go towards total transparency. It may however be envisioned that in the debate on the vision and structure of future mobile telecommunication networks, the user could eventually cast the final vote and as such, we may assist the return of reputation-based security. After all, in chaotic situations it is human nature to only hang around with trusted friends!

9.8. Bibliography

[BAR 03] E. Barkan, E. Biham, N. Keller, “Instant ciphertext-only cryptanalysis of GSM encrypted communication”, *Advances in Cryptology*, Vol. 2729, pp. 600-616, 2003.

[BIR 00] A. Biryukov, A. Shamir, and D. Wagner, “Real time cryptanalysis of A5/1 on a PC”, *Lecture Notes in Computer Science*, vol. 1978, pp. 1-18, 2001.

[BOM 02] K. Boman, G. Horn, P. Howard, and V. Niemi, “UMTS security”, in *IEE Journal on Electronics and Communication Engineering*, Vol. 14, Issue 5, pp. 191-204, 2002.

[GEA 03] “Specification of the A5/3 encryption algorithm for GSM and ECSD, and the GEA3 encryption algorithm for GPRS”, *3GPP TS* 55.216, version 6.2.0, 2003.

[GPR 02] “General Packet Radio Service (GPRS) service description”, *3GPP TS* 101.344, version 7.9.0, 2002.

[GSM 01] J. Eberspaecher, H.J. Voegel, C. Bettstetter, *GSM: Switching, Services and Protocols*, John Wiley & Sons, Second Edition, 2001.

[ISA 98] M. Briceno, I. Goldberg, D. Wagner, "Internet Security, Applications, Authentication and Cryptography (ISAAC)", University of California, Berkeley, see http://www.issac.cs.berkley.edu/issac/gsm-faq.html.

[JUN 04] A. Bavosa, "GPRS Security Threats and Solution Recommendations", White Paper, Juniper Networks, 2001.

[KOE 02] G. M. Køien, "An introduction to access security in UMTS", *IEEE Wireless Communications*, Vol. 11, Issue 1, pp. 8-18, 2004.

[LOR 01] G. Lorenz *et al*., "Securing SS7 telecommunications networks", *Proceedings of the 2nd IEEE Workshop on Information Assurance and Security*, pp. 273-278, June 2001.

[MAP 05] "MAP Application Layer Security", *3GPP* TS 33.200, version 6.1.0, 2005.

[NDS 06] "IP Layer Security", *3GPP* TS 33.210, version 7.1.0, 2006.

[PDN 05] "Interworking between the Public Land Mobile Network (PLMN) supporting GPRS and Packet Data Networks (PDN)", *3GPP* TS 101.348, version 7.10.1, 2005.

[RAO 02] J. R. Rao, P. Rohatgi, H. Scherzer, and S. Tinguely, "Partitioning attacks: or how to rapidly clone some GSM cards", in *Proceedings of the IEEE Symposium on Security and Privacy*, May 2002.

[SEI 98] Reiner Seiler, "Security service in an open service environment", *Proceedings of the 14th IEEE Computer Security Applications Conference (ACSAC)*, pp. 223-234, December 1998.

[SEN 00] S. Sengodan, D. Smith, and M. Abou-Ritzk, "On End-to-End security for Bluetooth/WAP and TCP/IP networks", in *Proceedings of the IEEE Conference on Personal Wireless Communication (ICPWC'2000)*, pp. 399-403, 2000.

[SEN 05] H. Sengar, D. Wijesekara, S. Jajodia, "MTPSec: customizable secure MTP tunnels in the SS7 network", *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, April 2005.

[TEK 01] "Tekelec Eagle STP", White Paper, http://www.tekelec.com/productportofolio /eagle5sas, 2001.

[TEL 01] Telcordia, "General Function of Messages and Signals", Technical Report GR-82-CORE, 2001.

[UMT 05] "UMTS Security Architecture", *3GPP* TS 33.102, version 7.0.0, 2005.

[VER 02] Verizon, "SS7 Security Gatekeeper", Technical Report Request for Information – Verizon Communications, May 2002.

[WAL 02] P. Wallace, A. Hoffmann, D. Scuka, Z. Blut, K. Barrow, *I-mode Developer's Guide*, Chapter 21, pp. 483-510, Addison-Wesley, 2002.

[WAP 99] M-J Saarinen, "Attacks Against the WAP WTLS Protocol", *IFIP Conference Proceedings*, Vol.152, pp. 209-215, 1999.

[WTL 01] "Wireless Transport Layer Security", *WAPForum* WAP-261-WTLS, version 6.0, 2001.

[XEN 06] C. Xenakis, "Malicious actions against the GPRS Technology", in *Journal in Computer Virology*, Springer Paris, Vol. 2, N. 2, pp. 121-133, 2006.

Chapter 10

Security of Downloadable Applications

10.1. Introduction

Most mobile handsets from the first or second generation were *closed devices*. This means that it was not possible to modify their software and install new components after their purchase. With the 2.5G generation of handsets, *open* terminals have appeared[1]. On an open terminal, it is possible to install new applications after the device was sold to the customer almost as easily as on a regular desktop computer (and sometimes more so). In this chapter, we present the security issues raised by this new functionality.

We begin with a presentation of the new risks introduced by the opening of the devices and the security objectives that have been defined mainly to protect the end-users of the terminals, but also, in some cases, content providers.

We then detail the different architectures implemented to reach these objectives and their validation:

– open operating systems (Symbian, Linux or Windows Mobile),

– virtual machines (with en emphasis on Java).

Often, applications are verified outside the terminal before they are distributed: we will study the validation techniques used on applications before they are distributed

Chapter written by Pierre CRÉGUT, Isabelle RAVOT and Cuihtlauac ALVARADO.

[1] NTT DoCoMo introduced the first DoJa models in January 2001. Motorola introduced the first MIDP models in April 2001.

and the infrastructures that guarantee that only validated applications have access to the critical resources of the terminal.

Regarding security, there is unfortunately no foolproof solution to prevent malware. However, network monitoring can detect the emergence of new malicious applications and antivirus software makes it possible to eradicate them on terminals or networks, while remote device management can replace vulnerable components.

We conclude this chapter with an overview of the most promising research directions for improving the safety of downloaded applications.

10.2. Opening the handset

Several technologies have been used to provide an execution environment for downloaded applications:

– the applications may be directly executed by the processor:

- usually an open operating system is used to ensure the management of the different applications and the core services of the device. The main difference between closed and open operating systems is that the latter offers a standardized interface to all the resources of the platform with a well-defined usage policy that encompasses a security policy. The main open operating systems available on mobile devices are Windows Mobile, Symbian, Linux and OS X,

- sometimes a set of standardized API over a closed operating system is provided to downloaded applications as an interface with the hardware. System applications still directly access the internals of the operating system. The Brew (Binary Runtime Environment for the Wireless) platform from Qualcomm is an example of such an architecture;

– applications can be executed in a virtual machine. The most widespread environments nowadays are based on the Java ME virtual machine (Java Platform Mobile Edition). According to Sun Microsystems, 1.2 billion compatible handsets were delivered by mid-2006.

The two most common variants of the Java ME platform are MIDP, standardized by a relatively open consortium known as the Java Community Process (JCP), and Doja, a proprietary platform from NTT DoCoMo. There are several ways to integrate the virtual machine into the software of the mobile phone:

– the virtual machine can be placed directly over a closed operating system. This is the case in most low-end or middle-end terminals;

– some mobiles offer an operating system almost completely developed in Java; only the lowest layers are programmed natively. RIM Blackberry, Savaje, Danger and, to some extent, Google’s Android are the main representatives of this architecture. These platforms may offer two different Java environments: one for system applications and a more controlled one for downloaded applications;

– finally, most high-end handsets that use an open operating system offer in parallel one or several virtual machines for the execution of downloaded applications. These terminals then accept both kinds of applications: native applications and Java applications.

10.3. Security policy

10.3.1. *Actors*

The goal of the security policy is to define the rights and duties from the different actors to guarantee for each actor the protection of his assets. The security policy is enforced by the security mechanisms of the execution platform for downloaded applications. It is necessary to distinguish the security policy from the mechanisms enforcing it so that a handset can be tuned to the specific needs of the actors. The main actors are as follows:

– The *handset manufacturer* is the provider of the hardware platform and of the base software provided with it (some of the components may have been developed by third parties).

– The *network operator* provides the network connectivity that gives access to contents. He is often the seller of the handset and sometimes of the contents executed on the platform. He is responsible for providing after-sales services.

– The *content provider*: in this category we have several different actors: individual application developers, software houses, aggregators (intermediaries who propose structured catalogs of applications to on-line service providers (operators, download platforms)).

– The *end-user* of the terminal.

10.3.2. *Threats and generic security objectives*

The security policy for downloadable applications is a subset of the global security policy of the handset [OMT 08, JAN 08].

The main objectives of security mechanisms are the protection of the owner of the terminal, of the end-user and of the subscriber to network access. Often, the owner, the end-user and the subscriber are seen as a single actor, who we then call the user. However, there are several cases where they must be distinguished because they have different objectives and needs. For example, this occurs when a handset is given to a child by his parents, or when the handset belongs to a company and is lent to employees. In all cases, the goal is to protect the assets of one of these actors against a malicious application. We only consider the cases where the actions performed are done in a hidden way: we only consider the case of Trojans.

The owner must be protected from irreversible damage done to the phone. This mainly relates to the risk of destruction of critical data necessary in the use of the handset: configuration data and applications.

The user must be protected against any attempt to jeopardize the confidentiality and the integrity of personal data such as the contact list or the call log. Some mobile phones can also record some audio or video clips: these functionalities could be used to spy on the user. The use of geo-localization services without the knowledge of the user creates the same risks.

The subscriber must be protected against any abusive use of access to the network (risks of over-billing or identity theft) and against denial of access to the network.

10.3.2.1. *Protection of the network*

In contrast to a classical IP network, modern mobile networks are instead closed and equipped with protection mechanisms against malicious terminals or users. Thus, the security mechanisms embedded in the handset are not designed with the protection of the operator network as a primary objective. Nevertheless, some risks must be taken into account. For example, an application could wait until a predefined date to launch an attack against a resource on the network using its wide distribution for launching a distributed denial-of-service (DoS) attack.

10.3.2.2. *Protection of content providers (DRM)*

The legislation of many countries allow the owners of the intellectual property of contents (video, audio, applications, etc.) to use mechanisms to restrict the redistribution of these contents by the user. In some cases, these content providers expect that handset manufacturers and network operators implement some mechanisms that prohibit or limit the copy of these contents out of the terminal. The objectives of these security mechanisms are known as Digital Rights Management (DRM).

The intent of DRM is to protect the content provider against the user of the phone. The main weakness of such technologies lies in the fact that, in most countries, the user of the phone is also its owner and he has full access to both the software and the hardware, which is not tamper-proof. New chipsets try to bury some private keys deep in their hardware to limit the risks, but contents must be unscrambled at some point in order to be used.

10.3.3. *Risks specific to some kinds of applications*

All the applications that may be used on a mobile handset do not have the same security requirements and constraints. Nevertheless, they can be split into broad categories with similar security features and needs. In this section, we give a quick survey of some of these categories.

10.3.3.1. *Telephony applications*

The base applications of the mobile, in particular the handling of phone calls (transmitting and receiving phone calls) and communications with the (U)Sim card have very strong availability requirements. Replacing these applications or using third parties applications concurrently with the native applications create some legal problems as it breaks the phone certification (known as *Type Approval*). Such a risk must be prevented by the security policy of the terminal.

Brew, Windows Mobile and Symbian platforms offer mechanisms to develop telephony applications. However, these can be installed on the phone only if they have been digitally signed by a trusted entity (usually restricted to the network operator – see section 10.5.4.4). The evolution of the MIDP platform described in JSR 253 (Telephony API) uses similar protection mechanisms.

10.3.3.2. *Payments and contracts*

The European Digital Signature Directive [EUR 99] defines a digital signature as a numerical data associated with a document that can be used to identify and authenticate a unique signer because it has been created by means under the exclusive control of the signer. Moreover, the signature must be linked with the data signed in such a way that any modification to the data can be detected. Electronic payment is only a specific case of digital signature where the signed document is related to the purchase of a good from a seller at a given price.

Technical solutions adapted to mobile terminals can be designed to handle this general framework. ETSI has defined an architecture and the security requirements for such an application on a mobile handset [M-C 03]. Of course, it is based on the use of asymmetric cryptography to authenticate the signer.

The main difficulty of electronic payment is the risk of fraud.

Systems must prohibit transactions that are not authorized by the user to protect him against the risks of theft and fraud (diversion of the signed text, use of the device without the consent of the user, theft of secret keys). Nevertheless, this mechanism is also a protection for the vendor against dishonest buyers who would pretend that they have not authorized the transaction made (this is the *non-repudiation* property). All the transactions made must conform to the information presented during the request for authorization so that the transaction has a legal contractual value.

The current software and hardware architectures of mobile handsets are usually considered as inappropriate for operations such as electronic commerce because even if the SIM can host the component that performs the signature in a safe way, the handset cannot establish a safe environment where displaying the contents of the transaction and handling the user answer can be performed in a totally secure way. However, there are recent proposals for new architectures (see section 10.5.5) more adapted for these kinds of transactions and some field trials are underway in Europe.

Finally, in some Asian countries, mobile handsets can be used for electronic payments. It is usually assumed that the risk of fraud is culturally lower in these countries. This also limits the deployment to these areas. To increase confidence in the application, these are usually pre-installed applications and cannot be downloaded afterwards.

10.3.3.3. *On-line gambling*

Games with monetary prizes are a kind of electronic commerce. If the player wins because of his agility to interact with the mobile, or as the result of a lottery, the reward can be considered as a financial transaction. In the case of a reward based on agility on the handset (highest score, competition between players), the risk of fraud is extremely high. The server that operates the game has no means to distinguish a truthful notification from a fake one and this model of transaction cannot be implemented in a completely safe way. In the case of a lottery on the server, the authentication of the user is a necessary and sufficient condition.

10.3.3.4. *Cryptographic libraries*

Most execution environments on mobile handsets implement libraries of security services: encryption and decryption, signature, authentication and secure communication channels. Unless there are restrictions imposed by the handset security policy, applications are free to use these libraries for their needs. Developers should not use these libraries (especially low-level cryptography) unless they use it in a really safe way. Unfortunately, a lot of applications use these in an unsafe way that jeopardizes their security objectives. There are also some applications that exceed the level of trust that can be achieved with these libraries. Finally, some of these library implementations have exploitable vulnerabilities.

10.3.4. *Impacts*

A malicious application can cause several damages to the user. We present some of them (more information on Trojans can be found in [F-S 05]):

– Over-billing. A malicious application may use some services that incur a cost for the user such as downloading data through the Internet using GPRS/UMTS connections, sending text messages or making phone calls without the knowledge of the victim. For example, the CommWarrior virus generates over-billing by sending some MMS to the contacts registered in the victim's address book.

– Attacks on personal information. Mobile phones contain private data belonging to the end-user such as the address book, the call log, the list of received messages, the agenda or personal photos and videos. A malicious application can modify and destroy these data or send them to a third party. For example, the Trojan Cardblock erases SMS and MMS recorded in the infected mobile. Pbstealer tries to send the address book of the infected phone to every nearby phone via Bluetooth.

– Deterioration of the phone behavior. By this we mean any action that prevents normal use of the phone. The Trojan Dampig blocks access to some of the applications of the phone. Fontal and Doomboot install corrupted files on the phone that prevents it from correctly booting during the next start-up. Because it uses Bluetooth heavily, Cabir drains quickly the battery of the phone.

– DoS attacks. A malicious application that is widespread over the network and that can produce a high number of requests toward a given service at a given date can overload the servers and lead to an interruption in service. In 2001, mobile phones from customers of the Japanese operator NTT DoCoMo who had received a malicious e-mail automatically called the police emergency number, preventing the police from correctly answering real urgent calls [BEL 03].

10.3.5. *Contractual and regulatory landscape*

The European legislation [EUR 00] is based on the protection of data:

– the protection of the intellectual property of content provider,

– the protection of private data of the user,

– the crackdown of the spreading of illegal (pedophile or racist) contents.

It is important to notice that the law takes a wide definition of the notion of private data: it is any data that can lead to the identification of its owner. The IMEI of a phone can be considered as a way to identify the owner and any data derived from it is therefore considered as personal data.

Preventing the diffusion of malicious applications is more indirect. Spreading such an application is rather seen as an intrusion on a system that diverts it from its normal use and can undermine the integrity of data. The Convention against Cybercriminality [L'E 01], established by the European Council to harmonize the penalties for computer wrongdoings, is the most comprehensive text on the subject.

These texts have been translated into national legislation (in France for example, the crackdown of cybercriminality is based on sections 323-1 and 323-7 of the penal code) and national laws can be more restrictive: the French legislation also imposes some constraints on publicizing the cost of communications (consumer law) or restricts the creation of databases (the control is ensured by a commission known as the Cnil).

Finally, operators and content providers protect themselves against malicious developers through legal contracts that specify what are licit behaviors for an application in a much more precise way than what can be achieved through the security mechanisms of mobile terminals.

10.4. The implementation of a security policy

10.4.1. *Life-cycle of applications and implementation of the security policy*

When threats have been identified and when a security policy with a set of security objectives has been defined, it is now time to deploy a set of mechanisms to enforce the security policy and to check that they correctly fulfill their role. Confidence in a system is therefore the product of the coherence of the objectives (and of the policy), the quality of the mechanisms and the quality of the evaluation of these mechanisms.

Taking security into account, whether by implementing new mechanisms or by evaluating them, is an activity that spreads throughout the life-cycle of the system and cannot be taken in isolation:

– During the design, the definition of a security architecture with the identification of security components helps to prevent the introduction of vulnerabilities.

– During development and before deployment, removal techniques are used to reduce the mistakes introduced during development: code reviews, automatic analysis campaigns (see section 10.6.3.3 for some applications) or testing (see section 10.6.2).

– During the exploitation of the system, diagnosis tools and mitigation tools are used to survey the attacks and mitigate their impact: antivirus, device management and network supervision (see section 10.7.4).

With respect to downloadable applications, one particularity of the system is that it can be naturally divided in two parts: on one side, an infrastructure that can be considered as fixed comprises at least the mobile handset (its validation is mentioned in section 10.5.7), the network and the download platforms. On the other side we have downloadable applications that are not controlled by the main security actors (customers, network operators, phone manufacturers): in particular, they are not available before being deployed (their validation is the subject of section 10.6).

10.4.2. *Trusted computing base and reference monitors*

G. Nibaldi first defined the notion of the trusted computing base (TCB) [NIB 79] and the American evaluation criteria for the security of systems (TCSEC) [DOD 85] adopted it as their basis. It is a set of mechanisms that are responsible for the implementation of the security policy of the systems and the mechanisms that isolate enforcement mechanisms from the remainder of the system.

The TCB is the core of system security and ideally should be small and simple so that it can be formally verified. Unfortunately, in most cases, it comprises the full kernel of the operating system and all the programs that are executed with high privileges (the system administrator is usually the user).

The *reference monitor* [AND 72] is a part of the TCB that can resist attacks and that is in charge of controlling all access to data and peripherals.

10.4.3. *Distribution of security mechanisms*

It is possible to choose to implement all the security mechanisms on the handset using the techniques described in section 10.5. It is usually hard to enact strict generic rules for the use of dangerous APIs as some uses may be legitimate in a given context and dangerous for the security of user assets in another. Therefore, mechanisms implemented in terminals ultimately rely on the choice of the user who must decide whether or not he accepts the execution of a critical action. Some finer and more synthetic automatic analysis could be implemented on the phone only if more computing resources were available on the handset.

The alternative is to transfer the burden of controlling the security of applications out of the terminal, usually as a process implemented under the control of the operator. The goal of this is to guarantee the end-user that the application is safe and that it will warn him before attempting any action that he may refuse to do (such as actions that incur communication costs). Nowadays, the link that is used between the validation process and the handset to carry on confidence is the digital signature of the application (see section 10.6.4).

From the point of view of the TCB of the handset, this second solution reduces the size of trusted embedded code: validating a cryptographic signature is usually far easier than implementing controls and isolation mechanisms at the application level. However, this reduction is counter-balanced in the system as a whole by an external validation process that can be efficient only if the execution environment on the terminal ensures a few security properties on the execution of applications. In any case, the global TCB is in fact bigger in the case of an external validation of the applications.

10.5. Execution environments for active contents

A part of the software on the handset is in charge of managing third party applications (downloading, execution, resource sharing, disabling and removal of applications). However, the Application Management System (AMS) is also responsible for the protection of handset resources and user data against malicious applications and of isolating applications from each other. Depending on the hardware capabilities, various solutions can be implemented. This section will survey their principles. Finally we will quickly look at the architectures that protect content providers (DRM) and we will conclude with the validation of execution environments.

10.5.1. *The sandbox model*

A sandbox is a security mechanism that builds an execution environment isolated from the rest of the system so that we can safely execute an untrusted application in this closed world.

10.5.1.1. *Security objectives achieved by the sandbox*

The implementation of the security policy is mainly done by the handset. As far as downloadable applications are concerned, we must distinguish the three different categories of operations made by the execution environment and their link with security:

– downloading: the AMS assigns the permissions to access handset resources depending on the origin of the contents;

– activation: this controls the way the downloaded software can be started. This is important if the software can be run in the background;

– execution: the control of dangerous actions (such as transferring data) made by the applications during their execution.

The security policy of a mobile handset has specific features that distinguish it from that implemented on a multi-user computer system:

– there is usually one single identified user of the handset. The rules for isolating data are necessarily different from a global information system shared between different users with different rights. A security policy only based on the authentication of the user is meaningless as far as the security of downloaded applications is concerned;

– on this handset, there are usually a set of user assets whose use by a malicious actor is a really dangerous threat for the end-user: over-billing, identity theft, violations of privacy;

– the notion of delegation is central in such a system: the handset executes software provided by third parties for which the level of trust of the end-user can be very

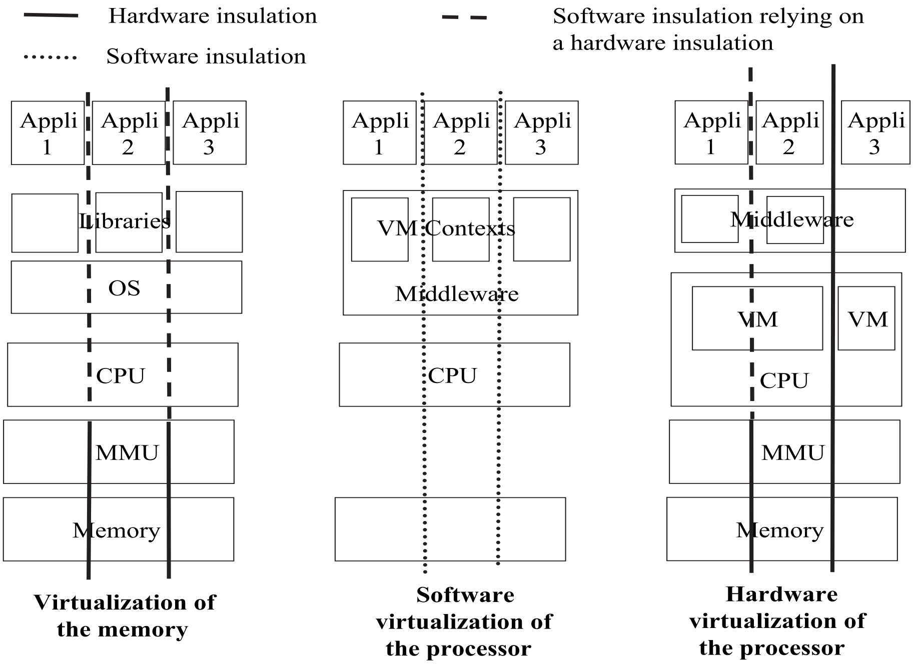

Hardware insulation
Software insulation
Software insulation relying on a hardware insulation
Appli 1
Appli 2
Appli 3
Libraries
OS
CPU
MMU
Memory
Virtualization of the memory
VM Contexts
Middleware
Software virtualization of the processor
VM
Hardware virtualization of the processor

Figure 10.1. *Virtualization techniques*

different. He may wish to delegate some activities to some trusted software agents that will need access rights to critical resources.

The security policy of a multi-user system has one primary goal: to isolate users and their data. Each user is then responsible for the use of his data and which applications may be used on this data. Installing a new application globally usually requires the highest security privileges.

On a mobile handset, the security policy must first control the level of delegation the user assigns to each of his applications. The purpose of this policy is different but the mechanisms implemented on mobile handsets are often directly inherited from multi-user computer systems. Symbian has introduced the notion of *capabilities* that can be viewed as a token representing the right to perform an action to address the problem of delegation.

10.5.1.2. *Virtualization*

Virtualization is a process that replaces a resource offered by the hardware to applications with a software equivalent. Virtualization mainly contributes to the objectives of isolating the TCB from the remainder of the system and of isolating the different software processes managed by the TCB.

Virtualization techniques can be classified into two broad categories depending on the kind of resource that is virtualized:

– virtualization of memory is the core technique used by multi-user operating systems and has been transposed in lighter forms on high-end mobile handsets;

– virtualization of the processor is usually a purely software technique. A bytecode-based virtual machine can simulate a secure processor that controls the execution of code. However on recent architectures, virtualization can be directly performed at the hardware level.

Virtualizing the processor with a virtual machine does not require any additional hardware component and is compatible with the capabilities of low-end and medium-end handsets (70% of the fleet by the end of 2008) whereas the use of a secure operating system based on virtualization of memory is still only available on high-end handsets (10% in 2008) because it uses hardware components that are not widespread for mobile handsets.

We will detail the advantages and disadvantages of these solutions in the following sections.

10.5.2. *Systems that do not control the execution of hosted software*

Some platforms offer very few security mechanisms during the execution of an application. This is the case for simple operating systems that do not offer a virtual machine or memory virtualization. Some old implementations of Symbian and the Brew platform offer unlimited access to memory to applications. Usually, such systems are only partially opened. Only well identified privileged third parties can install applications. For example, on a Brew system, only signed applications can be installed and the security of the platform relies completely on the verification of digital signatures and on the external validation of signed contents.

10.5.3. *Memory virtualization and open operating systems*

An operating system is said to be open if it can host third party applications. Each application is executed in one or several processes that virtualize computer resources. The security of the system relies on the virtualization of memory that isolates processes from each other.

Virtualizing memory requires hardware support and more specifically a Memory Management Unit (MMU) whose task is to translate virtual memory addresses used by the code of the process in real memory addresses that are mapped on the physical memory. It is then possible to create different segregated address spaces[2]. Each process has its own address range and is therefore isolated from other processes: it cannot read or write their data and its own data are protected. The MMU can also protect the access to peripherals and can restrict direct access to peripherals of the kernel that will delegate their management to administration processes. The management of the MMU itself must be confined to the kernel as it is the basis of the TCB.

The management of the file system also belongs to the TCB. In the case of a Unix operating system that abstracts every resource as a file, it can even be considered as the TCB of the system.

On a phone-like system, the security administrator, the system administrator and the user are confused. When there is an administrator role, it corresponds both to the management of rights and the management of programs. The consequence of this choice is that the TCB of the system extends not only to the kernel of the operating system, but also to all the system administration programs.

Mobile phones with an open operating system are usually *smartphones*, i.e. personal assistants integrating a mobile phone. Windows for mobile, Linux or the new iPhone operating system are lite versions of their desktop counterpart. The Symbian operating system follows very similar principles to these.

10.5.4. *Environment for bytecode execution and interpreters*

Another virtualization technique is to replace the instruction set of the processor with an abstract instruction set that is interpreted by a native program on the terminal: a virtual machine. As these instructions are often coded as small groups of bytes, they are called bytecode instructions.

Compared to the instruction set of a real processor, these instructions are usually more complex and are specialized for the compilation of a reduced set of high-level languages. This specialization is the key to reduce the execution time taken by their decoding. Therefore, paradoxically, this technique, which seems more expensive than the mere virtualization of memory, is in fact available on lower-end phones because it relies only on software mechanisms.

[2] Virtualization is not necessary to protect memory. However, it goes hand in hand with memory protection on almost every operating system because managing processes without virtualization means relocating code and is too complex. Nevertheless, some low-end handsets that do not integrate an MMU have an MPU (Memory Protection Unit) that lets them temporarily protect some memory banks.

In order to be exhaustive, we must also mention in this category interpreters that directly operate on the source program. Because interpretation is a very expensive process, languages that are usually interpreted (Shell scripts, perl, python, etc.) are not used much on mobile phones, but because of its ubiquity on the Web, more and more mobile phone browsers handle JavaScript. Most of the security measures developed for bytecode languages can be applied to interpreted languages.

One of the main characteristics of these abstract processors is usually that they naturally isolate the different control flows and data flows between the different applications but also between the applications and the host operating system.

This isolation relies on the strong typing of code: the verification of semantic rules that ensures that each instruction is executed on data compatible with its effect. Typing also implies syntactic correctness: only well-formed instructions can be executed and it is not possible to execute arbitrary data.

The type system of a programming language classifies the objects handled by programs in categories organized according to the meaning of these objects and their use in the program. Typing is the act of finding or verifying the object classification on a given program. It can be done during the execution of the program (dynamic typing) or before (static typing) usually at compile time but also when the program is loaded onto the phone.

When a language, in order to offer more flexibility to developers, lets the programmer bypass the restrictions of the type system, the type system is said to be weak. For example, the C programming language does not verify whether a type coercion is legitimate or not, either at compile time or execution time and it does not verify the use of pointer arithmetic (the use of arithmetic operations on pointers to compute new pointer addresses). In particular, it does not check that array bounds are respected. We will not study these languages further as they do not offer the security guarantees needed to isolate programs from each other.

Modern high-level languages complement the type system at the level of instructions with a visibility system at the level of components. Modules or classes group together a set of codes and data and their interface restrict the way other components can access data and the number of code entry points. Visibility is a corner stone of security in languages like Java where it is used to separate the code of the applications from the code of the system.

10.5.4.1. *Dynamically typed languages*

On mobile phones, this category mainly consists of object-oriented languages, variants of ECMAscript (ECMA 262 standard [ECM 99]). Nevertheless, although the specification defines a script language whose source code is interpreted by the execution environment, these languages, when used on the phone, are usually compiled into bytecode for efficiency reasons. The format of these bytecode languages is usually proprietary.

The execution environment must dynamically maintain for each object the value of its type (usually the representation of the object contains a field that points to a definition of the class of the object) and must check for each instruction that the arguments used are compatible with the action performed (checks are usually grouped for a given code block).

The TCB of such a system consists of the virtual machine (in particular, with the dynamic verification of types and the implementation of the rules controlling the visibility of data), critical libraries and access control mechanisms to these libraries (the high-level security policy implemented will be described later in section 10.5.4.3).

10.5.4.2. *Statically typed languages*

As indicated previously, type verifications are repetitive and therefore expensive in terms of computing time. The objective of static typing is to perform these operations once and for all before the execution of the program. Most often this operation occurs at compile time using the annotations given by the programmer on the type of variables. Unfortunately, when the program is executed, it is not possible to trust the result of typing performed by the developer's compiler.

It is also necessary to use a type system for the target language of the compiler: the bytecode. The role of bytecode verification is to ensure, at some point between the time where the program is loaded and the time where it is executed[3], that the bytecode is well-typed on the handset. At the same time, it also ensures the syntactic correctness of the bytecode. This verification is not a kind of dynamic typing because the program verified is not run.

The best known language in this category is Java. There exist several variants of the language adapted to the computing resources of the execution environments:

– Java Card [SUN 97] for smart cards (modern SIM cards integrate a Java Card environment);

[3] Depending on the execution environment, this verification can be performed when the program is downloaded or when the program is started or before the first use of a class (the compilation unit). Code optimization such as *just in time* compilation affect the timing where bytecode verification must be performed.

– Java Standard Edition (JavaSE), mainly deployed on desktop PCs and Java Enterprise Edition (JavaEE) for servers. JavaSE is also available on some high-end phones (Savage) and the Google version of the Java language for the Android platform is close to JavaSE;

– Java Mobile Edition (JavaME) with two variants:

- CLDC [JSR 00a, JSR 02b] (*Connected Limited Device Configuration*) for medium-end mobile phones. The reference implementation of the virtual machine (KVM) only takes a few tens of kilobytes of memory,

- CDC [JSR 01, JSR 05] (*Connected Device Configuration*) for personal digital assistants (PDA) and smartphones (PDA with an integrated mobile).

The CLDC variant is the most prevalent on mobile terminals with a set of libraries called MIDP [JSR 00b, JSR 02a]. There are other close variations of the CLDC technology on mobile phones, for example, Doja (NTT proprietary version) or mobile STIP [CON 04] (a standard from GlobalPlatform for secure banking services).

The main competitor is the Microsoft DotNet Compact Framework which is integrated into the latest versions of Windows Mobile and is the execution environment for downloaded applications. The environment is specified by the ECMA standard 335 [ECM 06] (Common Language Infrastructure – CLI).

Ensuring that the bytecode is well-typed can be expensive, especially if type information in the bytecode is rare. The problem is closer to type inference than type-checking and the algorithmic complexity is very different: inferring is exponential, whereas checking can be done in almost linear time. The solution found by Eva Rose [ROS 03] that has been chosen for the CLDC configuration is to perform some computations outside the terminal that add annotations guiding the verification process. If bad annotations are provided with the program, the type verification will simply fail. The TCB of such a system includes the virtual machine and the bytecode type checker incorporated into it. However, the preverifier that adds type annotation is not part of the TCB. The type verifier in the handset is both very critical and very complex and in the past, some implementations of this component presented dangerous vulnerabilities [GOW 04]. Another technique of attack against the VM is to modify the bytecode after typing, for example, causing memory errors by heating it [GOV 03], but this kind of technique cannot be used by a remote attacker.

As in the previous case, the TCB includes the most critical libraries and the mechanisms for controlling the access to these libraries.

10.5.4.3. *Implementation of high-level security policy*

Typing is a primitive that provides insulation between the various execution flows. Based on this primitive, it is possible to construct the mechanisms enforcing the access policy for the critical resources available to programs.

First, because it provides a classification of objects in programs and it controls the execution flow, typing can restrict the access to some data to certain parts of a program.

The semantics of the language sets the rules for the visibility of variables. For example, in Java, fields and methods of a compilation unit (a class) can be visible by either everybody (public), a group of classes (package) or only the class itself (private). The security policy is implemented on top of these mechanisms. Each call to a system method handling system resources is protected by a monitor that will, for example, ask the consent of the user before performing the operation. The code and the data of the monitor are part of the data that are not directly accessed or modified by the code of the downloaded application.

Naturally, checks are part of the TCB of the system, but also all the native libraries as the security of the system depends on the correctness of their code. Indeed, a vulnerability like a buffer overflow in native code can be exploited by the Java application [GOW 04].

10.5.4.4. *An example of security policy: MIDP2*

Because of its exemplary nature, we will present in more detail the choices made by MIDP, the most popular Java profile on mobile phones, whose standardization is the more open. Some elements are however not in the MIDP standard but in an appendix entitled “best practices”. The principles of MIDP security have been selected by the association of GSM operators as foundations for the security policy of downloaded applications regardless of the technology of the platform [GWA 05].

The policy defined by MIDP is based completely on the control of actions deemed to be dangerous. In particular, the policy of MIDP (like most existing dynamic policies) ignores the security problems of information flows and in particular the risk of leaking out private data.

The principle is the validation of dangerous actions by the end-user. For application providers, this policy has the advantage of basing the responsibility of the actions performed by the terminal on its owner even if he has not directly launched these actions. The limitations of such a solution are numerous:

– a policy based on the user introduces a new category of indirect risks into the system: that he takes the wrong decision for his own safety. Some attacks are aimed directly at increasing the likelihood of such an event, taking advantage of various psychological factors (social engineering attacks):

- the user may be overwhelmed by the number of messages or the repetition of messages may reduce his vigiliance,

- he has only a partial view of the process. He may then regret a permission he had to give before knowing the result of the transaction in progress;

– it is enforced independently on each terminal by each user. If the application is sufficiently widespread, it is statistically certain that some users will take the wrong decision for their safety. If every error leads to a financial gain for the malicious developer, this is a profitable strategy for the attacker.

The validation of actions by the end-user is legitimate and necessary but it is not always sufficient.

To implement this mechanism, the code of each dangerous method of the execution environment begins by displaying a screen that seeks permission from the end-user, and will trigger a security exception in case of refusal. The number of screens displayed can be very large, so this principle is tempered by three mechanisms acting on the permissions associated with each dangerous method:

– some questions are asked only once. MIDP defines three frequencies for asking: once for all for the application (*blanket*), once per execution of the application (*session*) or for each use (*oneshot*). Using a given method can be either completely forbidden, or authorized without verification, or authorized only after the user has given his consent, the consent being given with one of the above frequencies;

– regarding security, Java applications are organized in groups called security domains. The user can control, to some extent, the number of screens he is ready to view by configuring the rights attached to each security domain;

– by digitally signing an application, the operator, the manufacturer or a third party who has received a delegation from one of the previous two assigns to the application a given security domain that represents a confidence level. For a given confidence level and for each permission, the security policy of the Java platform defines a default value and a set of authorized values (normally, the user can modify the frequency of messages asking his agreement before performing an action).

Thus, security is based on the questions posed to the end-user and as their number may be too large, it may be necessary to develop strategies to reduce their number. Unfortunately, these strategies are not based on the knowledge of the software, but on a subjective confidence level given by the entity that must sign the software (usually the network operator is not the application provider). The validation of application code studied later in this chapter aims to provide factual elements to motivate the decision of the signatory.

Finally, the MIDP security policy does not address the security of information flows at all. The only exceptions are:

– the recommendation that the user must be alerted that setting to blanket (authorized without control) both the permission to capture multimedia streams (audio/video) and the permissions to access the network is dangerous;

– the prohibition to jointly set to blanket the permission for self-activation (known as push registry) and the permission to access the network.

10.5.5. *Evolution of hardware architectures*

A typical mobile handset consists of several processors: usually, the handling of telephony is performed by a specialized DSP separate from the core processor that handles application. On high-end phones, the telephony applicative layer is separate from the generic applicative layer and is performed by a separate processor. On such phones, there is sometimes a specific DSP for video and graphics.

Moreover, processors may have different execution modes with different privileges. Each mode can be seen as a virtual processor with its own interfaces and its own set of registers.

It is then very natural to consider the solution of using a hardware insulated execution space for the most critical applications. This execution environment usually handles a set of cryptographic keys with various functionalities:

– to ensure the integrity of the remainder of the operating system (this functionality is already available in boot-loaders for medium-end chipsets that start up in a privileged mode) and to check the origin of third party software;

– to authenticate the handset on the network and to guarantee to the remote site the integrity of the firmware of the phone;

– to encrypt in a safe manner data that are not currently in use.

Strong authentication of the handset and its user mainly benefits the other actors (service providers, content providers and network operators) as it guarantees the non-repudiation of transactions.

Technically the solutions developed for the desktop computer in the TCPA and TCG consortiums have been recently adapted to processors for the mobile industry (for example, the TrustZone technology for ARM processors). An execution environment such as a STIP platform can be used to execute the code of security critical services.

10.5.6. *Protecting the network and DRM solutions*

There are several DRM mechanisms, their common characteristic is probably that they have all been attacked and cracked. There is not a single safe DRM solution. The caretaking engineer will assume that none exists. The intrinsic brittleness of DRM technologies makes them vulnerable to the BOBE (Break Once, Break Everywhere) syndrome. In the Internet era, it is enough that one single hacker breaks the DRM of a content so that all Internet users have a way to circumvent this mechanism [BID 03].

One of the fundamental flaws of DRM is that it does not protect the user but it tries to protect the content provider against the user. As the user owns a legal copy of the protected content and the content reader, he has considerable resources to access the content in a way that does not follow the terms of the license granted by the owner of the intellectual property rights. Thus, in order to implement DRM, we must implement a trusted agent (that enforces the rights associated with the contents) on a malicious platform which is very different from defeating malicious software on a trusted platform. There is no known technical solution to the similar problem where the agent controls an electronic commerce transaction [CLA 03].

In the case of audio or video content, the signal must be played in clear at some point. It is always possible for the user to make a digital copy of the content either by using a high quality analog-to-digital converter or with a recorder cutting the flow just before the digital-to-analog conversion. So, even if there is no possible software attack to access to the unscrambled digital contents on the platform, this is not a guarantee that the content is safely protected on an open platform.

In the case where applications are considered as contents, the IP rights owner wants to protect their contents not only against illegal copies but also against retro-engineering of the software. Obfuscators are applications that modify the executable code of an application (bytecode or native code) to make retro-engineering more difficult. In practice, these mechanisms are not really efficient for fighting piracy and it has been proved that they are in fact not safe [APP 02]. The use of a technique known as steganography has been advocated to find the perpetrators of illegal copies. Steganography can mark contents in a so-called undetectable and inalienable way associated with the identity of the holder of the legitimate copy. In the presence of an illegal copy, the steganographic mark can be read in order to discover the identity of the author of the illegal copy. This approach faces several problems. First, technically, the mechanisms used for steganography are not safe, and marks can be removed or altered. Then, legally, the identity associated with the mark shows that the illegal copy is derived from the copy of the license holder, but it does not establish that this individual is the author of the unauthorized copying.

10.5.7. *Validation of execution environments*

We present some of the techniques implemented by operators to ensure that the mobile handsets they sell use execution environments for third party applications that are free from known vulnerabilities.

We will also look at the ambiguous role of fragmentation which is at the same time a source of cost for the development of new applications and for the validation of terminals, but also the best defense against the spreading of malicious applications.

The diversity of flaws makes the validation of handsets a difficult task. Some typical problems are as follows:

– flaws in the implementation of the kernel of the execution environment and in its security mechanisms. Fortunately, this kind of vulnerability is rare. See [GOW 04] for some publicly known examples;

– flaws in the interface between the virtual machine and the host computer (in particular, buffer overruns);

– flaws in the implementation of core services of the operating system or in components activated by the execution environment (communication protocols and codecs).

10.5.7.1. *Fragmentation of execution environments*

The fragmentation of execution environments can be defined as anything that prevents a given version of an application (a unique executable code) from working indifferently on all platforms. There are evident fragmentation points, such as the ones that distinguish the main execution platforms: MIDP, Doja, Windows Mobile, Brew or Symbian. However, there are also fragmentation points inside a given platform: for example, between the different versions of a given platform, when optional components are used, when the standard defining the platform authorizes some implementation options or because of implementation bugs. Finally, from the application point of view, there are also differences that, although they are not in the platform code, are perceived as fragmentation points; this is the case with the differences of screen sizes, differences of available memory, the availability of optional peripherals, or the configuration settings of the mobile network. All these fragmentation points make the development of a truly portable application that can be deployed on a wide variety of hardware platforms very difficult. It is usually necessary to create many slightly different versions of an application, sometimes even several versions for a given phone model. The fragmentation of mobile platforms is probably the main challenge facing the different actors (developers, operators, contents distributors). This is an important source of costs that inhibits many initiatives.

On the other hand, as far as security is concerned, fragmentation of mobile execution platforms is probably the best impediment to the spreading of a large number of malicious applications on mobile phones. Mobile phone platforms do not suffer from the software monoculture encountered in the desktop PC world that makes it vulnerable to a quick pandemic. This positive statement must be qualified: in fact, as the Java platform is available on more that 1.2 billion terminals (mid-2006 figure), it could be a software monoculture. Fortunately, the plurality of existing implementations (as authorized by the JCP) limits the scope of each bug, except for the bytecode verifier which plays a central role for the security of Java ME. This component had to be implemented using the source code from Sun[4]. It is precisely in this component that A. Gowdiak discovered not one but two bugs [GOW 04].

[4] This restriction has since then been lifted by SUN.

10.5.7.2. *Validation processes*

The proprietary platforms such as Windows Mobile, Symbian or Brew are not validated outside the internal qualification tests conducted by the providers of these systems. There are few specifications defining requirements for application execution platforms. 3GPP has ceased its activities in this area (MExE). The GSM Association (GSMA) has defined platform independent security requirements in the "Mobile Application Security" (MAS) working group. The "Open Mobile Terminal Platform" (OMTP) standard body resumed the work done in GSMA/MAS and pursued its development. Unfortunately, these specifications are not normative and are not accompanied by requirements and testing criteria. At the initiative of the network operators, mobile platforms could be certified by a process comparable to the "Type Approval" certification of PTCRB or GCF and initiatives have been launched in this direction.

The Java ME platforms, especially MIDP but also Doja to a lesser extent, are subject to a validation process defined by the Java Community Process (JCP), the body defining the Java specification. Java and Java programming interfaces are not standards, but are proprietary technologies like Windows Mobile and Symbian. However, unlike other proprietary platforms, the Java platform is not the intellectual property of a single company, as each specification (the virtual machine, the standard library, the main libraries) has a different owner. In the JCP, the company leading a working group (a Java specification request – JSR) on a Java technology acquires a monopoly for ensuring the compatibility with this technology. It must provide a test suite (technology compatibility kit – TCK) that any candidate must pass in order to comply. The TCB of a standard JavaME platform includes at least CLDC and MIDP. The TCK of these components are indirectly guarantees of the safety of an implementation of the JavaME platform.

10.6. Validation of active contents

In this section, we present the steps taken by network operators to verify applications before making them available to their customers on the portals they operate. Validation is an essential step if the operator must sign the application to give it the necessary rights to its implementation on the mobile terminal. First we present the whole process globally and then explain the two main techniques in use: testing and automatic code verification. Making a code review is another technique, but its prohibitive cost makes it unsuitable for industrial use.

Finally, we finish this section with a presentation of digital signature infrastructures that are used to provide differentiated and greater rights on the terminal to the applications that have been validated.

10.6.1. *Certification process for active contents*

10.6.1.1. *Organization*

The application certification process defines the mechanisms that can lead to the signature of an application. These processes define the roles of the different actors and the nature of the exchanges between these actors. There are several programs: JavaVerified and the Unified Testing Initiative for MIDP, Symbian Signed for Symbian, Mobile2Market for Windows Mobile and True Brew for Brew. The new players, Google and Apple, have also created their own process. We present the overall organization of these processes that is shared by all these players. Details may vary from one process to another.

The main roles are the following (see Figure 10.2):

– the certification body defines the roles and exchanges between actors, the requirements and guarantees associated with applications;

– the test laboratory is in charge of the analysis of an application and checks that it complies with the requirements defined by the certification body;

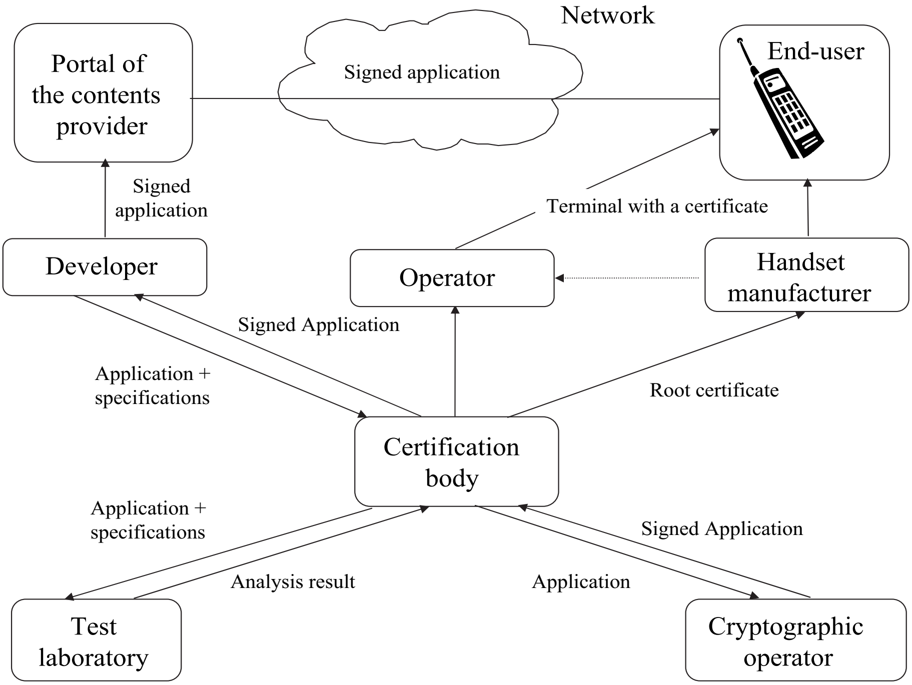


Figure 10.2. *Validation of application processes*

– the cryptographic operator holds the private keys used to perform the cryptographic operations: emitting certificates, signing an application and revoking the rights granted to a dangerous application;

– the developer provides the application;

– the handset manufacturer installs root certificates of the certification programs that are used to authenticate the origin of applications in the firmware of the handset.

The role of each actor is best described by the dynamics of the certification process. Several scenarios and phases are possible.

Program set-up

– The certification body authorizes a cryptographic operator and one or more test laboratories. The cryptographic operators provides the cryptographic root certificate to the manufacturer either directly or via the certification body or the mobile operator.

– The manufacturer installs the root certificate in the handsets he produces; in theory it is written in a non-rewritable memory.

Developer registration

– The developer provides the information proving his identity to the operator of the certification program.

– The operator of the certification program and the developer are bound by a contract. The operator of the certification program and the laboratory test will have access to the application source code. The operator of the certification program sends the developer's commitment to the cryptographic operator who stores information on the developer's identity and sometimes issues a certificate for the developer.

Certification of an application

– The developer provides an application to the operator of the certification program for transmission to the laboratory. The test laboratory analyzes the application and provides a report forwarded to the developer and the operator of the certification process.

– If the report indicates that the application meets the requirements of the operator certification, the certification body authorizes the cryptographic operator to digitally sign the application on his behalf.

– The cryptographic operator issues a certificate specific to the application. The cryptographic operator signs the application and issues a certificate. The certificate is authenticated by the root certificate of the certification body.

Installing a certified application

– The user of the mobile obtains a signed application and tries to install it.

– The mobile checks the signature of the application: if the signature is valid, the application is installed and enjoys the status of being a privileged application. Several different levels of privileges can co-exist, depending on the root certificate used to authenticate the application.

Application revocation

– The operator of the certification programs discovers that one of the signed application is malicious. The information is forwarded to the cryptographic operator.

– The cryptographic operator revokes the certificate used to sign the application and broadcasts the revocation information.

– Mobile handset informed of the application revocation take defensive counter measures:

- by uninstalling and deactivating the application, with or without the agreement of the end-user,
- by degrading the application privileges.

The processes described above suffer from several weaknesses:

– the developer, whether he is an individual or a company, may lie about his identity (false identity or fictitious company);

– the possibility of legal recourse is low in the case where a malicious application has been signed because of the international nature of these programs operated over the Internet;

– the requirements may be insufficient, unverifiable or contradictory;

– the integrity of the root certificate on the handset must be preserved. There are effective attacks against these certificates. Although the MIDP standard requires that the installation of new certificates is impossible, this requirement may not be fulfilled by every handset;

– although there are standard procedures for revoking a certificate (OCSP and CRL), this operation is a source of concern: who should pay for the messages exchanged to handle the revocation process? Should the buyer of the content be reimbursed?

10.6.1.2. *Contents of the assessments and guarantees they provide*

Evaluations usually rely on three elements:

– the review of the code by the laboratory;

– the test of the application by the laboratory;

– the statements of the developer.

These elements are not sufficient to associate strong guarantees with signed applications.

The certification process has mainly a deterrent effect on ill-intentioned hobbyist developers. As a professional, it is easy for a developer to guard against the risk of breach of contract to which he is exposed by having a malicious application signed. Note that these means are mainly to falsify or conceal the identity of an individual or a corporation, which may be relatively easy, even for an individual, especially in countries where the risk of prosecution is very low.

10.6.1.3. *The impact of execution environment fragmentation on the validation of applications*

The fragmentation of the market for downloadable applications leads to a multiplication of the number of versions per application. The developers often produce dozens and sometimes hundreds of versions (indeed, they must adapt their code to specific terminals, internationalize the dialogs, and to a lesser extent, adapt to specific networks). In general, test laboratories charge a cost equivalent to one engineer week for each version of an application. A different signature must be issued for each version and the cryptographic operator will also charge each of these. If a significant number of versions must be signed, the overall cost of certification of an application is prohibitive.

10.6.2. *Application testing*

Testing is the most widely used method to evaluate applications. While there are other techniques available for applications written in Java, it is in any case the only applicable method for native applications. The conditions for performing testing are generally not very good. The resources available for testing are very limited. To limit the cost for an application used worldwide, operators and manufacturers are trying to develop joint programs, for example, JavaVerified [UNI 04] for Java, that deliver a valid certificate for all operators of the consortium. Testing is done in black box: the code is not always available and, most often, neither are the specifications. However, test criteria, such as JavaVerified [UNI 05], may make it necessary to supply some documents for the submission of an application, such as a user documentation and at least a schema representing the different screens of the application and transitions between these screens. Finally, a test is not particularly suitable for security analysis. The test is based on an assumption of uniform distribution of errors, which provides a theoretical basis for the concept of coverage. Unfortunately, properly hidden malicious code does not meet this criterion. Also, the main goal of a test campaign is to verify the functional properties of the application and its compatibility with the terminal.

Testing must be performed on the real device for which the assessment is performed. Indeed, fragmentation among different implementations is too high to ensure confidence in the portability of applications. In particular, emulators cannot be trusted. A serious evaluation should reproduce the conditions for using the application, particularly the network operator which enforces its own constraints and peculiarities (naming of entities, choice of protocols, timing requirements, etc.).

Thus, testing is mostly manual. The operator must download the application on the terminal and perform all the operations corresponding to the test suite. He will probably need special cheat codes to quickly reach the different screens of a game. There are commercial solutions to drive mobile terminals from a desktop computer. These solutions are based on the use of modified devices. If the behavior of the application is sufficiently predictable, it is possible to produce systematic test cases corresponding to various conditions. Such a robotic system is very expensive to operate and keep up to date. It is usually shared by various test houses and software development companies.

10.6.3. ***Automatic analysis techniques***

10.6.3.1. *Automatic validation techniques*

An ideal validation campaign should take into account all possible states of a system. If a comprehensive test campaign is not achievable, against a given fixed property, it is often possible to simplify the system analyzed to take into account only the actions related to the compliance with the property and to decide on this simpler model whether the property is respected or not. This is the principle of automatic validation techniques that can be classified into three broad categories: automatic deduction techniques, model-checking and static analysis. In practice, static analysis techniques are these that give the best results. It should be noted that Java bytecode verification is an example of static analysis.

10.6.3.1.1. Automatic deductive proofs

The system and the property to be proved are modeled in a logical framework. We can then formalize the set of reachable states (usually as an inductive constraint starting from the predicate characterizing the initial statements) and study the properties of this set of states. Heuristics can guide the search for evidence (decision-making procedures), but these tools also use code annotations. JML [LEA 99, LEA 04] is the de facto standard for these annotations in Java code and [BUR 03] presents an overview of existing tools using JML. The expressiveness of the logics used is almost unlimited and the most complex behaviors can be modeled, but these tools can only handle simple examples at the moment because of the complexity and size of each model. Therefore, these techniques are mostly used to verify small JavaCard applications with high security requirements.

10.6.3.1.2. Model-checking

The principle of software model-checking is to abstract the system state to ideally obtain a finite number of states so that the behavior of the system can be exhaustively explored. There is a corresponding abstraction of the behavior of the program that will now operate on abstract states and not on the actual states. The exploration of the state space of the program is performed using the best techniques for sharing information such as Binary Decision Diagrams (BDD) to avoid recomputing already encountered execution paths. The risk is that, despite the abstraction, the number of states to explore grows beyond the resources allocated to verification. Model-checking gives good results for the analysis of protocols or synchronization problems [COR 00, VIS 03] but in practice it is not used for security analysis because it is immediately confronted with the problem of combinatorial explosion on the data structures manipulated.

10.6.3.1.3. Static analysis

The basic idea of static analysis is quite similar to model-checking but first, we build, from the syntactic structure of the code, relations between the sets of possible states for the execution of each statement (control flow analysis) or relations between variables and their respective content (data flow analysis). The result is a system of equations between the sets of statements or variables. If some simple mathematical properties hold on the domains used by the equations, the existence of a solution to these systems is guaranteed [TAR 55] and there are efficient algorithms for finding them [KIL 73]. For more details, the reader can read the introductory book from F. and H. R. Nielson and C. Hankin [NIE 99], which describes these techniques in great detail and presents the different techniques of analysis and the theory of abstract interpretation [COU 77] used as the underlying unifying formal model.

10.6.3.2. *Advantages and limits of static analysis*

The main advantage is the complete automation of the process. In a few seconds, without human intervention, a property of an application can be proved and a positive result is a guarantee that the property holds. On the other hand, a given analysis only has a limited scope and can only check a given kind of properties. It cannot find arbitrary problems in the way that testing can.

The first limit of static analysis is that it only applies to languages that effectively limit by construction the possible control-flows and data-flows. Static analysis is a technique that is only applicable on strongly typed high-level languages or strongly typed bytecode (at least for security properties).

Indeed, it will not give satisfactory results on arbitrary machine code (but see [XU 00] for a counter-example). As the evaluators generally only have access to compiled code, in practice this limits the scope of this technique to downloadable software written in Java or C#. Finally, even when the source code is available, analysis of code written in weakly typed language such as C is limited to finding bugs (a kind of symbolic test) and cannot provide strong guarantees.

Even in a limited framework like Java, all properties are not verifiable and some language constructs pose daunting problems, in particular the use of multi-threading and memory shared between different threads. The memory model is the specification of the semantics of operations writing and reading in memory. It defines the values that can be read by a thread for a variable that is modified by one or more threads. The Java memory model has been designed to facilitate the optimization of the code by the compiler and the use of hardware acceleration techniques (memory caching) at the expense of the simplicity of the semantics of language. In these conditions, practical analyses that scale to real world programs are those that do not rely on the program control-flow (or only locally inside a method body).

Finally, like any technique based on approximation, static analysis may be unable to conclude in some cases. It may be useful to impose some restrictions on the programming style to ensure that correct analyzed programs will pass the analysis.

10.6.3.3. *Controlling the use of dangerous functionalities*

We have seen that the MIDP security policy was designed to monitor the use of critical features by requiring the user agreement before their use. However, to limit the number of such screens, the operator could sign the application. The question raised is whether the operator can safely remove the control screens without creating a security risk to the user. To solve it, the evaluation must answer the following questions:

– What are the critical features used by the application?

– What are the parameters of the calls when these features are used (recipient's phone number of SMS for example)?

– How many times are these features used and in what context?

Finding the classes used by a program is very simple. This feature is built into most download portals to avoid the presence of proprietary APIs in the code of applications. Controlling the methods used is somewhat more complex because in object oriented-oriented languages, virtual method calls must be dynamically translated in actual target methods. The call to a method of an object carried by a variable refers to a method defined in the class of the variable. At runtime, it must be translated into a method of the real object used whose class may be a subclass of the class of the variable. This method can be implemented in different ways. Devirtualization is a static analysis that computes the possible outcomes of the dynamic resolution that takes place at each call to a method of a variable. [GRO 97] presents an old overview of some of these techniques. Its reading can be supplemented by the following references [MIL 05, SUN 00].

In practice, the parameters of critical calls to analyze are mostly strings (URLs used to open connections or types of medium handled) and more rarely integer constants corresponding to an enumeration. It is possible to build static analysis [CHR 03, CRÉ 05, LIV 05] approximating the values of strings. This analysis takes advantage of the fact that strings are not immutable in Java and that the only important operation is the concatenation of strings performed via the use of the StringBuffer class.

Counting the number of uses of a given method is more complex; several works address the problem of counting method calls on a single user interaction [BES 06a, CAC 05]. It is necessary to have techniques to build back the global flow of control defined by the GUI of the application in order to count the number of uses of an API on an application transaction that spans over several user interactions [CRÉ 07].

10.6.4. *Signing contents*

Signing an application means putting a distinctive sign on the application that is recognized by the terminal as a mark that a trusted third party has confidence in the executable content of this application and that the terminal can safely release its security policy regarding this content. The signing of applications transfers the burden of monitoring the application on the terminal to a third party validation scheme. Thus, the TCB based on the mobile terminal is reduced to a system of verification of signatures but at the cost of extending the TCB to the whole validation process that generally consists of manual processes in addition to automated processes.

10.6.4.1. *Principle of public key cryptography*

Public key cryptography is based on the use of asymmetric cryptographic algorithms using two separate keys for encryption and decryption of messages. One of the keys (private) will be kept secret while the other (public) is broadcast. The signing of a document is the encryption of a cryptographic hash of the document with the secret key. Checking is done by deciphering the encrypted result with the public key to verify that the cryptographic hash corresponds to the document. It establishes the identity of the owner of the secret key.

These algorithms are used in a public key infrastructure (PKI) which enables a certification authority to register and certify the identity of users of this infrastructure. The authority issues certificates, i.e. documents signed with the private key that establish a link between a public key and a user. The aim is to limit the number of public keys residing on the system to check the source of the documents to the key belonging to the trusted third party and not to all the keys of users of the system. The root certificate is the self-signed certificate describing the public key of the authority of confidence and it is prerecorded on the phone.

The PKI infrastructure must also allow the revocation of certificates awarded to entities that are no longer safe or whose private key has been disclosed (this is done using revocation lists) and delegate the right to certify keys of other entities (using certificate chains).

10.6.4.2. *Implementation of PKI*

The execution platforms can use a mechanism for checking digital signatures to authenticate the identity of the person signing the application. The identity of the person signing the application indirectly authenticates the author of the application. The identity of the author of an application can be used in two ways:

– depending on the identity of the owner of the root certificate used for the authentication, it is possible to offer various levels of privileges to the application;

– comparing the identities of the authors of two applications, the system may decide to restrict inter-application communications to applications with the same author.

10.7. Detection of attacks

In this section, we present the different techniques that can be used to detect and block malicious applications.

10.7.1. *Malicious application propagation*

10.7.1.1. *Classification*

Malicious applications can be classified into three main groups:

– viruses: a virus is an application that replicates by inserting itself into executable files or others documents stored on the infected system. It requires an external action to propagate [SZO 05];

– worms: a worm is an application which replicates and propagates by using the network connectivity. As opposed to a virus, a worm does not need a host file to replicate itself [WIK 06];

– Trojan horses: a Trojan horse is an application that appears to be legitimate but that performs undisclosed malicious functions. It does not have the ability to replicate [WIK 06].

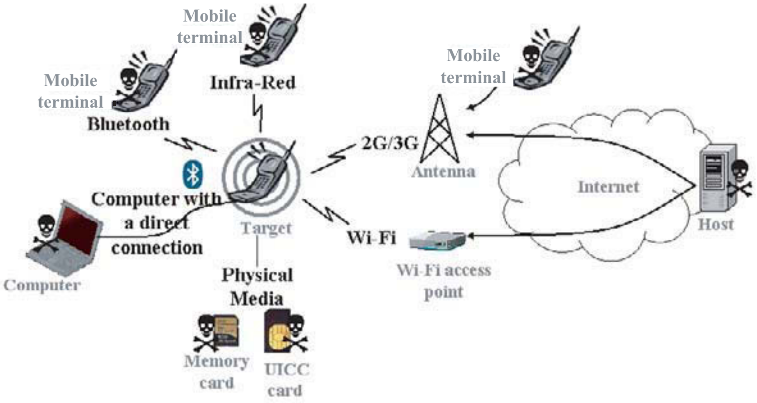


Figure 10.3. *Main infection vectors for mobile phones*

10.7.1.2. *Infection mechanisms*

Infection mechanisms are multiple for a mobile phone; Figure 10.3 presents the main ones:

– direct vectors where the infection source is physically connected to mobile phones:

- insertion of an infected memory or UICC card,

- connection of the mobile phone to a computer which downloads a malicious application;

– short distance vectors where the infection source is in close proximity to the mobile phone:

- Bluetooth connection with an infected mobile phone,

- infrared connection with an infected mobile phone;

– long distance vectors where the infection source is at an arbitrary distance of the mobile phone:

- reception of infected messages (e-mail, SMS, MMS, etc.),

- downloading of infected content (for example, a game) on the Internet via an HS(CSD), GPRS, EDGE, UMTS, HSDPA or Wi-Fi.

10.7.2. *Monitoring*

When a new malicious application appears, the shorter the time to react, the better the chances of stopping its propagation and limiting the potential damage it could cause. Monitoring is as a consequence essential.

10.7.2.1. *New malicious application detection*

10.7.2.1.1. Victim complaints

The infection of a mobile phone by a malicious application can have consequences that induce the user to call his operator or manufacturer customer care to complain. An increase of the number of complaints describing problems such as those mentioned in section 10.3.4 can indicate that a malicious application is currently being propagated.

10.7.2.1.2. Monitoring the content available on the Internet

Security experts monitor the content available on the Internet (forums, websites, etc.) for the purpose of:

– discovering, at the same time of malicious applications writers, new methods and vulnerabilities;

– obtaining versions of these applications before a massive propagation.

10.7.2.1.3. Monitoring the traffic in operator networks

Several products are at the disposal of operators in order to monitor the appearance of new malicious applications when they are propagating on their networks:

– network antivirus;

– fraud management systems [MAT 03];

– firewalls [ZWI 00];

– intrusion prevention and detection systems [END 03];

– antispam systems [ZDZ 05].

Network antiviruses are described in section 10.7.3.3.

10.7.2.2. *Alert*

When a new malicious application is found, it is sent to antivirus companies for analysis in order for them, as quickly as possible, to update their antivirus product and produce an alert that will be displayed on their website. These alerts contain a description of the behavior of the application, the available methods to disinfect a terminal and an evaluation of the risk.

The GSM Association (GSMA), which is an international organization whose members are the GSM operators, provides a forum allowing its members to rapidly share information about a new malicious application.

10.7.3. *Antivirus*

With the first malicious applications targeting mobile phones, antivirus solutions have appeared on the market in order to offer the possibility of protecting users against these type of attacks.

Such solutions can be divided into two main categories: antivirus developed specifically for mobile phones and antivirus deployed in operator networks.

Before presenting the features of these two solutions, we will study the general antiviral methods of detection and eradication that can be implemented in an antivirus product.

10.7.3.1. *Antiviral methods*

10.7.3.1.1. Detection methods

Detection methods can be divided into three main groups: files scanners, integrity verification and behavior analysis.

File scanners

A files scanner scans the content of a file to identify the presence of malicious code [LIN 06, SZO 05]. It should be able to analyze different types of file formats and decompress archived files (.zip, .rar, etc.). Files scanners can be based on different methods. We present the main methods below.

Signature scanning

This method is the one used most often, as it makes it possible to detect the presence of a malicious code in a file without executing it. It consists of verifying whether the analyzed file contains the signature of a malicious application. A signature is a sequence of bytes extracted from the executable code of a malicious application and identifying it uniquely. Some signatures are defined in such a way that they could identify several variants of the same malicious application. The purpose of this type of signature is to be able to detect new minor variants of a malicious code before a signature which would be specific to them is established.

A scanner based on signature scanning must have access to a database which contains all the signatures of known malicious applications. This database must be updated regularly, otherwise the scanner will become useless. In order to optimize the time required to analyze a file, some antivirus associate with the signature some additional information such as the format of the file that can be infected or, when known, the position where the signature should be found in the file [MUT 00]. The speed of the analysis of a file also depends on the algorithm used to look for the signature [BOY 77, COR 01].

This method makes it possible to detect known malicious applications and possibly new variants of these applications. It generates a low number of false alarms. However, it is not efficient to detect unknown malicious applications or polymorph applications whose appearance changes from one infection to the other while retaining the same behavior.

Code emulation

This method makes it possible to detect polymorph viruses which encrypt themselves and modify the decrypting routine at each replication [LIN 06, NAC 96, SZO 05].

In order to be able to detect such viruses, antivirus executes the file to analyze in a secure virtual environment in the antivirus. During its execution, the virus decrypts itself and exposes its code to the antivirus that can then use the signature scanning method. This method is relatively slow and by consequence is not systematically used. It is also ineffective against unknown malicious applications.

Heuristic analysis

The purpose of heuristic analysis is to detect unknown malicious applications. There are two types of heuristic analysis:

– Static heuristic analysis which looks in the application code for instructions frequently used in malicious applications. When a suspicious instruction is detected, it increases a counter. If the value of the counter exceeds a predefined threshold, the file is considered as suspect.

– Dynamic heuristic analysis which executes the file to analyze in a secure virtual environment, as in the case of the code emulation method. This makes it possible to detect if actions frequently implemented by malicious applications are executed, such as the search for executable files.

This method can detect new malicious applications, but it also generates false alarms.

Integrity verification

This is a reactive method that can only detect the presence of a malicious application in a network after it has started to propagate [SZO 05]. This method is based on the principle that any modification of an executable file is suspect and may indicate the presence of a malicious application in the system. The method consists of calculating, for every executable file stored on the system, a checksum that will be stored and will be used to verify the integrity of the system. These checksums are in general calculated thanks to cryptographic hashing functions in order to prevent malicious applications from finding a modification that will conserve the same checksum. If an executable file is modified, the value of checksum will change. The antivirus could generate an alarm to inform the user.

This method must be able to distinguish legitimate modification, for example, when a file is recompiled or when an update is performed. It generates many false alarms.

Behavior analysis

The method requires that the antivirus runs in the background in order to monitor the behavior of files currently executing on the system [LIN 06, NAC 02].

When an application tries to execute a suspicious action such as opening an executable file and modifying it, this action will be intercepted and the antivirus will either close the application or ask the user which option he would like to take. However, for most users this is a difficult choice, even more so as the number of false alarms is high.

10.7.3.1.2. Malicious application eradication

When an infected file is detected, the user can in general choose from four options to handle the file:

– cleaning: the malicious content is removed from the file without altering it. Depending on the type of infection and file, this option is not always available. In particular, it requires that the malicious application has been identified with precision;

– quarantine: the infected file is stored in a directory where it cannot execute;

– deletion: the infected file is deleted from the system;

– released: the file is released and the user can execute it.

10.7.3.2. *Mobile phones antivirus*

To protect against malicious applications, mobile phone users can install an antivirus on their mobiles.

10.7.3.2.1. Overview

Detection

Most mobile phone antivirus software available today is based on the signature scanning method (see section 10.7.3.1.1). They make it possible to detect known malicious applications independently of the infected mechanism used. They can work in two different modes:

– real-time: every time a file is introduced in a mobile or opened to be read, modified or executed, it is analyzed by the antivirus which is resident in the memory;

– on-demand: the antivirus is not resident in the memory and its execution can be manual or planned by the user. In this mode, the antivirus analyzes all the files stored in the mobile phone.

These antiviruses are usually able to detect viruses that are resident in the memory by performing a signature scanning in the Random Access Memory (RAM).

In general, the signature database is a file stored in the mobile.

Updates

Currently, there are two methods of performing antivirus updates to the signature database:

– the first method requires an active connection to the Internet in order for antivirus to download the last signatures of malicious applications. Antivirus can be configured to perform this type of update at regular time intervals under the condition that a connection is available;

– the second method uses SMS messages. The updates are contained in the SMS and no downloading is required.

It is also necessary to update the software. These updates are used to install new components, apply patches or modify the configuration parameters. They make it possible to improve the system performance or to correct vulnerabilities. As these updates may be large, the user can chose whether he would like to download them.

In order to not introduce new risks, the antivirus must verify the integrity and the source of these updates. Ideally all updates exchanged with the server and the antivirus must be signed.

10.7.3.2.2. Limitations

Impact on system performance

Mobile phones have limited computational power, storage and battery life capacities. Antivirus products must adapt to these constraints. Due to the low number of malicious applications, it is difficult to evaluate the impact of current product on the system performance. If this number should drastically increase, it may become necessary to limit the number of stored signatures in the database by eliminating for example the oldest. This would limit the quantity of memory required for the antivirus and the time to scan a file. However, the user would no longer be protected against old malicious applications.

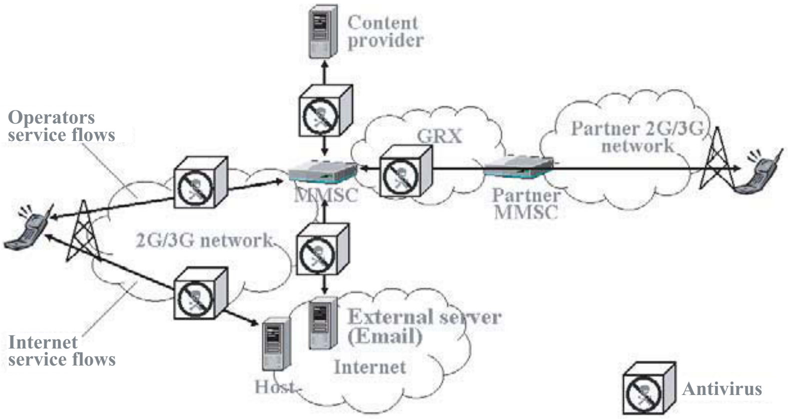


Figure 10.4. *Data flows that can carry infected contents*

Moreover, these constraints make it difficult to implement additional detection methods such as code emulation, heuristic analysis, integrity verification or behavior analysis. These methods tend to protect the users against unknown malicious applications and polymorph applications while the signature scanning method mainly protects them against known malicious applications.

However, the computational power and storage capacities of mobile phones should increase constantly and antivirus products should be able to support more sophisticated methods in the future.

Platform fragmentation

An antivirus can be executed only on the execution environment for which it has been developed, such as most malicious applications. Today there is not an antivirus for every execution environment that can be targeted by a malicious application.

Installation and update management

The installation, the configuration of antivirus and the management of updates are not always performed correctly by the user. Some operators and manufactures propose some mobile phones with an antivirus pre-installed.

User education

Most mobile phone users do not realize the risk linked to the installation of applications on their mobile phones, and as a consequence very few of them install an antivirus.

10.7.3.3. *Network antivirus*

Mobile operators have the possibility of analyzing the content transmitted on their network in order to detect and block malicious applications before they infect the mobile phones of their customers.

10.7.3.3.1. Overview

Traffic analysis

Figure 10.4 shows the main data flow that can carry infected content in the network of a mobile operator to date.

These data flows can be divided in two groups:

– service operator flows, such as these carrying MMS message traffic (see Figure 10.4), traffic based on the IP Multimedia Subsystem or others;

– service Internet flows. These are mainly composed of content downloaded on webpages or WAP, transferred by FTP or exchange by e-mail, instant messenger or as a result of peer-to-peer applications.

Detection

The traffic flows are intercepted by the antiviruses that reconstruct the carried files from data packets. The antivirus must know the protocol used to transmit them to their destinations; the main ones are HTTP, FTP, SMTP, POP, IMAP, instant messenger protocols and peer-to-peer protocols.

Like their counterpart for mobile phones, network antiviruses are mainly based on the signature scanning method in order to detect malicious applications. However, some of them implement some complementary methods such as code emulation or heuristic analysis, allowing them to detect malicious applications that are still unknown (see section 10.7.3.1.1).

These antiviruses can detect users who have sent infected content on the network and can inform them automatically by sending a message whose content can be configured by the operator. In general the message contains some information on the process to follow to disinfect their mobile phones. It is possible to attach to this type of notification an application allowing the user to disinfect their mobile phones; however, this solution presents a risk. Indeed, malicious individuals could send messages that look like this type of notification but whose attachment is a malicious application.

10.7.3.3.2. Limitations

Impact on network performances

File reconstruction and its analysis can introduce a delay in its transmission to the destination.

Infection vectors

Network antivirus cannot protect users against malicious applications that infect mobile phones via short distance connections (Bluetooth, infrared), Wi-Fi or direct connection such as the synchronization with a computer or the insertion of an infected memory card.

Unknown malicious applications

Network antivirus can detect some unknown malicious applications but does not ensure protection against all of them.

10.7.4. *Remote device management*

10.7.4.1. *Overview*

Device management is the generic term given to all technologies allowing third parties (mobile operators, service providers or companies) to remotely manage terminals for the users. Remote device management is performed as a result of a server controlled by a third party, which communicates with the client application installed in each terminal.

The Open Mobile Alliance (OMA) standardization group has published some specifications which define protocols and mechanisms required for device management in the OMA device management standard [OMA 06b] which is based on the SyncML standard [OMA 05]. These specifications are independent of the types of connections used between the client and the server, it can be short distance connections (Bluetooth, Infrared), via a fixed network or wireless.

We present below the main functions of the device management:

– initial device provisioning: specific network and user preference parameter configuration;

– new services activation: automatic configuration of services parameters;

– application lifecycle management: installation, updating and automatic configuration of applications;

– firmware management: updating and automatic configuration of the firmware [OMA 06c];

– diagnosis and repair: in case of a malfunctioning device, the customer care department can remotely perform a diagnosis and repair the device when possible. The diagnosis makes it possible to determine the device model, the operating system version, the configuration parameters or the installed applications.

The standard [OMA 06a] defines the specifications making it possible to protect the integrity and confidentiality of data exchanged between the server and the client application. It also defines the required mechanisms for the mutual authentication between the server and the client application.

10.7.4.2. *Remote protection of users*

This section presents the different potential uses of device management by mobile operators in order to remotely protect their customers against malicious applications.

10.7.4.2.1. Automatic installation of firmware updates and other applications

When a security vulnerability is detected in the firmware or any other applications, device management makes it possible to identify the vulnerable devices and perform automatic updates for the concerned application.

10.7.4.2.2. Removing malicious application

When a user contacts his customer care department to complain that his device is not working properly, the customer care department can use device management to verify if the device is infected by a malicious application and remotely remove it if it is the case. Infected mobile phones could also be detected if an antivirus is installed on the network (see section 10.7.3.3.1) and if this antivirus can communicate to the DM server the list of infected customers.

10.7.4.2.3. Network interface configuration

If a malicious application propagates using a specific network interface (Bluetooth, infrared, GSM, GPRS, EDGE, UMTS, HSDPA or Wi-Fi), thanks to device management, the operator can decide to deactivate the concerned interface temporarily or permanently. For example, in the case of Bluetooth, the operator can decide to configure it so that it goes into non-visible mode a few minutes after the user has paired with another device.

10.7.4.2.4. Modification of the security model

As indicated in section 10.5.4.4, the security model defines access rights of applications installed on a device depending on their origin. To protect the user against malicious applications, it could be necessary to use device management to change these access rights.

10.7.4.2.5. Mobile antivirus management

Installation, configuration and update management of mobile antivirus are not always performed correctly by users; device management can be used to perform these tasks.

10.8. Conclusion

10.8.1. *Research directions*

10.8.1.1. *Harmonizing security policies*

As explained in this chapter, various technologies (open operating systems, virtual machines) can be used to execute active contents. Each technology has a different security policy or at least a specific implementation. This heterogenity is very confusing for the user, especially if these execution platforms coexist on the same terminal. A goal of operators is the definition and implementation of a unified security policy (for consistency) that is also shared among operators (to limit the cost of validation).

10.8.1.2. *The reduction of security warning screens*

We have seen that too many security screens could distract the user and ultimately facilitate attacks. It is desirable to allow grouped permissions and to impose a limited number of authorization screens per application. Some preliminary solutions for Java have already been proposed [BES 06a]. They can be seen as a generalization of very fine grained capabilities inside applications. Static analysis techniques are used to verify the consistency between the applications and rights granted by the user.

10.8.1.3. *Monitoring the information flow*

As noted above, the legislation focuses on data protection, whereas implemented security policies focus on the actions performed by programs. When an application accesses the address-book of the user, he is alerted by a security screen. However, the right question is not whether the application accesses the phone book but rather the use of the data extracted from it:

– to compose a letter to the user's request;

– to send these data to a malicious actor which can use them to spy on the user or to send spam or spread a virus.

In order to answer this question, it is necessary to monitor the use of confidential data in the program. Such analysis is an analysis of the flow of information. The problem of the analysis of the flow of information is that it depends on all possible executions and not simply on the ongoing execution. For example, in the following code:

```
y = z; if (x == 0) y = 1;
```

The value of y of course depends on that of z but also on x even if x is for example 1. Yet in this case, y is not changed as a result of the test of x. The flow between x and y is called an implicit flow and the flow between z and y is explicit. Finally, it is possible to surreptitiously pass information between two execution threads using temporal information. We then say that there is a hidden channel between the two threads.

Static analysis is a much more promising technique than dynamic monitoring to check the properties of confinement on the flow of information. However, it is still the subject of research and results (only available for Java and bytecode languages) are not yet exploitable by operators.

Readers may consult the work of Andrew Myers [MYE 99] that presents a solution at the level of the source language, the following survey of potential solutions [SAB 03] or consult the deliverables of the European Mobius project [MOB 05].

10.8.1.4. *Conveying confidence*

Section 10.6.4.1 has shown how to use PKI solutions to enable a trusted third party to communicate its favorable ruling on an application to the terminal client. However, we would ground this trust not on guarantees provided by a third party validator who does not have a complete knowledge of the code, but on guarantees on the intrinsic quality of the code made by the developer himself. Proof-carrying code (PCC) is a technique to combine executable code with properties on this code and proofs of these properties. The issuer has to find proofs of the relevant property (with varying degrees of automation depending on the technique used and the complexity of the property) and to send to the recipient just enough information to enable him to verify this property. If the PCC applications to the transmission of security guarantees are still mainly a research topic, it should be noted that the bytecode verifier on a CLDC virtual machine already uses this principle. Indeed, standard bytecode verification is an expensive process because the verifier must infer a lot of information and its cost is beyond the reach of low-end terminals. The operation has been divided into two parts:

– pre-verification conducted by the developer that adds annotations in the code, simplifying the verification phase;

– the verification itself that is performed on the handset and is driven by the annotations but whose correction does not depend on it: if the annotations are wrong, the verification will fail, it will never validate an ill-typed code.

For more details, the reader can look at the deliverables of the European Mobius project [MOB 05] and especially at the first results obtained for some simple static analysis for which the same principles have been applied [BES 06b, BAR 08].

10.8.2. *Existing viruses and malware*

Currently, there are few malicious mobile applications and very few users have been exposed to these applications. Nevertheless, some applications, including the CommWarrior virus, have caused serious damage to a small but detectable number of users. The risk is significantly higher for mobile phones with an open operating system which is more difficult to secure. To summarize in terms of risk management, the potential impact is severe up to very serious, while the probability is somewhere between unlikely and possible. On closed handsets, which still represent about 40% of the world fleet, the likelihood of a successful attack is extremely low and therefore the risk is almost zero. However, the techno-economic analysis of any new feature offered on environments for executing downloaded applications must take into account the risks it creates and balance the benefit of providing this feature with the additional costs of the measures to reduce the risks.

10.9. Bibliography

[AND 72] ANDERSON J., Computer security technology planning study, Report, US Air Force Electronic Systems Division, October 1972.

[APP 02] APPEL A. W., "Deobfuscation is in NP", available on Andrew Appel website, 2002.

[BAR 08] BARTHE G., CRÉGUT P., GRÉGOIRE B., JENSEN T. and PICHARDIE D., "The MOBIUS Proof Carrying Code infrastructure", *Proc. of the 6th International Symposium on Formal Methods for Components and Objects (FMCO'07)*, vol. 5382 of *Lecture Notes in Computer Science*, Springer-Verlag, p. 1–24, 2008.

[BEL 03] BELSON K., "Beware the worm in your handset", *New York Times*, November 2003.

[BES 06a] BESSON F., DUFAY G. and JENSEN T., "A formal model of access control for mobile interactive devices", *ESORICS*, vol. 4189 of *Lecture Notes in Computer Science*, Springer-Verlag, p. 110–126, 2006.

[BES 06b] BESSON F., JENSEN T. and PICHARDIE D., "Proof-carrying code from certified abstract interpretation and fixpoint compression", *Theoretical Computer Science*, vol. 364, no. 3, p. 273–291, 2006.

[BID 03] BIDDLE P., ENGLAND P., PEINADO M. and WILLMAN B., "The darknet and the future of content distribution", *Proceedings of 2nd ACM Workshop on DRM*, vol. 2696 of *Lecture Notes in Computer Sciences*, p. 155–176, 2003.

[BOY 77] BOYER R. and MOORE J., "A fast string searching algorithm", *Communications of the ACM*, vol. 20, no. 10, p. 762–772, 1977.

[BUR 03] BURDY L., CHEON Y., COK D., ERNST M., KINIRY J., LEAVENS G.T., LEINO K.R.M. and POLL E., "An overview of JML tools and applications", in ARTS T. and FOKKINK W. (Eds.), *Eighth International Workshop on Formal Methods for Industrial Critical Systems (FMICS '03)*, vol. 80 of *Electronic Notes in Theoretical Computer Science*, Elsevier, p. 73–89, 2003, preprint University of Nijmegen (TR NIII-R0309).

[CAC 05] CACHERA D., JENSEN T., PICHARDIE D. and SCHNEIDER G., "Certified memory usage analysis", in FITZGERALD J., HAYES I. and TARLECKI A. (Eds.), *FM 2005*, no. 3582LNCS, Springer-Verlag, p. 91–106, 2005.

[CHR 03] CHRISTENSEN A.S., MØLLER A. and SCHWARTZBACH M.I., "Precise analysis of string expressions", *Proc. 10th International Static Analysis Symposium, SAS '03*, vol. 2694 of *LNCS*, Springer-Verlag, p. 1–18, June 2003, available from http://www.brics.dk/JSA/.

[CLA 03] CLAESSENS J., PRENEEL B. and VANDEWALLE J., "(How) can mobile agents do secure electronic transactions on untrusted hosts? A survey of the security issues and the current solutions", *ACM Trans. on Internet Technology (TOIT)*, vol. 3, no. 1, p. 28–48, ACM Press, 2003.

[CON 04] CONSORTIUM S., GDP/STIP 2.2 Specification for Java, Report, Global Platform, 2004.

[COR 00] CORBETT J.C., DWYER M.B., HATCLIFF J., LAUBACH S., PASAREANU C.S., ROBBY and ZHENG H., "Bandera: extracting finite-state models from Java source code", *ICSE '00: Proceedings of the 22nd International Conference on Software Engineering*, New York, NY, ACM Press, p. 439–448, 2000.

[COR 01] CORMEN T., LEISERSON C., RIVEST R. and STEIN C., *Introduction to Algoritms*, MIT Press, 2001.

[COU 77] COUSOT P. and COUSOT R., "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints", *Conference Record of the Fourth Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, Los Angeles, California, ACM Press, New York, NY, p. 238–252, 1977.

[CRÉ 05] CRÉGUT P. and ALVARADO C., "Improving the security of downloadable Java applications with static analysis", *BYTECODE 05*, vol. 141-1 of *Electronic Notes in Theoretical Computer Science*, p. 129–144, 2005.

[CRÉ 07] CRÉGUT P., "Extracting control from data: user interfaces of MIDP applications", in BARTHE G. and FOURNET C. (Eds.), *TGC 2007*, vol. 4912 of *Lecture Notes in Computer Sciences*, Springer-Verlag, p. 41–56, 2007.

[DOD 85] DOD, Department of Defense Trusted Computer System Evaluation Criteria, Standard no. 5200.28-STD, American Department of Defense, December 1985, known as the "Orange Book".

[ECM 99] ECMA, ECMAScript Language Specification, Standard no. 262, ECMA, 1999, known as the ISO/IEC 16262 norm.

[ECM 06] ECMA, Common Language Infrastructure (CLI) 4th edition, Standard no. 335, ECMA, 2006.

[END 03] ENDORF C., SCHULTZ G. and MELLANDER J., *Intrusion Detection and Prevention*, McGraw-Hill Professional, 2003.

[EUR 99] EUROPÉENNE U., "Directive 1999/93/CE du Parlement européen et du Conseil, du 13 décembre 1999, sur un cadre communautaire pour les signatures électroniques", *Journal Officiel L13 du 19.1.2000*, p. 12–20, 1999.

[EUR 00] EUROPÉENNE U., "Directive 2000/31/CE du Parlement européen et du Conseil du 8 juin 2000 relative à certains aspects juridiques des services de la société de l'information, et notamment du commerce électronique, dans le marché intérieur", *journal officiel L 178 du 17.7.2000*, p. 1–16, July 2000.

[F-S 05] F-SECURE, "Virus Description Database", http://www.f-secure.com, 2005.

[GOV 03] GOVINDAVAJHALA S. and APPEL A.W., "Using memory errors to attack a virtual machine", *IEEE Symposium on Security and Privacy*, Oakland, May 2003.

[GOW 04] GOWDIAK A., "Java 2 micro edition (J2ME) security vulnerabilities", *Hack In The Box Conference*, Kuala Lumpur, Malaysia, 2004.

[GRO 97] GROVE D., DEFOUW G., DEAN J. and CHAMBERS C., "Call graph construction in object-oriented languages", *ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages and Applications*, p. 108–124, 1997.

[GWA 05] Mobile Application Security, Background. Deliverables. Recommendations, Executive summary, mas gen doc 002, GSM Assocation, February 2005.

[JAN 08] JANSEN W. and SCARFONE K., Guidelines on Cell Phone and PDA Security (Draft), Special publication 800-124, NIST, 2008, Draft.

[JSR 00a] JSR 30 EXPERT GROUP, Connected Limited Device Configuration (CLDC), Version 1.0, Java specification request, Java Community Process, 2000.

[JSR 00b] JSR 37 EXPERT GROUP, Mobile Information Device Profile (MIDP), Version 1.0, Java specification request, Java Community Process, 2000.

[JSR 01] JSR 36 EXPERT GROUP, Connected Device Configuration 1.0, Java specification request, Java Community Process, March 2001.

[JSR 02a] JSR 118 EXPERT GROUP, Mobile Information Device Profile (MIDP), Version 2.0, Java specification request, Java Community Process, November 2002.

[JSR 02b] JSR 218 EXPERT GROUP, Connected Limited Device Configuration (CLDC), Version 1.1, Java specification request, Java Community Process, 2002.

[JSR 05] JSR 218 EXPERT GROUP, Connected Device Configuration 1.1, Java specification request, Java Community Process, August 2005.

[KIL 73] KILDALL G.A., "A unified approach to global program optimization", *POPL '73: Proceedings of the 1st Annual ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, New York, NY, ACM Press, p. 194–206, 1973.

[L'E 01] DE L'EUROPE C., Traité sur la cybercriminalité, Traité STCE no. 185, Conseil de l'Europe, Budapest, 2001.

[LEA 99] LEAVENS G.T., BAKER A.L. and RUBY C., "JML: A notation for detailed design", in KILOV H., RUMPE B. and SIMMONDS I. (Eds.), *Behavioral Specifications of Businesses and Systems*, p. 175–188, Kluwer Academic Publishers, Boston, 1999.

[LEA 04] LEAVENS G.T., POLL E., CLIFTON C., CHEON Y., RUBY C., COK D. and KINIRY J., JML Reference Manual, July 2004, spécification en corus d'évolution.

[LIN 06] LINK R., Server-Based Virus-Protection on Unix/Linux, Diploma Thesis, University of Furtwangen, 2006, http://www.openantivirus.org/diploma-thesis.pdf.

[LIV 05] LIVSHITS V.B. and LAM M.S., "Finding security vulnerabilities in Java applications with static analysis", *SSYM'05: Proceedings of the 14th Conference on USENIX Security Symposium*, Berkeley, CA, USA, USENIX Association, p. 18–18, 2005.

[M-C 03] M-COMM WORKING GROUP, Mobile Commerce (M-COMM); Mobile Signature; Security Framework, Report no. 1O2 206, ETSI, 2003.

[MAT 03] MATTISON R., *The Telco Revenue Assurance Handbook*, Lulu Press, 2003.

[MIL 05] MILANOVA A., ROUNTEV A. and RYDER B.G., "Parameterized Object Sensitivity for Points-to Analysis for Java", *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 14, no. 1, p. 1–41, January 2005.

[MOB 05] "MOBIUS IP/FET IST project", 2005, http://mobius.inria.fr.

[MUT 00] MUTTIK I., "Stripping down an AV engine", *Virus Bulletin Conference*, September 2000.

[MYE 99] MYERS A.C., Mostly-static decentralized information flow control, PhD Thesis, Massachusetts Institute of Technology, January 1999, Technical Report MIT/LCS/TR-783.

[NAC 96] NACHENBERRG C., "Understanding and managing polymorphic viruses", *The Symantec Enterprise Papers*, vol. 30, 1996.

[NAC 02] NACHENBERG C., "Behavioral Blocking: The Next Step in Anti-Virus Protection", March 2002.

[NIB 79] NIBALDI G.H., Proposed Technical Evaluation Criteria for Trusted Computer Systems, Report no. M79-225, AD-A108-832, MITRE Corp., Bedford, Mass., October 1979.

[NIE 99] NIELSON F., NIELSON H.R. and HANKIN C.L., *Principles of Program Analysis*, Springer, 1999.

[OMA 05] OMA, "OMA SyncML Common Specification v1.2", 2005.

[OMA 06a] OMA, "OMA Device Management Security", 2006, OMA-TS-DM-Security-v1-2-20060602-c.

[OMA 06b] OMA, "OMA Device Management v1.2", 2006.

[OMA 06c] OMA, "OMA Firmware Update Management Object Architecture", 2006, OMA-AD-FUMO-V1-0-20060625-C.

[OMT 08] OMTP, "Security Threats on Embedded Consumer Devices", 2008.

[ROS 03] ROSE E., "Lightweight bytecode verification", *Journal of Automated Reasoning*, vol. 31, no. 3-4, p. 303–334, 2003.

[SAB 03] SABELFELD A. and MYERS A., "Language-based information-flow security", *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 1, p. 5–19, January 2003.

[SUN 97] SUN MICROSYSTEMS, The Java Card 2.0 Language Subset and Virtual Machine Specification, Report, SUN Microsystems, 1997.

[SUN 00] SUNDARESAN V., HENDREN L., RAZAFIMAHEFA C., VALLÉE-RAI R., LAM P., GAGNON E. and GODIN C., "Practical virtual method call resolution for Java", *ACM SIGPLAN Notices*, vol. 35, no. 10, p. 264–280, 2000.

[SZO 05] SZOR P., *Virus Research and Defense*, Symantec Press, 2005.

[TAR 55] TARSKI A., "A lattice theoretical fixpoint theorem and its applications", *Pacific Journal of Mathematics*, vol. 5, p. 285–309, 1955.

[UNI 04] UNIFIED TESTING INITIATIVE, Java Verified Program Guide, Report, Sun Microsystems, Motorola, Nokia, Siemens, Sony Ericsson, February 2004, Version 1.0.

[UNI 05] UNIFIED TESTING INITIATIVE, Unified Testing Criteria for Java Technology-based Applications for Mobile Devices, Version 2.0, Report, Sun Microsystems *et al.*, May 2005, http://javaverified.com.

[VIS 03] VISSER W., HAVELUND K., BRAT G., PARK S. and LERDA F., "Model Checking Programs", *Automated Software Engg.*, vol. 10, no. 2, p. 203–232, Kluwer Academic Publishers, 2003.

[WIK 06] WIKIPEDIA, http://fr.wikipedia.org, 2006.

[XU 00] XU Z., MILLER B.P. and REPS T., "Safety checking of machine code", *PLDI '00: Proceedings of the ACM SIGPLAN 2000 Conference on Programming Language Design and Implementation*, New York, NY, ACM Press, p. 70–82, 2000.

[ZDZ 05] ZDZIARSKI J., *Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification*, No Starch Press, 2005.

[ZWI 00] ZWICKY E., COOPER S. and B.C., *Building Internet Firewalls*, O'Reilly, Sebastopol, USA, 2000.

PART 3

Emerging Technologies

This page intentionally left blank

Chapter 11

Security in Next Generation Mobile Networks

11.1. Introduction

The concept of next generation mobile networks appeared with the interconnection of telecommunication networks based on heterogenous telecommunication technologies and with specific value-added services proposed by different providers. Before such interconnection and independently of the type of technology used, there was in practice approximately one type of network per service. For example, a cell phone connected to the Internet could only access the limited Internet services proposed by its provider. Instead, the objective would be to have a unique, possibly heterogenous, network for access to all telecommunication services. However, with heterogenous technologies and services, security concerns must be clearly addressed. Indeed, how can we guarantee data or network integrity or how can we control a correct billing for subscribed services when we have to deal with multiple intermediaries?

We already discussed the convergence of networks in Chapter 9. In this chapter, we would like to go a step further and discuss the convergence of services notably multimedia. Previously, an operator used to offer access to a service located on a different platform or even at a different operator. In the new vision, the operator owns or is the multimedia service and access should be totally transparent to the subscriber irrespective of its location or the telecommunication technology used. It therefore becomes conceivable to envision the appearance of multimedia services on mobile telephones or to foresee the transformation of mobile operators into

Chapter written by Jérôme HÄRRI and Christian BONNET.

multimedia or television service providers challenging the market controlled by public or private television operators.

The *SIP (Session Initiation Protocol)* has been created with this objective in mind. It not only makes it possible to establish multimedia sessions on the Internet but may also be used by any network connected to the Internet or having access to it. It works similarly to SS7 with respect to call establishments and is actually intended to replace it in the near future. The most prominent SIP application is *Voice-over-IP (VoIP)*. Unfortunately, SIP is not capable of managing user or network mobility by itself. The community therefore proposed an extension called *IMS (IP Multimedia Subsystem)* which significantly improves access control and subscriber management. IMS not only administers a controlled access for subscribers to networks, but also enables the interconnection of heterogenous networks. The objective of IMS is first to guarantee transparent access for subscribers to services and second to facilitate the establishment of new services proposed by different operators irrespective of to the telecommunication technologies employed or the exact location of the subscribers.

VoIP is probably the first popular service to benefit from the interconnection of telecommunication operators and the Internet. With VoIP, it is indeed now possible to communicate with a party at a low cost irrespective of the party locations and possibly even totally free if both parties use a computer. With VoIP we actually assisted the return of traditional telecommunication services offered through the Internet, a novel concept when considering that the Internet had been initially created as a service offered by traditional telecommunication networks. Moreover, VoIP had a catalytic effect on emerging competitive operators on which we do not have much visibility or control or, more precisely, whose security provisions cannot excess those offered on the Internet. By shunting the local exchange carrier (LEC), the subscriber delegates the securitization of its calls and data to virtual telecommunication operators for which it may be difficult to obtain confidentially credentials. A subscriber therefore accesses a service that could be controlled neither by the operator nor by the network, therefore illustrating issues relating to privacy, security and access control.

A recent evolution of mobile networks appeared with multimedia applications and generated the new vision for mobile telecommunication networks illustrated in Figure 11.1. The core network is the Internet nebulae on which various heterogenous networks (landline, mobile or corporate) get connected. Even though the landline network appears disconnected to the Internet nebulae on the diagram, a tight and mutually beneficial collaboration actually exists between landline and Internet operators, where the former may lease Internet lines for their services and in other locations or on other occasions lease their landlines to Internet operators for their own applications.

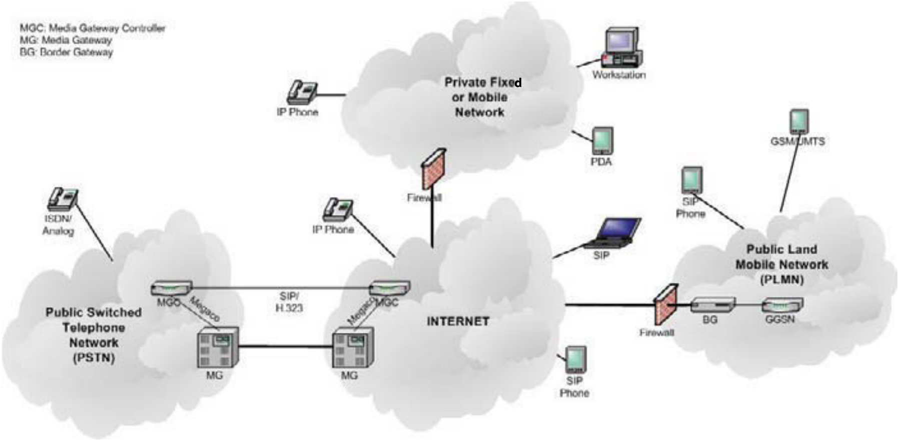


Figure 11.1. *Interconnection of telecommunication networks*

With the interconnection of multiple networks and virtual operators, it is now important to understand that the security of mobile telecommunication networks is no longer specific to the sole cellular operators but must be envisioned as end-to-end at the application layer and in collaboration with all other telecommunication network actors being irrespectively fixed, mobile or Internet. The subscriber's mobility or roaming in such telecommunication nebulae even provides further justifications for such an end-to-end security collaboration. As an example, let us consider the case of the security between an Internet service provider (ISP) and a corporate network. If security is provided end-to-end, then the Internet access guarantees access to the corporate network; if not, then a first authentication for Internet access is required, followed by a second VPN authentication to access the corporate network. If the user's mobility factor is further considered, the access control and security management then become very difficult if not considered end-to-end.

In this chapter, we will describe the various security mechanisms employed by different signaling or transmission protocols found in the so-called next generation mobile networks. Our objective is also to emphasize various security flaws and their probable malicious uses. With the strengthening of security provisions, we will also mention confidentiality and lawful interception provisions that have been required by law to all next generation mobile network actors.

11.2. The SIP

The SIP is an application-layer session initiation protocol standardized by the IETF. It is in charge of authenticating and locating the various actors of a SIP session. The SIP being independent of the type of data traffic, any type of communication protocol may be used. However, the Real-time Transfer Protocol (RTP) is the most widely used in practice for audio and video sessions. The SIP is also the open standard used by VoIP.

11.2.1. *SIP generalities*

SIP is a text protocol and shares similar response codes with HTTP. However, SIP differs from HTTP as a SIP agent is at the same time a client and a server. Figure 11.2 depicts SIP functionality. In general, SIP is composed of the following elements:

– *User Agent (UA)*: we may find it in all SIP phones or any other SIP-based applications. A communication between two SIP agents is established based on a *URI (Uniform Resource Identifier)* that is similar to an e-mail address.

– *Registar*: as we obviously need to know the IP address of the target SIP UA to establish a communication, the Registar is in charge of registering and maintaining this IP address into a database that will then link it with the target URI.

– *Proxy*: a SIP proxy has a middleman role between two SIP UAs in order to obtain their respective IP addresses. The SIP proxy retrieves the destination IP address from the database and then contacts the destination SIP UA. Data traffic never travels through a SIP Proxy but is directly exchanged between two SIP UAs.

– *Redirect Server*: a SIP redirect server receives requests from a SIP UA and is in charge of returning a redirection response indicating where the request should be retrieved.

– *Session Border Controller (SBC)*: this is a SIP-ready intelligent firewall. When a SIP UA initiates a SIP session, two connections are built, one for signaling and one for data transmission. Although this process does not pose any problem when both SIP UAs are located within the same subnetwork, firewalls or NAT separating different networks may not be aware of the relationship between these two connections. They could therefore reject traffic from a subscriber in its subnetwork even if signaling successfully established that connection. NATs further generate address translation problems between multiple temporary addresses established by ISPs and their visibility on the Internet. In order to correct these issues, it has therefore been proposed to create a Session Border Controller (SBC) acting as an application-layer gateway and guaranteeing a correct address translation and assisting network administrators in managing the flow of sessions passing through their subnetworks.

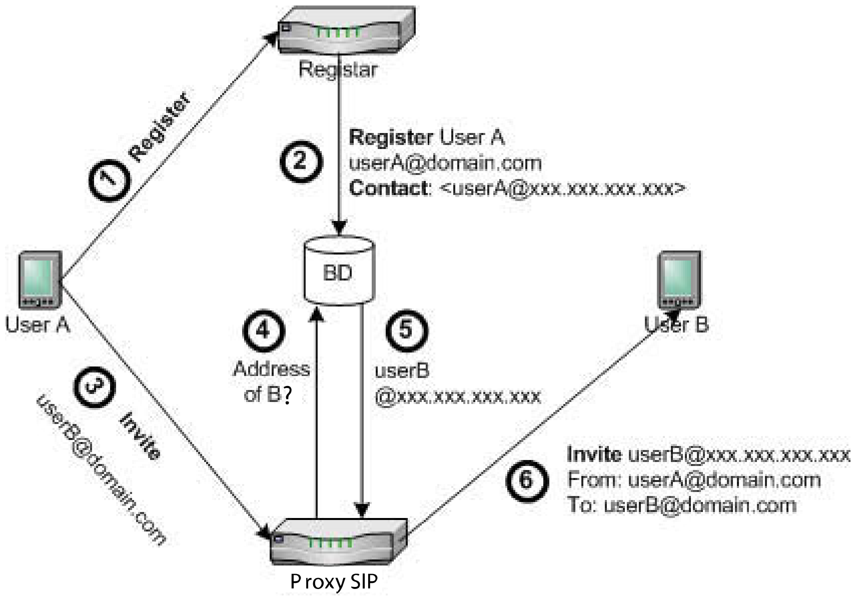


Figure 11.2. *SIP functionality*

11.2.2. *SIP security flaws*

Like SS7, SIP has not been conceived with default security mechanisms and like any textual protocol, it is very sensitive to attacks. We now provide some examples of typical attacks on SIP-based applications. For more details, we refer the interested reader to the IETF standard [SIP 02] or [SSC 03]:

– *Registration hijacking*: a Registar evaluates the identity of a SIP UA based on the message header. The “FROM” field of the SIP header may yet be arbitrarily tampered with and opens the door to malicious (de-)registrations. By impersonating a SIP UA, a malicious user may request to replace URI contact addresses with its own contact information on the database. This demonstrates the requirement for authentication provisions between SIP UAs and SIP proxies.

– *Impersonating a proxy*: a SIP UA contacts a SIP proxy in order to correctly route its requests. The proxy may be impersonated by a malicious user and then perturb or even reroute requests to third parties. The mobility factor in a SIP network further exacerbates such a security flaw. In order to combat possible security breaches, a mutual authentication process must therefore be established.

– *Tearing down sessions*: by passively listening to SIP call parameters and then by inserting a SIP control message “BYE”, a malicious user may abruptly close a SIP session. By further inserting a SIP “RE-INVITE” message, it may then redirect a call to an arbitrary third party. In order to combat this kind of attack, SIP connection parameters must be hidden and the ID of the SIP UA must be authenticated.

– *Integrity*: it is unfortunately possible to arbitrarily tamper with the content of SIP messages with malicious data. A SIP proxy, even fully authenticated, should never have access to the content of a SIP message, especially during key agreement transactions.

– *Denial-of-Service (DoS)*: denial-of-service is an attack vector that aims at making a network element unreachable or unavailable. SIP proxies also have to be integrated into the Internet in order to be able to intercept legitimate requests from SIP UAs located around the world, SIP networks are therefore very vulnerable to a range of various DoS attacks. It should be noted that if SIP proxies are compromised or unavailable, the whole SIP network becomes non-operational considering that SIP UAs are not able to recognize each other and cannot access SIP databases. One vector to combat this security flaw is by controlling registration attempts.

11.2.3. *Making SIP secure*

One of the vulnerabilities SIP must quickly fight is the integrity of its signaling messages. One solution called *Secure SIP* relies on encrypted links based on the *Transport Layer Security (TLS)* protocol. Initially used to secure HTTP sessions, TLS may be reconfigured to secure SIP sessions against eavesdropping or tampering.

The authentication based on the HTTP Digest (MD5) algorithm makes it possible to authenticate the identity of a SIP UA or proxy. This protocol is based on the combination of challenge and credentials. In order to limit the transmission of confidential information between SIP Registars and SIP UAs, certificate exchanges are performed through TLS tunnels, therefore combating *Replay* attacks by credential interceptions.

SIP proxies authenticate to local SIP UAs or to other SIP proxies using TLS certificates provided by trusted Certificate Authorities (CA) and then delegate the established trust to intermediate SIP proxies in order to establish communication. For example, a SIP proxy cannot know how a SIP UA has been identified by another SIP proxy located in a different domain, but it trusts it as a TLS tunnel has been established between them.

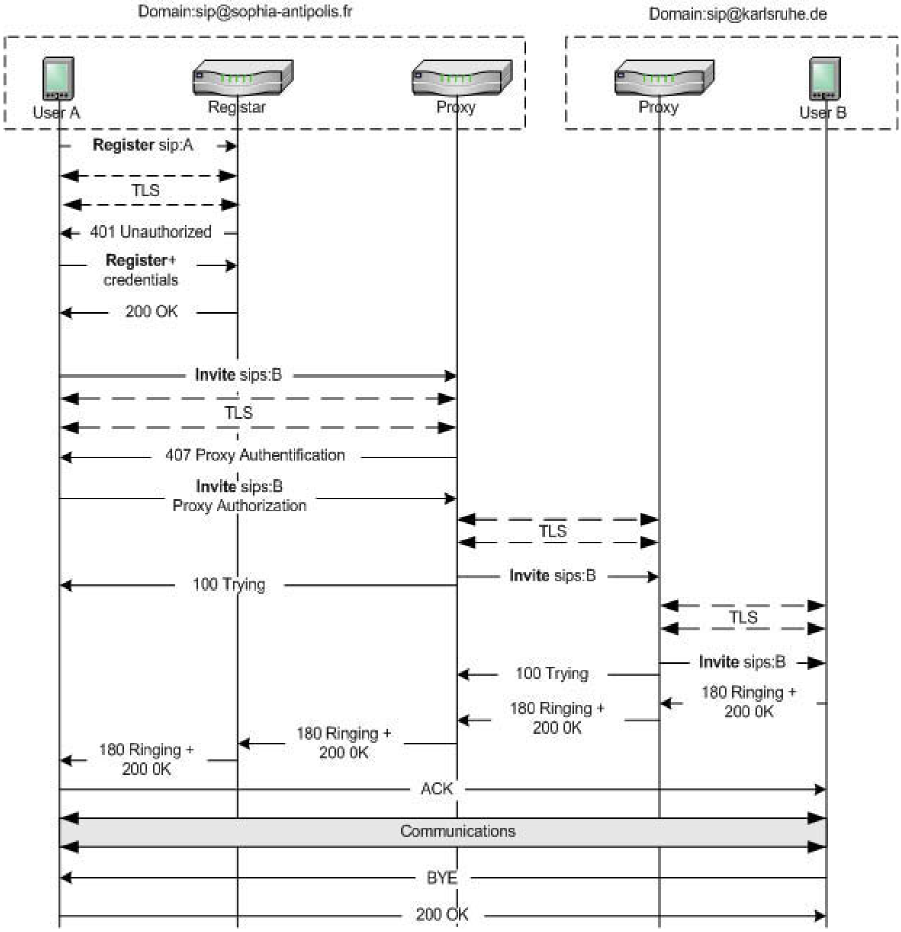


Figure 11.3. *Secured SIP communication establishment*

Another trust criterion guaranteed by SIP comes from the use of secured SIP addresses called *SIPS*. Like HTTPs, every time a SIP UA attempts to contact another SIP UA using a SIPS address, a segment-by-segment TLS connection is established between the SIP UA and the SIP proxy located in the destination domain (see Figure 11.3). However, security associations between this SIP proxy and the destination SIP UA only depend on the security provisions from the destination domain.

Using IPsec AH or ESP to guarantee end-to-end integrity or confidentiality of SIP messages is unfortunately not possible as SIP proxies require reading and even writing into SIP headers in order to correctly direct messages. It is however possible to use IPsec AH or ESP on a segment-by-segment basis. The major advantage of using IPsec instead of TLS is that it supports TCP and UDP transmissions. Finally, another possible solution is to use S/MIME. Indeed, the SIP being able to transport MIME messages, S/MIME allows SIP UA to protect the content of SIP messages without altering the headers. Using S/MIME to secure end-to-end communications using SIP tunnels is also possible. In order to avoid eliminating legitimate packets whose headers would have been legitimately tampered with, the SIP RFC suggests a set of rules that would make it possible to differentiate legitimate from malicious messages.

Figure 11.3 illustrates a message exchange example of the registration and then the establishment of a communication between two SIP UAs that would also include the security mechanisms previously described.

Last but not least, SIP securitization issues are also related to the visibility of users. Indeed, in order to receive SIP calls, a user must render two public IP addresses visible to the Internet, one for signaling and one for data transfer. This is similar to putting your postal address on the Internet and leaving your door open and unattended. Such a situation being obviously a major concern to ISPs, they accordingly created session controllers located outside their firewalls and that basically act as post boxes for SIP sessions.

11.3. VoIP

VoIP is a new technology that made it possible to federate the data and voice communication worlds. Before VoIP, the only solution to transmit voice communication was to establish a circuit between the caller and the callee, which had the advantage of guaranteeing very good communication quality unfortunately at an equivalently high price. With Internet communications, it became absurd to be able to transfer millions of data bits around the world at a very competitive price but still pay a high toll just to be able to talk. VoIP therefore equilibrated the equation by transmitting real-time voice mostly through the Internet at that time but, thanks to the ITU Next Generation Networks (NGNs), soon also through any packet-switched communication network.

VoIP contains a signaling layer and a media transport layer. The signaling protocol, principally the H.323 used by operators although SIP has recently showed an increasing popularity, handles subscriber localization, communication setups and tear-downs. The media transport layer is principally the and is in charge of carrying media transmissions with real-time characteristics. IP eventually encapsulates the media packets and routes them through the network.

VoIP has been designed with a full interoperability in mind. If voice calls are established within a same packet-switched network (wireless or IP), then no further structure is required. However, if voice calls are transmitted from or to circuit-switched networks (PSTN or MPLN), then VoIP requires the following new elements:

– *Media gateway (MG)*: a media gateway interrupts a voice communication of a circuit-switched network, then samples and encodes the voice before eventually delivering it as voice packets to the IP network. The reverse operation is performed for a voice communication from an IP network.

– *Media Gateway Controller (MGC)*: also called "soft switch", this receives VoIP signaling information and assigns resources to MGs such as instructing them to send or receive voice packets.

– *Signaling Gateway (SG)*: this provides a transparent signaling interconnection between the SS7 network and the IP network. It is in charge of interrupting SS7 signaling if necessary or converting it to the IP format before directing it to the MGC. As such gateways are critical to VoIP networks, they are typically deployed in swarms.

– *IP-enabled Service Control Point (IP-SCP)*: this has a similar role as an ordinary SCP but is totally integrated into an IP network. It may also still be reached by SS7 networks.

These elements do not have a unique denomination as they are developed by different standardization bodies or research groups. In H.323 for example, a MGC is called a Gatekeeper (GK), while SGs and MGs are simply both called a Gateway.

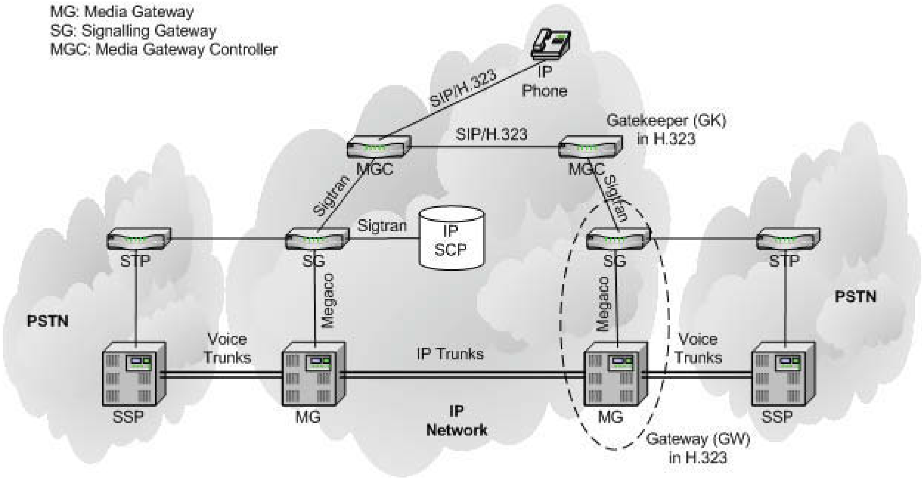


Figure 11.4. *VoIP SIGTRAN architecture*

Several VoIP standards exist at the ITU or the IETF but the latter have recently appeared to take the lead over ITU. We now provide a brief description of the VoIP protocol stack proposed by the IETF:

– *Stream Control Transport Protocol (SCTP)*: this is actually the SIGTRAN protocol in charge of transporting SS7 signaling between a SG and a MGC or between a SG and an IP-SCP.

– *Megaco (H.248)*: this represents a control protocol between a MGC and several MGs.

– *Session Initiation Protocol (SIP)*: this manages calls between MGCs or between MGCs and SIP phones.

–*RTP*: this is a protocol based on UDP and transports voice packets with real-time constraints.

Figure 11.4 illustrates a typical example of a VoIP architecture.

11.3.1. *VoIP security flaws*

The revolutionary aspect of VoIP is to be able to remove the complex proprietary structures of circuit-switched telecommunication operators. A VoIP may indeed avoid spending a fair amount of money in telecommunication switches and replace them with routers or soft switches at a more attractive price and accordingly transpose this financial saving on competitive prices for its services. The corollary is that the required architectures to build VoIP networks are basically “off the shelf”, bringing more virtual operators into the arena and accordingly increasing the chances of intrusion or impersonation of VoIP networks. VoIP must therefore protect its signaling and data networks, as various attack vectors, which we briefly describe below, should not be ignored:

– *Confidentiality*: signaling is as important as communication considering that a compromised signaling may let a malicious user obtain sensitive information about a legitimate subscriber. For example, a compromised SG could be the source of eavesdropping attempts on VoIP calls or of the logging of calls passed by a legitimate subscriber.

– *Eavesdropping*: a conversation itself is also put at risk if a MG is compromised, a malicious user being able to intercept and tamper with VoIP packets in order to eavesdrop on it.

– *Man-in-the-Middle*: VoIP conversations are also vulnerable when it comes to Man-in-the-Middle attacks which could typically allow a malicious user to intercept a call and tamper with its parameters. Such an attack is also considered critical as it leads to identity thefts or call redirections that are totally transparent to the legitimate subscriber or the VoIP network.

– *DoS*: unlike circuit-switched networks, there is basically no guarantee on the available resources provided by VoIP networks. They may therefore be easy targets for DoS attacks rendering critical network elements totally inoperable and significantly reducing the Quality of Service (QoS) provided to the subscribers.

– *Non-repudiation*: once a destination accepts a call, it is important to have mechanisms guaranteeing that this destination cannot later deny having accepted it.

– *VoIP servers and terminals*: being computers in the first place, these are very vulnerable to attacks. It is indeed not trivial to compromise an analog telephone but the software contained in VoIP phones can be easily be tampered with.

11.3.2. *Making VoIP secure*

Various solutions have been proposed to secure VoIP networks. The first approach is to secure its signaling layer. The signaling between MG and SG being based on IP, the SIGTRAN [SIG 99] protocol suites have been proposed. SIP is still required for signaling between MGCs. Some VoIP operators also use proprietary solutions instead of the SIP or SIGTRAN. In order to secure their messages, the SIP and SIGTRAN are both based on IPsec and TLS. For a comprehensible introduction to SIGTRAN, see [DAR 06].

It is then also important to secure the conversations themselves. Several encryption ciphers may be envisioned such as IPsec or Secured-RTP which all offer a sufficient privacy level at a reduced conversation quality that is totally acceptable to subscribers and operators. Within Secured-RTP, packet authentication is performed by MiKEY.

However, IPsec becomes problematic when IP traffic is transmitted through NATs, as illustrated in [IPV 06]. This issue is actually not problematic when only one VoIP subscriber is located behind a NAT, as it typically swaps a private IP address with a public IP address. However, this becomes critical when several IPsec sources are behind this NAT and communicate with a unique server beyond it. The IP address translation indeed becomes problematic and generates an asymmetric address translation to a single target address and accordingly redirects all VoIP traffic to a single VoIP subscriber. IPsec may therefore not be used when multiple VoIP subscribers in a same subnetwork communicate with the same server beyond a NAT.

It is finally necessary to also protect the VoIP network elements. Similar protection mechanisms to IP networks are used, such as port or router access control with the addition of redundancy to project key elements.

For more information relating to VoIP security, an exhaustive list of VoIP security flaws and the respective recommendations to correct them, see [DOS 06].

11.4. IP Multimedia Subsystem (IMS)

The IP Multimedia Subsystem (IMS) is a new 3.5G to 4G standard and constitutes a further evolution compared to the SIP as it provides an intermediate layer in core networks to move from a classical call mode (circuit) to a session mode. IMS is based in part on SIP signaling but enhances it with its ability to open several sessions while on call. IMS may be considered as an intelligent SIP as it is able to open multimedia sessions and also to add intelligent routing rules in order to manage multimedia sessions considering new parameters such as localization, and the availability or the type of terminal. Initially created for cellular networks, IMS has been extended to wireless and landline networks in collaboration with TISPAN. The IMS architecture therefore symbolizes the convergence between the worlds of mobile and fixed networks and the Internet.

11.4.1. *IMS architecture*

IMS includes a set of functions that are not specifically distributed per node. Different functions may exist on the same system or the same function may be distributed in different systems. We give below a summary of the various IMS entities:

– *Home Subscriber Server (HSS)*: this is a major database informing IMS core elements about call or session parameters. It has a similar role as a HLR/AuC in GSM networks.

– *Media Resource Function (MRF)*: this hosts multimedia resources in the subscriber's home network.

– *Application Server Function (AS)*: this hosts and executes telecommunication services such as MMS, SMS or Lawful Interception (LI).

– *Serving Call Session Control Function (S-CSCF)*: this is a central node of an IMS network and is located at the bottleneck of all signaling messages. It is actually a SIP server that is also in charge of controlling IMS sessions. It is always located in a subscriber's home network. The S-CSCF uses the DIAMETER protocol in order to securely contact the HSS to obtain information about a subscriber.

– *Interrogating Call Session Control Function (I-CSCF)*: this is a SIP proxy Server located at the edge of the IMS domain and acts as a gateway to a subscriber's home network. It also uses the DIAMETER protocol in order to question the HSS to obtain the location of a subscriber.

– *Proxy Call Sessions Control Function (P-CSCF)*: this is an IMS gateway and a SIP proxy server at the same time. It is in charge of authenticating a subscriber and initiating a transport mode IPsec ESP secured link with a subscriber. The P-CSCF accordingly protects information accessing an IMS network.

– *Breakout Gateway Control Function (BGCF)*: this is a SIP server that contains all routing functionalities for telecommunication networks. It is used when a subscriber calls a telephone number located in a circuit-switched network (PSTN or PLMN).

Figure 11.5 illustrates a typical architecture example between a home IMS and a visited IMS, making it possible to connect a UMTS subscriber to a PDA located on a WLAN.

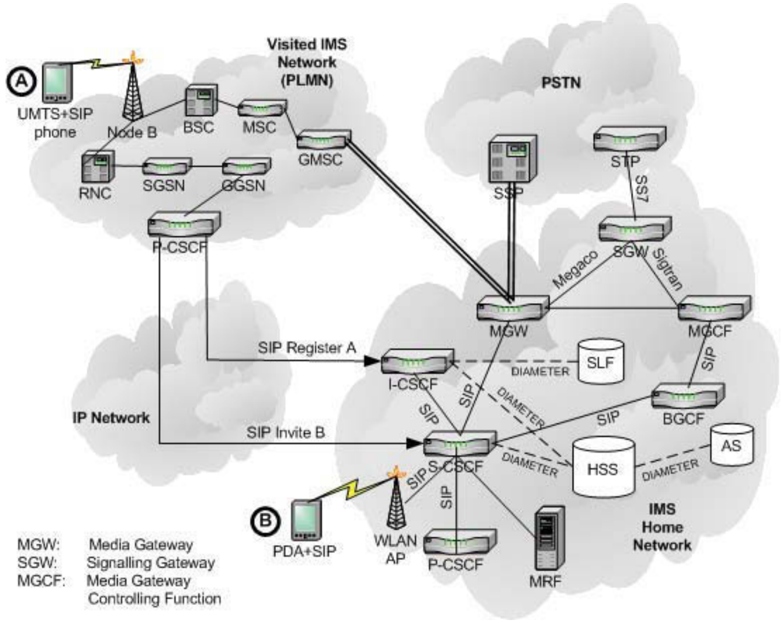


Figure 11.5. *IMS architecture*

11.4.2. *IMS security*

Unlike most of the solutions and protocols proposed in the past, IMS has been created with advanced up front security provisions. The *IMS AKA* is based on *secure SIP* and the *UMTS AKA*. The IMS security architecture can be divided into two categories: IMS core network security and visited IMS network security.

11.4.2.1. *IMS core network security*

The IMS core network security architecture is similar to the recommendations for the 3GPP for the UMTS. For example:

– *Confidentiality and integrity*: communications between the different entities forming the IMS core network are protected by the IPsec ESP in tunnel mode. Moreover, communications between the HSS and IMS entities are based on DIAMETER. As all IMS entities implement SIP, they natively include TLS. It is therefore also possible to guarantee confidentiality and integrity using TLS instead of or in conjunction with IPsec in the IMS core network. Internal communication of an IMS core network having a low probability of crossing a NAT, the IPsec/NAT issues illustrated in section 11.3 do not apply here.

11.4.2.2. *Visited IMS network security*

When a subscriber accesses an IMS network outside of his home network, it must connect to a P-CSCF. IMS provides the following security provisions to secure the link between a P-CSCF and a subscriber:

– *Authentication*: the authentication of a subscriber is handled by the S-CSCF. The authentication process is called IMS AKA and provides a mutual authentication between the subscriber and the home network. The 3GPP standard recommends using the UMTS AKA mechanism. The authentication vector is carried by the SIP and is obtained similarly to UMTS. Figure 11.8 depicts a typical authentication case.

– *Re-authentication*: even though subscribers are always fully authenticated after a successful network registration, the IMS may decide to initiate a new authentication process during a session if it has legitimate doubts about a subscriber. If so, the S-CSCF sends a re-authentication request.

– *Confidentiality*: subscriber confidentiality is guaranteed by using two different IMS-specific identifiers: the *IM Private Identity (IMPI)* that is safely held by the HSS and by the *IP Service Identity Module (ISIM)* in the subscriber's USIM card, and the *IM Public Identity (IMPU)* that is transmitted over the network. Confidentiality provisions for SIP signaling between a subscriber and a P-CSCF depend on the security policies in each visited network, but the 3GPP standard recommends rejecting subscribers that would refuse or would not have the capacity to encrypt communications. The two parties must negotiate the IPsec ESP cipher algorithm having the choice between the DES-EDE3-CBC and the AES-CBS and then determine the encryption key CK^{exp} from an expansion of the cipher key CK generated during the IMS AKA.

The 3GPP standard recommends the expansion rule illustrated in Figure 11.6.

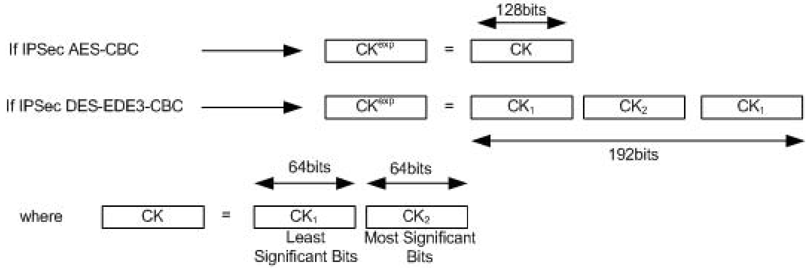


Figure 11.6. *Cipher Key (CK) expansion proposal*

– *Integrity*: in order to protect SIP signaling, integrity checks based on IPsec in transport mode are implemented between a subscriber and the P-CSCF. First, the two parties agree on the employed cipher between HMAC-MD5-96 and HMAC-SHA-1-96 and then they extract the integrity key IK^{exp} from an expansion of the IK generated during the IMS AKA.

The 3GPP standard recommends the expansion rule illustrated in Figure 11.7.

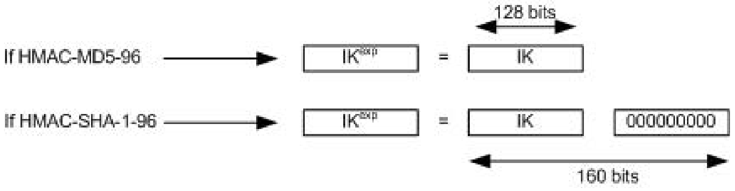


Figure 11.7. *IK expansion proposal*

Figure 11.8 depicts the two-step IMS registration mechanism: the first step is a challenge to the subscriber while the second step effectively registers it if the challenge has been correctly answered.

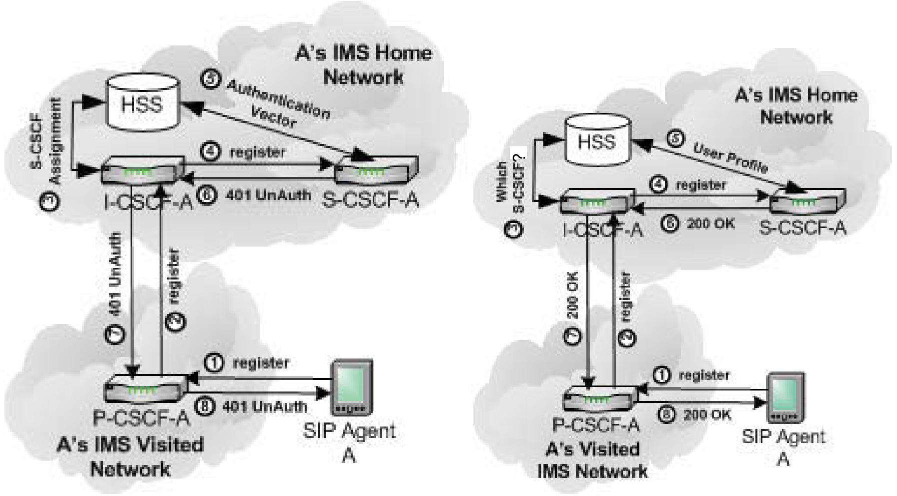


Figure 11.8. *Two-step IMS registration: IMS sends a challenge to A (left); IMS registers A (right)*

Figure 11.9 illustrates the IMS AKA mechanism in more detail. We assume that all communications in the subscriber's IMS home network up to the P-CSCF are secured by IPsec in tunnel mode and that the cipher key generation mechanism is the one recommended by the 3GPP standard for UMTS. The P-CSCF is the subscriber's access point to his home network and therefore has the double responsibility of authenticating him and also of creating security associations with him. Like the UMTS AKA, secret keys are never transmitted on the link between the P-CSCF and a subscriber.

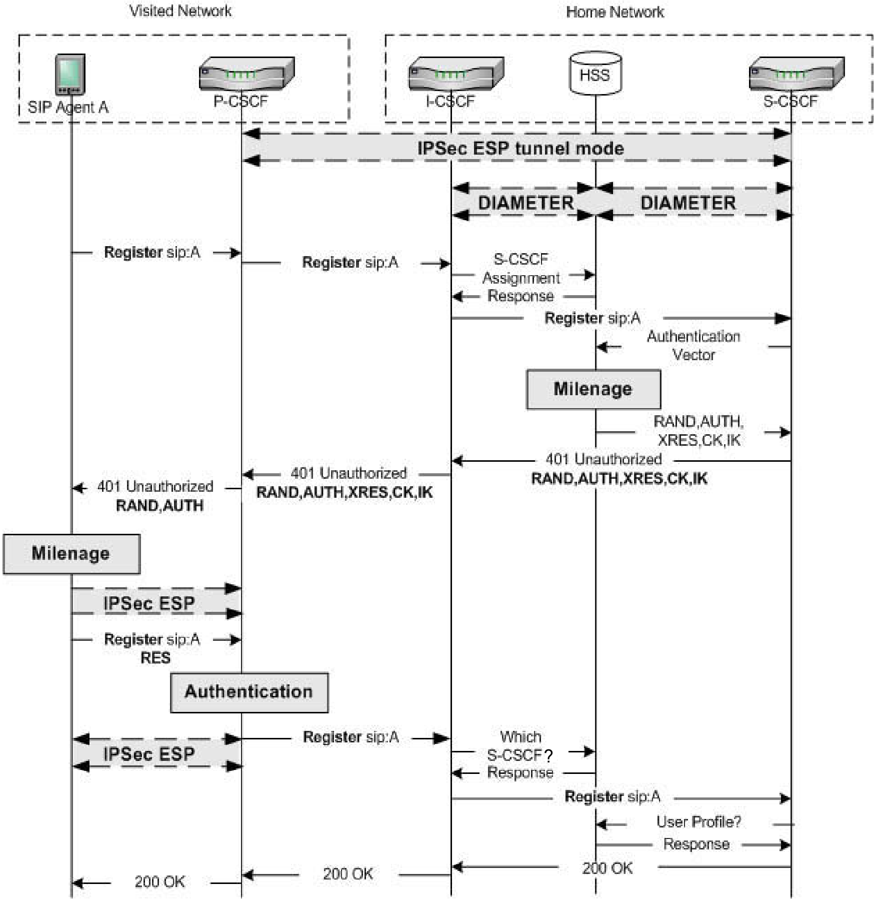


Figure 11.9. *IMS AKA*

Finally, Figure 11.10 depicts the negotiation for cipher algorithms and security associations between a P-CSCF and a subscriber. The two entities exchange a list of supported algorithms in order to agree on a common value. If they cannot, the 3GPP recommends rejecting the access.

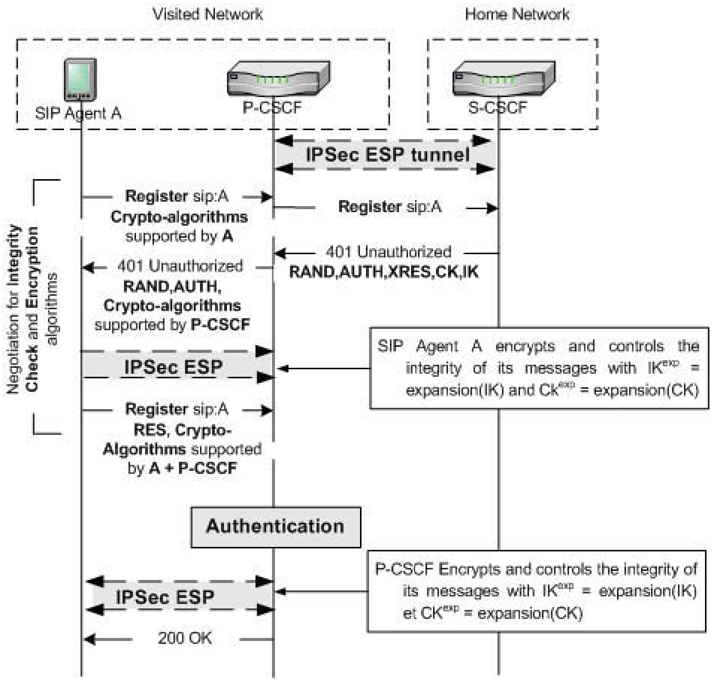

Visited Network
Home Network
SIP Agent A
P-CSCF
S-CSCF
IPSec ESP tunnel
Negotiation for Integrity Check and Encryption algorithms
Register sip:A
Crypto-algorithms supported by A
Register sip:A
401 Unauthorized
RAND,AUTH,XRES,CK,IK
401 Unauthorized
RAND,AUTH, Crypto-algorithms supported by P-CSCF
SIP Agent A encrypts and controls the integrity of its messages with IK^exp = expansion(IK) and CK^exp = expansion(CK)
IPSec ESP
Register sip:A
RES, Crypto-Algorithms supported by A + P-CSCF
Authentication
P-CSCF Encrypts and controls the integrity of its messages with IK^exp = expansion(IK) et CK^exp = expansion(CK)
IPSec ESP
200 OK

Figure 11.10. *IMS supported cipher algorithms negotiation*

11.4.3. *IMS security flaws*

Despite the care taken by the 3GPP community to provide sufficient security provisions to IMS, some security flaws may still be exploited and are listed in the IMS specification [IMS 08]:

– *Early IMS*: a simplified early version of IMS called *Early IMS* also exists and does not include some security provisions such as IPsec or the joint use of USIM and ISIM. It had initially been proposed in order to ease the deployment of IMS either when equipment was too expensive or the mechanisms were too complex. This IMS version may not be considered as secured. In the latest IMS standard, *Early IMS* is no longer tolerated.

– *Unauthenticated re-registration*: by initiating an unauthenticated re-registration, a malicious user masquerading as a legitimate subscriber can actually generate DoS attacks by answering with a false RES to the legitimate IMPU-based challenge sent by the P-CSCF. The IMS would accordingly close the session attached to the legitimate IMPU.

– *P-CSCF bypass*: once the user has successfully authenticated to a P-CSCF, it can maliciously try to send SIP messages directly to a S-CSCF and accordingly bypass integrity checks provided by IPsec ESP at the P-CSCF. The following security breaches could be generated:

- the P-CSCF is not able to generate charging information;

- the malicious user accessing a S-CSCF could masquerade as other legitimate users and send SIP "INVITE" or "BYE" messages and potentially stealthily eavesdrop on or tear down their sessions;

- a malicious user may act as a legitimate P-CSCF.

In order to combat these problems, subscribers must never be able to contact a S-CSCF directly. SIP source address spoofing should also be avoided.

11.5. 4G security

The so-called fourth generation networks, also called next generation networks (NGN) by the ITU, are currently under development. They promise an unprecedented maximum wireless throughput of 100 Mb/s. While 3G networks witnessed the appearance of heterogenous networks with the transparent interconnection of IP, PSTN and PLMN, 4G networks will make provisions for a full heterogenity in the radio sub-system and the total superposition and cooperation of various radio technologies. For example, it is envisioned that a WLAN and a cellular network will be transparently connected without any communication or QoS interruption.

At the network subsystem, the ETSI with TISPAN and the 3GPP with IMS work together to define a network subsystem including the IMS as a core component which would be in charge of guaranteeing a total cooperation between the various fixed and mobile networks (PSTN, PLMN, WLAN) in an all IPv6 secured environment. Such 4G subsystems will be transparent to the subscriber, hiding their technical complexity and delegating to 4G User Agents (UAs) the choice and negotiation for the best communication technology to be used as a function of the requested multimedia application. A larger cooperation between the various communication actors is therefore expected in order to provide high quality services to users in a secured environment.

Figure 11.11 is a schematic representation of 4G networks composed of four key layers: *user*, *access*, *transport* and *service*, and a total transparency in the communication technologies and protocols used at each layer.

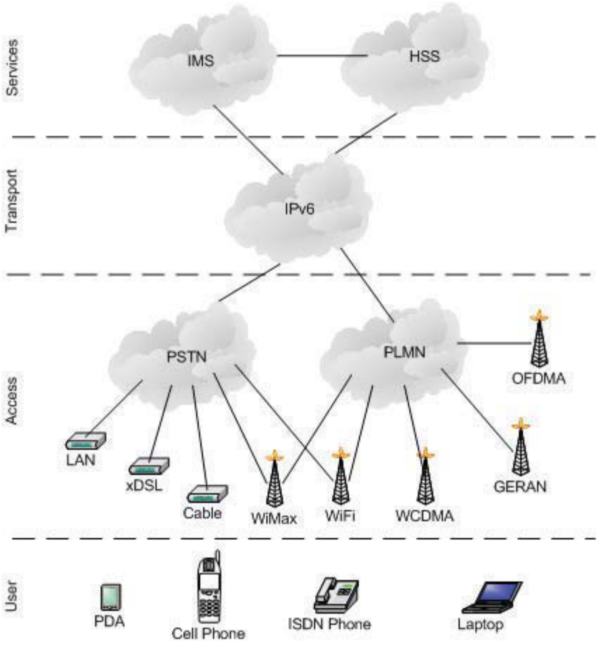


Figure 11.11. *Schematic vision of the architecture of 4G networks*

At this time, beside the security flaws already identified for IMS, UMTS or GSM/GPRS, 4G technology is not mature enough to clearly evaluate its security weaknesses. It is obvious that such typical heterogenity between real or virtual operators will contribute to probable security breaches.

The TISPAN working group, acknowledging that transmission through NAT was not initially considered by the 3GPP IMS standard, added an appendix in the new version of the IMS standard describing an UDP option for IPsec ESP in tunnel mode. This would indeed make it possible to use IPsec ESP through NAT, notably between a subscriber and a P-CSCF. Further provisions have also been added to include SIP digests, TLS and the co-existence of different authentication schemes in the latest IMS standard [IMS 08] to smoothen out the interconnection with TISPAN. For more details about TISPAN and its interconnection with IMS, see the ETSI TISPAN specification in [TIS 06] or [IMS 08].

11.6. Confidentiality

The interconnection of networks and their (lack of) cooperation make confidentiality management a very complex task. National and international agreements have been signed in order to guarantee the confidentiality of users and data. It is however important to keep in mind that with the current objective of a transparent network and service heterogenity, it might not be possible to guarantee such confidentially. National laws currently make network operators liable for any leak regarding personal information on their network.

For a long time, operators secured their network by obscurantism using weakly secured protocols. The relative protection of their networks came from a total network topology hiding and from only using proprietary equipments. With network interconnections, these measures are no longer conceivable. More robust security mechanisms have therefore been developed and made successful security breaches more a matter of cryptanalysis than a proven protocol flaw. The complexity of these encryption algorithms efficiently protected telecommunication operators. A remaining and recurring question therefore appeared: what about lawful interception?

Lawful interception (LI) is a mechanism used by all governmental security agencies in the world in order to legally obtain information on or even the communications of an individual or a group of subscribers. In the past, law enforcement agencies (LEAs) used identified security flaws in telecommunication networks to their profit. For example, they used cryptographic backdoors to decrypt communications or act as legitimate access points or even networks to eavesdrop on conversations or infiltrate networks.

With the current increase in network security, this kind of interception has become more limited. A response has obviously been proposed or more precisely imposed. It is now compulsory to be able to provide open access to LEA of any operator's network based on a simple legal order. In concrete terms, mechanisms must be established by each operator in order to identify and track individuals in a centralized way when legitimate court orders request it. This new approach of lawful interception significantly alters our vision of our private life. Indeed, we no longer trust complex security mechanisms to guarantee our confidentiality, but instead the justice and legal system of a sovereign state.

The counterpart of this new legal control form is the absolute requirement to include lawful interception mechanisms at the conception of a protocol. The Achilles' heel of this approach is the centralization of all interception capabilities that could create potential targets for malicious intrusion attempts. From an architectural point of view, a lawful interception is triggered by logging on a specific interface in the MSC for the UMTS and then intercepting any targeted signaling, voice or data. It is also possible to have a real-time report of a targeted subscriber's activity.

11.6.1. *Terminology*

Different types of lawful interceptions have been defined as a function of the interception's target:

– *Network interception:* access point-based interception irrespective of any targeted subscriber.

– *User interception:* user-based interception directed at a targeted subscriber ID. A subscriber may have several IDs within the same network.

– *Local interception:* interception limited to a specific part of a sub-network that can be network or user-oriented.

11.6.2. *Protection of interception mechanisms*

In practice, interception mechanisms provide a total interception capacity on information related to a user or a network element. It is therefore desirable to secure such an "operation center" for subscriber privacy. Several mechanisms have been proposed not only to guarantee that an LEA can access an operator's network but also to guarantee that an operator may authenticate the LEA that requested access to its network:

– *Flexible interception*: it is possible to limit the impact of lawful interceptions to a specific legal framework or simply to deactivate it.

– *Centralized administration:* only the *Administration Function (ADMF)* may have access to the interception interfaces in telecommunication networks.

– *Confidentiality, integrity and authentication*: communications between the different interception interfaces and the ADMF, and between the ADMF and LEA, are guaranteed at least by security algorithms such as VPN or CUG (Closed User Group).

Figure 11.12 illustrates the macroscopic functionality of lawful interception mechanisms. We refer readers interested in the lawful interception field to [INT 06] or [AQS 05].

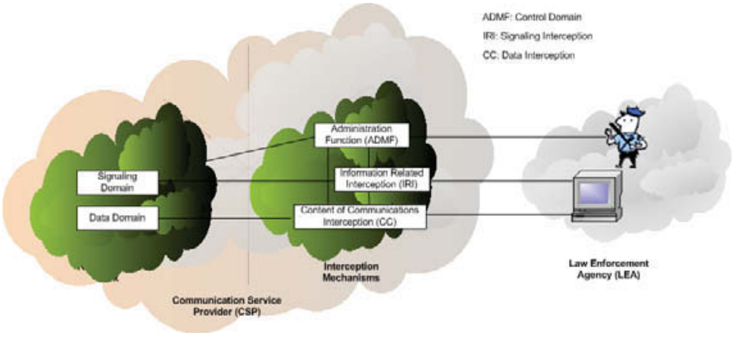

ADMF: Control Domain
IRI: Signaling Interception
CC: Data Interception
Administration Function (ADMF)
Signaling Domain
Information Related Interception (IRI)
Data Domain
Content of Communications Interception (CC)
Interception Mechanisms
Law Enforcement Agency (LEA)
Communication Service Provider (CSP)

Figure 11.12. *Lawful interception schema*

11.7. Conclusion

One feature of the telecommunication world is that it always witnessed an alternate succession of mobile and fixed telecommunication orientations. The first Intelligent Network (IN) was in its time the GSM, a mobile network. Understanding the potential of IN, PSTNs adapted it on their SS7 landline networks. Based on their commercial success, INs were later extended by PSTN operators before being adapted again to mobile networks with CAMEL for instance.

A similar analysis could be done with the recent evolution of the SIP towards IMS and its extensions that have created transparent infrastructures for mobile multimedia services. Facing such a strong potential and even before the large scale deployment of the IMS, PSTN operators started working on a new network architecture called NGN that would eventually also contain in part the IMS.

The convergence of telecommunication networks has been made even more difficult by the various operators and development consortia (ITU, IETF, ETSI, 3GPP). Although we have witnessed multiple mergers between mobile and fixed telecommunication operators over the last few years, it has been more about commercial merging than technical merging, each one of them being interoperable but not based on a common architecture.

Facing the complexity of such a total convergence conundrum, security concerns remain critical today. The IMS brought significant progress compared to the SIP or even SS7. Indeed, IMS not only guarantees an access control between different networks and subscribers, but also for multimedia services. Despite all efforts created on security mechanisms, it is unfortunately illusory to believe in a large scale totally secured public network. The "total security" dream is unfortunately not possible unless you physically unplug your network. It is more reasonable to hope for a system where the potential threats from identified security flaws depend on a tradeoff between the financial and technical capacities to actually employ them and the actual benefit obtained from successful breaches based on these threats.

As an example, let us consider the example of credit card-based financial transactions. It is sometimes astonishing to see the global concern behind the danger of transmitting one's credit card number to a service provider on the Internet, whereas it is seen as totally acceptable to give it to any service provider on the phone (hotel reservations for instance). The potential financial benefits are similar but the low technical expertise required to access transaction packets on IP networks compared to impersonating a proprietary PSTN or PLMN justified a complex encryption of all financial transactions on IP networks.

We may therefore observe that user susceptibility with respect to the security of their data is mostly based on a trust relationship between a user, a service provider and the network transmitting the requested service. By challenging such a trust relationship, the convergence of networks and services into heterogenous and transparent networks significantly eases the appearance of potential security breaches whose success will mostly depend on the popularity of the services offered by NGMNs.

11.8. Bibliography

[AQS 05] AQSACOM, "White Paper on Interception of IP Networks", http://www.aqsacomna.com/us/articles/LI3GWhitePaperv4%2E.pdf.

[DAR 06] Jim Darroch, "Introduction to Sigtran", *White Paper*, Artesyn Technologies, 2006.

[DOS 06] A. Doswald *et al.*, "Best Practices for VoIP-SIP Security", University of Geneva, www.td.unige.ch/pdf/BP_VoIP_Security.pdf.

[IMS 08] "3G Security: Access Security for IP based Services", 3GPP TS 33.203 version 8.4.0, 2008.

[INT 06] Newport Networks, "Lawful Intercept Overview", White Paper, http://www.newport-networks.com/cust-docs/87-Lawful-Intercept.pdf.

[IPV 06] Newport Networks, "IPSec in VoIP Networks", White Paper, http://www.newport-networks.com/cust-docs/91-IPSec-and-VoIP.pdf.

[NEW 06] Newport Networks, "SIP, Security and Session Controllers", White Paper, http://www.newport-networks.com/cust-docs/38-SIP-Security.pdf.

[SIG 99] IETF, "Framework Architecture for Signaling Transport", RFC 2719.

[SIP 02] IETF, "Session Initiation Protocol", RFC 3261.

[SSC 03] IETF, "Security Mechanism Agreement for the Session Initiation Protocol (SIP)", RFC 3329.

[TIS 08] "Protocols for Advanced Networking (TISPAN)", ETSI TS 182.006 v.2.0.4. 2008.

This page intentionally left blank

Chapter 12

Security of IP-Based Mobile Networks

12.1. Introduction

After the huge success of IP protocols in the interconnection of wired networks followed by a massive deployment of services to the non-mobile end user, today this protocol is expected to also offer different services to mobile users. Indeed, the development of wireless technologies and the evolution of smaller and smaller user terminals enabled the connection of mobile users through wireless networks. 3GPP2, one of the standardization bodies of mobile services, has after a delay expressed its interest in IP mobility. IP mobility helps mobile telecommunication technologies to deploy a simple protocol based on IP which is affordable to both the operator and the user. To achieve this objective, the standard mobile IP needs to provide, in addition to network connectivity of the mobile user, a guaranteed Quality of Service (QoS) and security. The standard IP mobility is defined both for version 4 of IP (MIPv4) and version 6 (MIPv6). In this chapter we will focus more on MIPv6 than MIPv4 as MIPv6 is a much improved version of the optimized MIPv4 from security and QoS point of view.

In this chapter, solutions for IP mobility are briefly described. We present security issues related to IP mobility. Indeed, the vulnerabilities associated with Mobile IP networks are identified and a detailed description of the mechanisms for securing data exchange with Mobile IPv6 is provided. More precisely, neighbor discovery, IP address auto-configuration and data protection via IP tunnel

Chapter written by Jean-Michel COMBES, Daniel MIGAULT, Julien BOURNELLE, Hakima CHAOUCHI and Maryline LAURENT-MAKNAVICIUS.

mechanisms are presented. Finally, security open issues are summarized at the end of this chapter. Other solutions to improve IP mobility support are presented, such as HIP (Host Identity Protocol) and NetLMM (Network Local Mobility Management).

12.2. Security issues related to mobility

The mobile IP communication model involves at least three communicating entities. As shown in Figure 12.1, a standard pattern of Mobile IP service (MIPv4 or MIPv6) is based on one or more mobility agents (HA: Home Agent, FA: Foreign Agent) in the network, a mobile node (MN) and a corresponding node (CN).

It should be noted that the FA entity exists only in version 4 of Mobile IP. Unlike the model for IP, IP mobility introduced intermediate entities between communicating nodes (MN, CN) to ensure the delivery of packets to the proper location of the mobile node. These intermediate entities are called mobility agents (HA, FA), whose features are detailed later. These entities are designed to redirect traffic between MN and CN, which introduces additional vulnerabilities compared to a conventional fixed IP (see section 12.2.1). It should be noted that some security issues of Mobile IP are related to transmission media used in the mobile access network: the wireless networks. Indeed, the link (Link 1 and Link 4) represented in Figure 12.1 is a wireless link and brings additional vulnerabilities to the Mobile IP communication model.

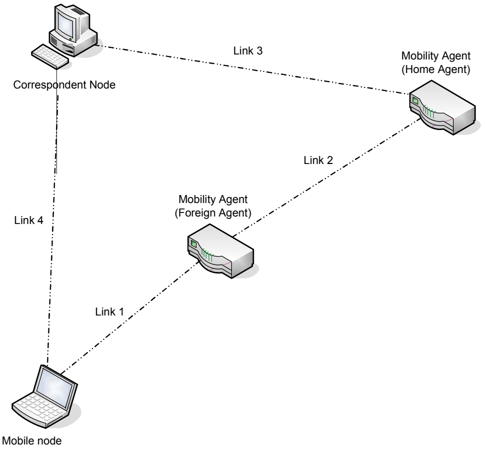


Figure 12.1. *Mobile IP basic communication model*

12.2.1. *Vulnerabilities of Mobile IP networks*

In this communication model, the MN is the most vulnerable entity because it visits different networks that have different levels of security. In addition, it also has access to wireless networks which are more vulnerable to eavesdropping than wired networks. We can classify security issues relating to MN, mobility agents and correspondent node as follows.

12.2.1.1. *Vulnerabilities related to wireless access*

The mobility of terminals is made possible thanks to wireless networks and the micro-electronics that can design powerful and small devices. Wireless networks are unfortunately more vulnerable to eavesdropping, replay attacks, man-in-the-middle attacks and IP spoofing, etc.

12.2.1.2. *Vulnerabilities related to the mobile node*

The fact that mobile node's packets can be delivered to any address of visited networks while it is moving is an open door due to possible diversion of traffic to non-legitimate nodes. This type of diversion may be caused by a malicious station that positions itself between the MN and one of the other entities in communicating with the MN (HA, FA, or CN).

The MN can also connect to false FAs who will then monitor its communications. For example, a false AR (Access Router) may behave as a fake FA operator and therefore attract victims on this false access network.

In other words, during the movement of the mobile node, it is important to secure the neighbor discovery phase, the IP address assignment and the auto-configuration process [RFC3756]. The authentication between all entities of the Mobile IP communication model is the first solution to these problems.

12.2.1.3. *Vulnerabilities related to mobility agents (FA, HA)*

The possibility of having a fake FA is another vulnerability of the Mobile IP communication model. A legitimate but compromised FA can also be used to spy on the content of MN-HA or MN-CN communications.

Another vulnerability is related to the HA. The fact that all traffic is redirected to the HA poses problems because, in addition to a bottleneck created at the HA, attacks carried out against the HA may cause a denial of service on the Mobile IP service.

12.2.1.4. *Vulnerabilities related to the corresponding node*

Finally, a CN could be the victim of false registration messages from the MN. In the case of Mobile IPv6, the mobile node registers with both the HA and CN. Another security issue relates to the case of a CN connected to a fake HA, in which case the fake HA may get all the traffic exchanged between MN and CN; this is a typical man-in-the-middle attack. It is then important to provide security between the CN and MN.

12.2.2. *Discovery mechanisms (network entities such as access routers)*

Securing Neighbor Discovery (ND)

The *Neighbor Discover Protocol* (NDP) is described in RFC 4861 [RFC4861]. Its goal is to enable IPv6 nodes to discover other nodes in the same subnetwork, and to enable autoconfiguration. The advantages, on a given link, for nodes to know each other is that each node has a cache of link layer address and can perform a link layer address resolution for a given IP address. One other possible advantage is to check whether or not one IP address is already used. This mechanism is also known as Duplicate Address Detection (DAD). Autoconfiguration is the ability of on IPv6 node to assign itself one IP address to one interface. It is a key concept in IPv6, since one IPv6 node should be able to regularly change its IP address. The NDP offers a number of facilities, leading to a number of security issues described in RFC 3756 [RFC3756].

SEcure Neighbor Discovery (SEND) [RFC3971] secures ND messages with different mechanisms based on the following options: *CGA*, *Signature*, *Timestamp* and *Nonce* options. The *CGA* option makes it possible to bind the IP address to a public key. The *Signature* option makes it possible to bind the NDP message to a public key. The public key is identified by its hash. The *Timestamp* option avoids the replay of the NDP message by sending the current time. The *Nonce* option enables matching between requests and answers.

SEND [RFC3971] secures autoconfiguration using certified routers. Nodes use routers for autoconfiguration and to route their packets. They must believe information from routers: the router is authorized to act as a router and to advertise routing prefixes. Authorization is provided by a trust anchor that can rely on a different deployment model, which can be centralized or decentralized. The path to the trust anchor must be specified by indicating the path of different trusted public keys. Both models rely on a Public Key Certificate (PKC). The path to the trusted router is done according to *Certification Path Solicitation* (CPS) and *Certification Path Advertisement* (CPA) messages.

Cryptographically Generated Addresses (CGAs) are described in RFC 3972 [RFC3972]. The idea is to generate 64 bits of the interface part of an IPv6 address that can be bound to the identifier, that is, its identifier, and network parameters like the subnet prefix. We also need the check to be computed very fast, when all parameters are provided, and we do not want any other parameters with the same result. The use of a hash function is recommended, as it is easy to perform and with almost no collision. The CGA is an IPv6 address where the 64 bits of its interface are the hash of the CGA parameters, where the 3 left most bits indicate the security level. The CGA parameters are the 16 byte *modifier*, the 8 byte *subnetwork* prefix, the 1 byte *collision* count, the *public key* of variable length and *extensions* of variable length. Due to hash function properties, it is believed that each time one of the parameters changes, the CGA is completely changed. The modifier can be any number and is used to enhance privacy, since different modifiers will result in totally different CGA addresses. As we can see, any node can create its own IPv6 address. Changing the modifier could help in impersonating a terminal and so one CGA could be generated with two different *keys* and *modifiers* on a given subnetwork. To avoid such an attack, the *modifier* is generated with special conditions that bind its value to the identifier. The level of security is indicated by the 3 bit *Sec* parameter. The modifier must be such that the 16**Sec* left most bits of the hash over CGA parameters give zero. This increases the complexity of finding different parameters that match the same CGA. Before assigning the CGA, the node must check the CGA is not already used by using the DAD mechanism. This address could for example be used as a non-CGA IPv6 address by another node. If a collision occurs, the collision field is incremented.

When a node wants to check if a CGA is valid, it first checks the value of the collision field in the CGA parameters and that the *subnetwork* is the same in the CGA parameters and in the CGA address. Then it checks that the hash of the CGA parameters is equal to the interface part of the IP address. Finally, it checks that the *modifier* fits the security requirements, that is, in the 16**Sec* left most bits of the hash give zero.

12.2.3. *Authenticity of the mobile location*

The basic security issue in Mobile IP is to prove that the registration messages updating the location of the mobile node are really sent from the mobile node itself and not from a fake node sketching a man-in-the-middle attack. The authenticity of such messages is important, otherwise it would be extremely easy to implement such an attack or hijacking attack where a machine intruder diverts mobile node traffic by sending a message to update the location instead of the mobile node itself. It is therefore very important to authenticate those entities involved in the operation of registration update of the mobile node. It is important to authenticate the mobile node and secure its messages to update its location. To prevent any fake FA and HA, the mobile node also needs to make sure that it communicates well with the real FA and HA. This requires very heavy security constraints because of security associations that will be mounted between these various entities (MN, HA, FA). These mechanisms are detailed below for Mobile IPv4 and IPv6.

12.2.4. *Data protection (IP tunnels)*

In addition to the problem of ensuring the authentication of the location update of the mobile node, it is also important to protect traffic exchanged between the MN and CN to guarantee non-disclosure of information to any spying node in the network. This is made possible using encrypted IP tunnels where the packets are encrypted then encapsulated in a new IP packet. In Mobile IP (MIPv4 or MIPv6), the IP tunnel or IP in IP enables the HA to make the mobility of the node transparent and ensures the delivery of the packets to the mobile node. The ends of the tunnel must trust each other and must process encapsulation (putting the protected IP packet in a new IP packet) and de-capsulation (retrieving the original protected IP packet from the received packet). To ensure the confidentiality of the traffic, the content of the tunnel is encrypted using mechanisms such as IPsec. In addition to encryption, the integrity and authenticity of data are also provided through the tunnel. For this, a negotiation of security associations between the ends of the tunnel is necessary before the start of transmission and the use of an IP VPN is recommended.

12.3. Mobility with MIPv6

MIPv6 is the main mobility protocol in IPv6. Many other MIPv6-based protocols have been specified to optimize the mobility or to obtain a dynamic deployment. This section is about all this family of protocols.

12.3.1. *IPv6 mobility mechanisms (MIPv6, HMIPv6, FMIPv6)*

IPv6 mobility relies on a family of protocols where each one answers specific needs (i.e. host mobility on a large scale or on a local scale, network mobility). The following sections describe the different standards of this family.

12.3.1.1. *Mobile IPv6*

The protocol description

The main protocol for IPv6 mobility is MIPv6. It is specified in the RFC 3775 [RFC3775].

A MIPv6 node, called an MN, owns two different IP addresses. The first is the *Home Address* (HoA): this is the main address linked to the *Home Network* (HN) and it may be registered in the DNS. The second is a temporary address called a *Care-of Address* (CoA), linked to any network, which is not the HN, that the MN is visiting; this type of network is called a *Foreign Network* (FN). An entity located in the HN, called a HA, is responsible for the MN mobility management.

When the MN is in its HN, it acts like a typical IPv6 node. Now, when the MN is in a FN, it obtains a CoA, linked to this network, and it informs its HA (of this CoA) by using a signaling message (*Binding Update* – BU). The HA confirms the reception of this by sending an acknowledgement (*Binding Acknowledgment* – BA).

When an IPv6 node, called a CN, wishes to communicate with the MN, IP packets sent by it to the MN and with the HoA as the IP address destination, are intercepted by the HA and are tunneled to the MN's CoA. In the same way, the MN encapsulates its IP packets in the same tunnel to the HA which forwards them to the CN. This process is called *reverse tunneling*.

An optimization exists, called *Route Optimization* (RO), which prevents all the traffic between the MN and the CN going through the HA. To do this, the MN sends a BU including its CoA to the CN. This last one confirms the reception of the message with a BA. Then, the CN sends its IP packets directly to the MN in including an IPv6 header extension, called "*Routing Header – type 2*", containing the MN's HoA. On the other side, the MN sends directly its IP packets in including an IPv6 header extension, called "*Home Address*", containing its HoA.

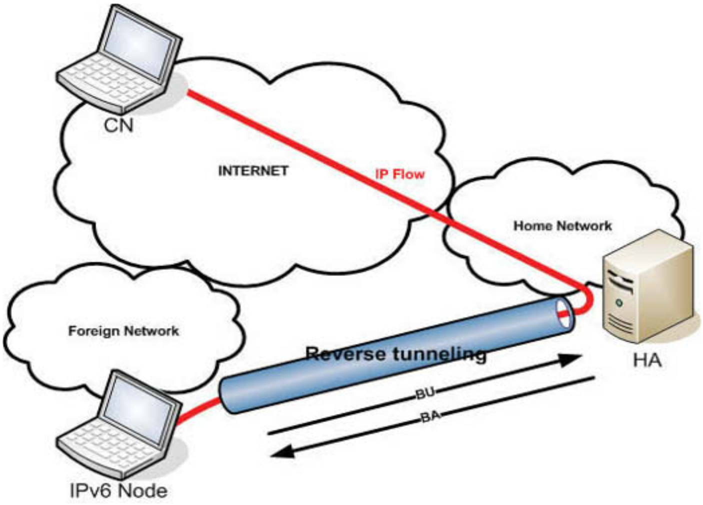


Figure 12.2. *MIPv6 architecture with reverse tunneling*

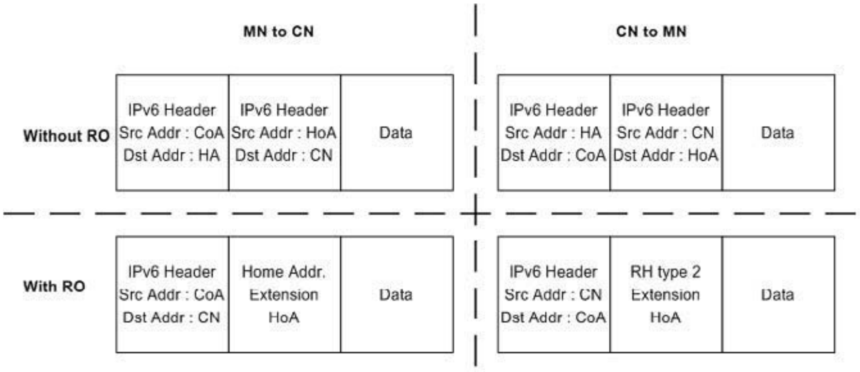


Figure 12.3. *IP packet format with RO and without it*

Security

MIPv6 signaling is critical for the mobility to work but can also be used to set up attacks if security mechanisms were not provided.

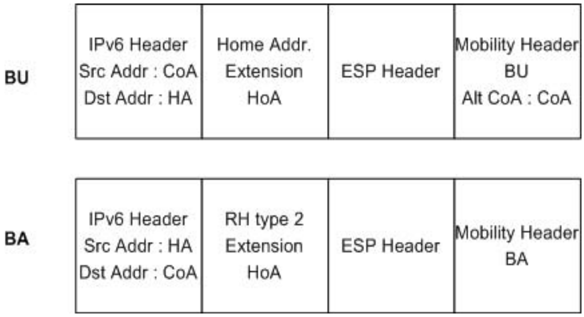


Figure 12.4. *BU/BA format*

Signaling security between the MN and the HA is specified in the RFC 3776 [RFC3776] and the RFC 4877 [RFC4877]. It is based on IPsec [RFC4301] to protect the BU and the BA between the MN and the HA. ESP [RFC4303] in transport mode is used to authenticate sent data: the IPsec *Security Association* (SA) guarantees that the sender is the real owner of the HoA and ESP ensures the integrity of the CoA which is located in the mobility option called the *Alternate Care-of-Address*. This prevents the BU's sender from impersonating another MN by using a fake HoA (that it does not own) and a node, located between the MN and the HA, to modify the CoA by intercepting the BU.

The standardized mechanism to secure RO signaling is complex because it is assumed that security cannot be based on an infrastructure. This mechanism, called *Return Routability* (RR), is specified in the RFC 3775 [RFC3775] and is used before the MN sends a BU to a CN. At first, the MN sends a message, called a *Home Test Init* (HoTI), in the *reverse tunneling* containing a cookie *(Home Init Cookie)*, its HoA and the CN's IP address. In parallel, it sends another message, called a *Care-of Test Init* (CoTI), directly to the CN which contains another cookie *(Care-of Init Cookie)*, its CoA and the CN's IP address. When the CN receives these two messages, it can generate a secret key, called a Kcn, and a nonce. Then it sends a message, called a *Home Test* (HoT), to the MN's HoA containing the *Home Init Cookie*, the index of a nonce and the *Home Keygen Token*. This last token is generated with Kcn, the generated nonce and the HoA. In parallel, the CN sends a message, called a *Care-of Test* (CoT), to the MN's CoA containing the *Care-of Init Cookie*, the index of a nonce and the *Care-of Keygen Token*. This last token is generated with Kcn, the generated nonce and the MN's CoA. When the MN receives these two messages, it can generate, from the *Home Keygen Token* and from the *Care-of Keygen Token*, a key, called Kbm, which will be used to secure information in the BU. Thanks to these four messages which are sent through two different paths, the RR mechanism provides a guarantee to the CN that the MN's location is really the CoA. Moreover, this mechanism provides proof to the CN that the MN is really the owner of the HoA, thanks to the HA vouching for the authenticity of the exchanged HoTI and HoT messages. Finally, it is necessary to protect these messages between the MN and the HA with IPsec.

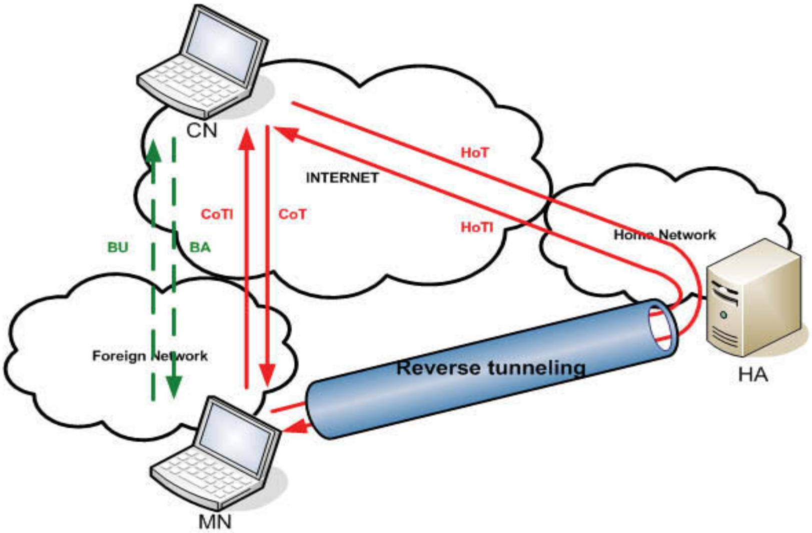


Figure 12.5. *Exchanges during the RR mechanism*

It is important to notice that a HA or a CN must drop any packet (except the RR mechanism messages in the case of a CN) coming directly from a MN when no BU was sent previously: this is to avoid “bombing” attacks as described in [DUP 07]. This sort of attack consists of sending BU messages containing fake information (the HoA or the CoA sent in the BU is in fact the victim’s IP address): the result is a DoS attack.

12.3.1.2. *HMIPv6*

The protocol description

The *Hierarchical MIPv6 Management* protocol (HMIPv6), which is based on MIPv6, was specified for a micro-mobility context: the MN only moves in the same FN (e.g. subnetworks of a company site). This protocol is specified in the RFC 5380 [RFC5380].

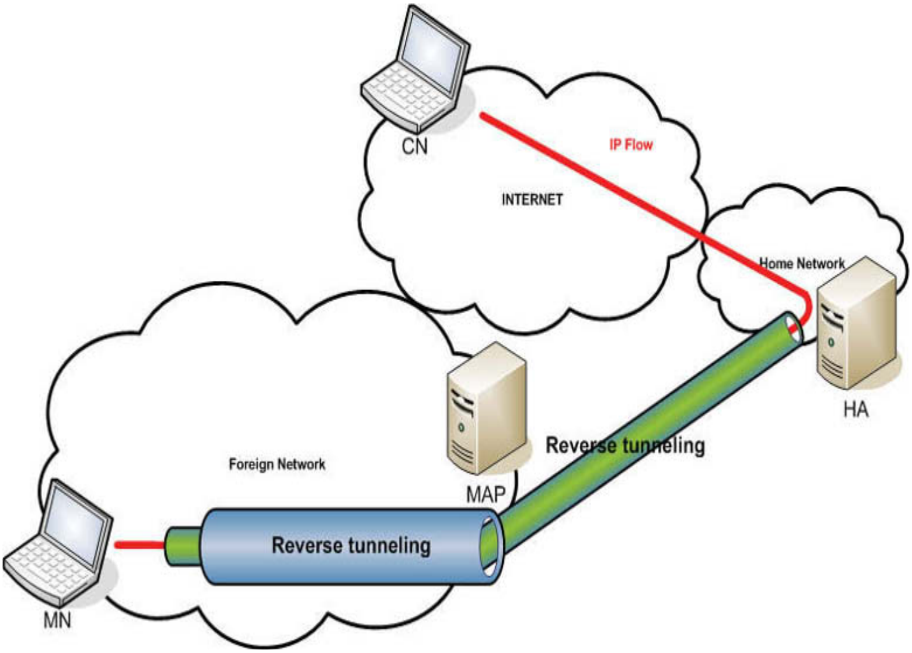


Figure 12.6. *HMIPv6 architecture*

In comparison to MIPv6, a HMIPv6 node, also called an MN, owns two types of CoA addresses: a *Regional Care-of-Address* (RCoA) and a *Local Care-of-Address* (LCoA). An entity, called *Mobility Anchor Point* (MAP), located in the FN, is in charge of managing the micro-mobility of the MN. Regarding MIPv6, the MAP may be considered as a local HA, the RCoA as a HoA and the LCoA as a CoA. So, when the MN arrives in the FN, it obtains a LCoA linked to the subnetwork where it is and a RCoA linked to the subnetwork where the MAP is located. Next, the MN sends a BU, containing the RCoA and the LCoA, to the MAP and sends a BU to its HA, containing its HoA and the RCoA. When the MN moves to another subnetwork into the same FN, it must only send a BU to the MAP. When a CN wishes to communicate with the MN, the IP packets sent by the CN, intercepted by the HA, are tunneled to the RCoA and finally, intercepted by the MAP and tunneled to the LCoA. The MN may perform RO with a CN by using either its RCoA or its LCoA instead of the CoA.

Security

HMIPv6 security is almost the same as in MIPv6.

The signaling between the MN and the MAP is similar to MIPv6: IPsec secures BU and BA. It is also assumed that there is a trust relationship between the MN and MAP as in MIPv6 between the MN and the HA.

For the signaling between the MN and the CN when RO is used, like MIPv6, the security is based on the RR mechanism. Now, here, the CoA must be replaced by either the RCoA or the LCoA.

12.3.1.3. *FMIPv6*

12.3.1.3.1. The protocol description

The protocol called *Fast Handovers for MIPv6* (FMIPv6) is a MIPv6 optimization allowing the MN to perform faster handovers between subnetworks. This protocol is specified in the RFC 5268 [RFC5268].

The goal is to allow the MN to communicate with the *Access Router* (AR): it will be able to inform them about its arrival, to quickly obtain a *new CoA* (NCoA) and also to redirect its traffic from the previous AR to its new AR. To do this, as described in the following figure, the MN asks its current AR (*Previous Access Router* – PAR) for information (e.g. access points, access routers, network prefixes, etc.) about the subnetworks around it thanks to the *Router Solicitation for Proxy Advertisement* (RtSolPr) message. The PAR provides it with the requested information thanks to the *Proxy Router Advertisement* (PrRtAdv) message. So, the MN informs the PAR it is going to leave this subnetwork thanks to the *Fast Binding Update* (FBU) message. Then, the PAR informs the New Access Router (NAR), selected by the MN, thanks to the *Handover Initiate* (HI) message. This acknowledges it with a message called *Handover Acknowledge* (HAck). Then, the PAR informs, using the *Fast Binding Acknowledgment* (FBack) message, the MN that it can do its handover to the new subnetwork and the NAR that the MN is going to arrive. Now, all the packets for the MN are intercepted by the PAR and tunneled to the NAR which stores them. Finally, when the MN arrives in the new subnetwork, it informs the NAR with the *Unsolicited Neighbor Advertisement* (UNA) message and gets back all the IP packets stored by the NAR. This is the "predictive mode".

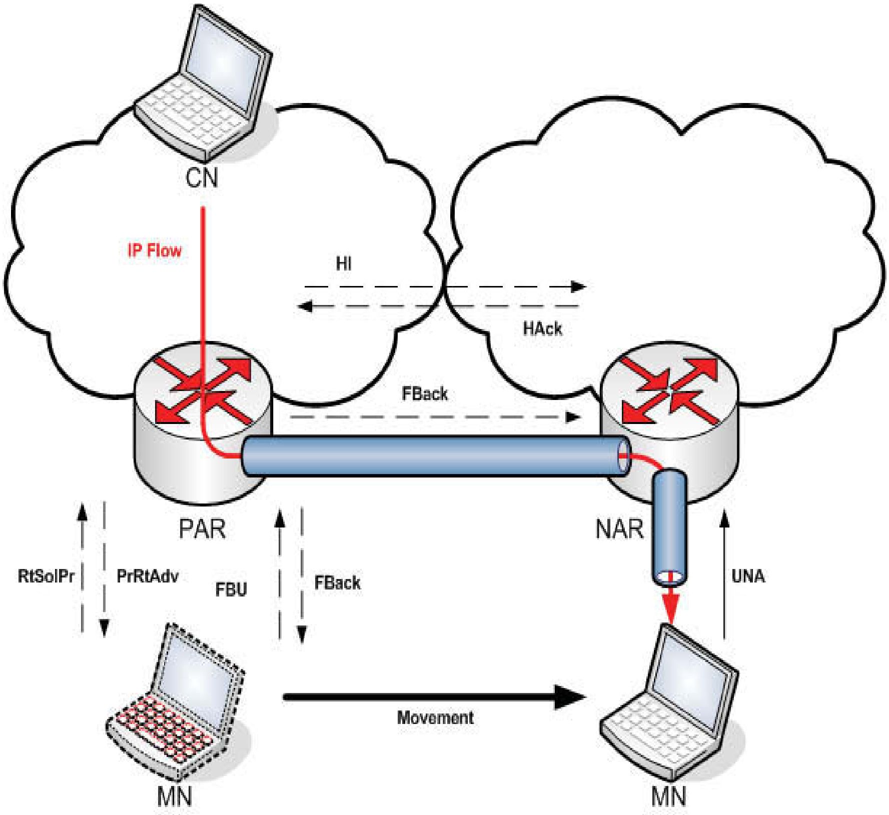


Figure 12.7. *FMIPv6 architecture and exchanges in "predictive mode"*

There is another mode named "reactive mode". In comparison with the previous mode, the MN has moved so quickly on the new subnetwork that it was not able to send the FBU in time. So, the MN sends a UNA to the NAR followed by a FBU to the PAR. Then, the PAR sends a HI to the NAR who replies with a Hack. Then, all IP packets that the PAR still receives for the MN are forwarded to the NAR.

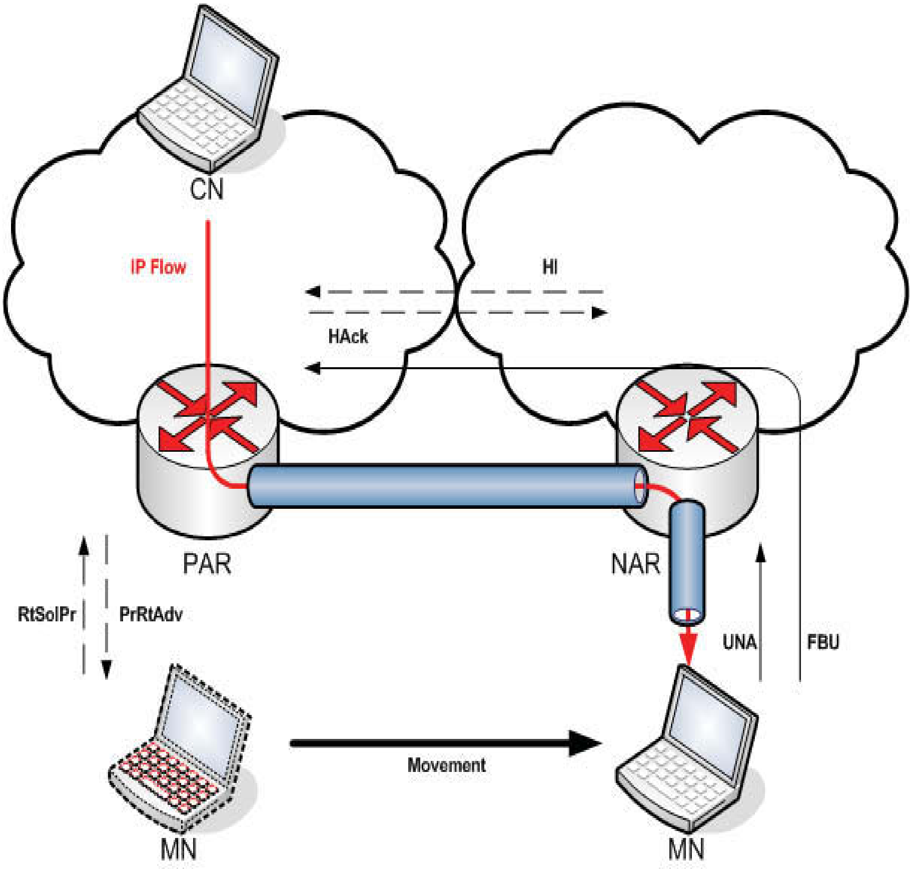


Figure 12.8. *FMIPv6 architecture and exchanges in "reactive mode"*

12.3.1.3.2. Security

Regarding the messages exchanged between the PAR and the NAR, it is strongly recommended to use IPsec, with IKEv2, to guarantee the integrity of the data.

As the RtSolPr and PrRtAdv messages are NDP extensions, these messages should be secured by using SEND. This last mechanism, SEND, should also be used to secure the FBU message as will be explained.

The security of the FBU message relies on a pre-shared key, exactly like the Kbm key specified for the RO in MIPv6: the CN here is the PAR. The solution standardized to set up this key is specified in the RFC 5269 [RFC5269] and is

SEND-based. The MN generates a public/private key pair, which will only be used to encrypt the Kbm key provided by the AR. This is obtained using a RtSolPr/PrRtAdv messages exchange, secured with SEND.

12.3.2. *Mobile IPv6 bootstrapping*

Since the Mobile IPv6 signaling is secured, it is necessary to have a technical solution in order to get a fast deployment in a large scale and sure from a security point of view. This procedure is known as the *Mobile IPv6 bootstrapping* or initialization mechanism.

12.3.2.1. *Problem*

The IETF has defined in the RFC 4640 [RFC4640] the necessary parameters in order to launch the Mobile IPv6 service as well as the different entities involved in the Mobile IPv6 bootstrapping process.

A Mobile IPv6 node needs the following information: a HoA, a Home Agent Address (@HA) and IPsec security associations with the HA.

The different entities involved in the bootstrapping process are the following:

– Access Service Provider (ASP): this is the network operator providing IP connectivity to a node;

– Access Service Authorizer (ASA): this is the network operator which authenticates the node and which authorizes the ASP to furnish IP connectivity to the node;

– Mobility Service Provider (MSP): this is the network operator which delivers the Mobile IPv6 service;

– Mobility Service Authorizer (MSA): this is the network operator which authenticates the node and which authorizes the MSP to offer the Mobile IPv6 service.

Figure 12.9 describes the location of the different entities in an Internet architecture.

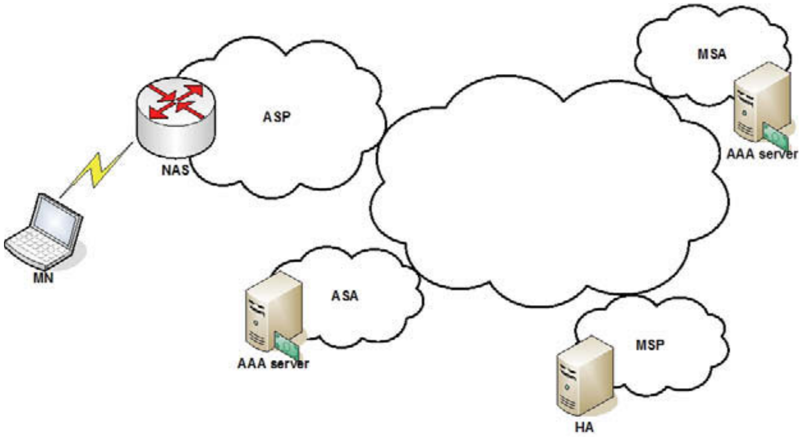


Figure 12.9. *Mobile IPv6 bootstrapping architecture*

Two scenarios are possible: either the MSA and ASA are not in the same administrator domain (*Split scenario*), or they are in the same administrator domain (*Integrated Scenario*). These two cases are studied below.

12.3.2.2. *Split scenario*

For this scenario, the mechanism is described in the RFC 5026 [RFC5026]. It is divided into the following steps: discovery of the HA IP address by the MN, establishment of IPsec security associations with this HA, HoA assignment and finally authentication and authorization of the MN by the MSA.

First of all, this mechanism assumes that the MN knows the MSP's domain name (e.g. by pre-configuration) and that it has access to the DNS service. Indeed, in order to obtain the IP address of an HA belonging to its MSP, the MN contacts the DNS server of the MSP's domain. The MN has two options: either it knows the HA's name in the DNS (i.e. its *Fully Qualified Domain Name – FQDN*) and it directly obtains the IP address thanks to a usual DNS request, or it does not know the HA's FQDN and it will use a DNS request of type service (i.e. MIPv6 service) based on the RR SRV which is described in the RFC 2782 [RFC2782].

In order to prevent the MN from receiving erroneous information from corrupted entities, the DNSSEC standard [RFC4033] may be used to secure this information.

Since the MN knows the HA's address, it uses IKEv2 [RFC4306] to establish IPsec SAs and to obtain its HoA. Indeed, the HA can assign a HoA to the MN. For this, the MN includes the INTERNAL_IP6_ADDRESS attribute in the

Configuration Payload during the IKE_AUTH exchange. The HA sets a HoA in the CFG_REPLY payload in the message sent to the MN. The document [RFC4877] gives details on this exchange. In this scheme, the MN cannot have a CGA [RFC3972] or *Privacy* [RFC4941] HoA. This problem can be solved if it is the MN which proposes a HoA to the HA. For this, the MN sends its HoA proposal to the HA via the INTERNAL_IP6_ADDRESS attribute in the CFG_REQUEST payload. If the HoA is valid, the HA confirms it in the CFG_REPLY with an INTERNAL_IP6_ADDRESS attribute containing the same HoA. If not, it sets a valid HoA that the MN will have to use.

It is worth noticing that in some deployment scenarios, the HA may not be able to authenticate and authorize the Mobile IPv6 service. This is the case when the MSP is not the MSA. The HA must contact the MSA either by using a Public Key Infrastructure (PKI) or by using an Authentication Authorization Accounting (AAA) infrastructure. The IETF is currently working on solutions to enable the MSP to communicate with the MSA.

Finally, the HoA may be registered in the DNS system in order to allow any correspondent to discover the MN's HoA. It is the MSP or the MSA which performs this operation because it is easier to set up security between a DNS server and one entity (HA in the case of the MSP or AAA server in the case of MSA) than between a DNS server and many IP nodes (MNs). Moreover, it allows the DNS server to be sure that the MN's name is tied to the right HoA, avoiding some DoS problems.

Figures 12.10 and 12.11 illustrate Mobile IPv6 bootstrapping in the split scenario.

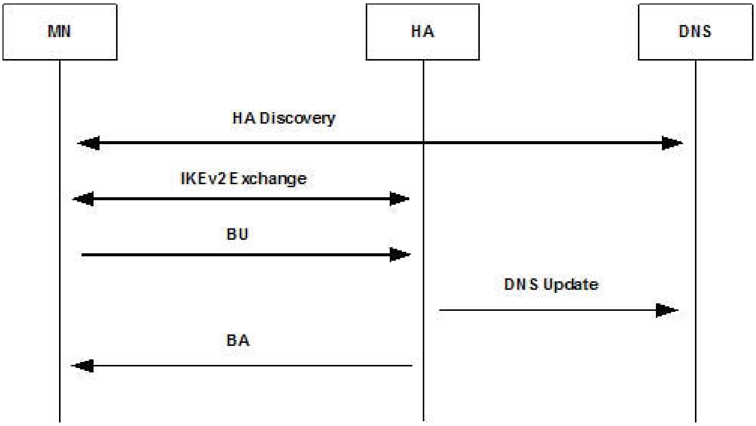


Figure 12.10. *Exchanges in the split scenario with DNS update by the HA*

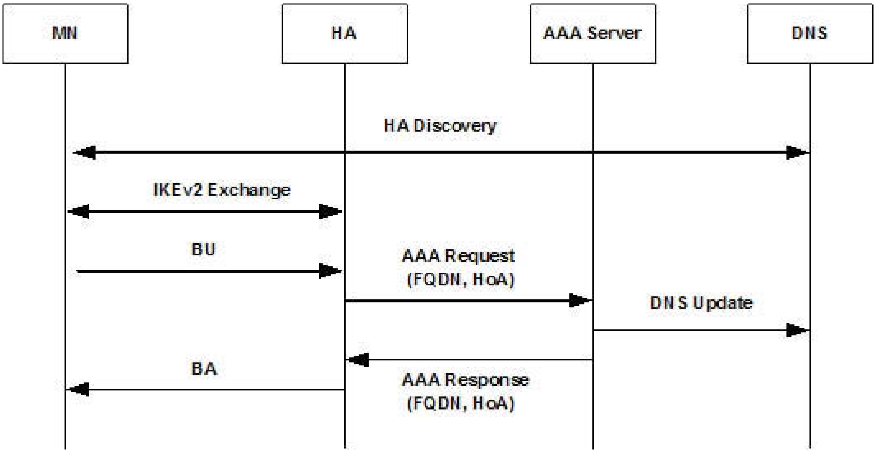


Figure 12.11. *Exchanges in the split scenario with DNS update by the AAA server*

12.3.2.3. *Integrated scenario*

The solution for this scenario is described in the IETF draft [CHO 08]. It relies on the use of DHCPv6 [RFC3315].

It reuses many components from the mechanism used for the split scenario: IPsec SA establishment between the MN and HA, HoA assignment and authentication and authorization of the MN by the MSA. However, in the integrated scenario, the HA IP address discovery relies on DHCPv6 [RFC3315] and the HA may be either in the MSP or in the network of the ASP. Figure 12.12 sums up the Mobile IPv6 bootstrapping solution based on DHCPv6, in the integrated scenario case.

In order to access the network, the MN must be authenticated by a NAS, thanks to PANA, IEEE 802.11i/802.1x or 3GPP mechanism for example. The NAS will then contact a AAA server in the ASA which is also the MSA. If the MN is authorized to use the Mobile IPv6 service, the AAA server assigns a HA in the MSP and delivers this information to the NAS. This information may be an IP address or an FQDN (domain name).

Since the MN knows that it can access the network, it performs a DHCPv6 request to obtain a HA thanks to an *Information Request* message containing the domain name where the HA must be located, ASP or MSP. The NAS, which also acts as a DHCPv6 relay, intercepts the request and inserts the information provided by the AAA server. It then forwards the request to the DHCPv6 server in the ASP.

If the MN asks for an HA in the MSP, then the DHCPv6 server extracts the information about the HA available in the DHCPv6 request sent by the NAS and includes it in the *Reply* message sent to the NAS. This latter transmits the message to the MN which now knows its HA.

If the MN asks for an HA in the ASP, then the DHCPv6 server replies with an HA that it has chosen in the *Reply* message towards the NAS. This transmits the message to the MN.

Figure 12.12 illustrates the exchange in the integrated scenario.

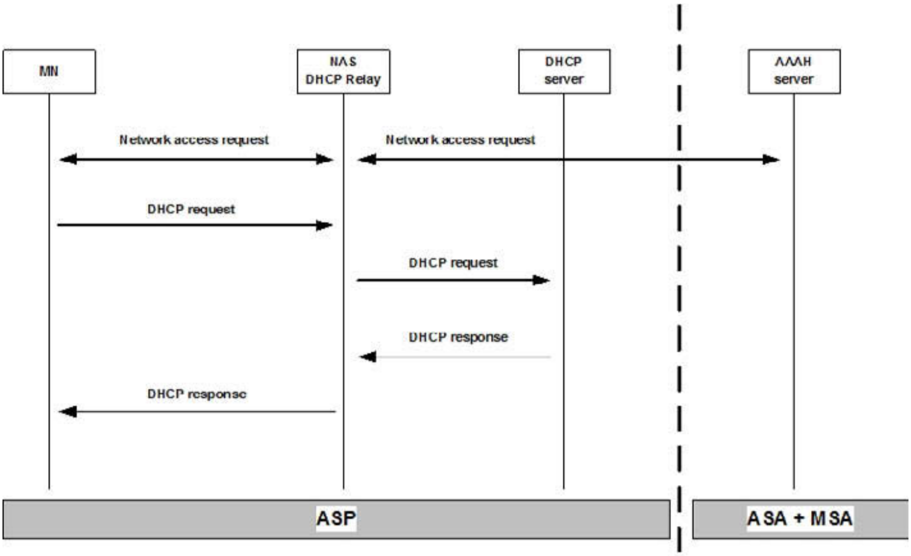


Figure 12.12. *Exchanges in the integrated scenario*

12.3.3. *Network mobility*

The concept of IPv6 mobility for a host may be extended to a router and the networks behind it. The IETF has standardized a protocol, based on MIPv6, without RO, allowing such a type of mobility which is called NEMO (*NEtwork MObility*) *Basic Support* described in the RFC 3963 [RFC3963].

The differences between MIPv6 and NEMO Basic Support are:

– the HA must manage, in addition to HoA and to CoA, *Mobile Routers* (MR): mobile network prefixes of the MR. To do this, it frequently updates a table containing the mobile network prefixes *(Prefix Table)* indexed by the MR's HoA;

– the tunneling function of the HA is based on prefixes instead of addresses;

– there are 2 modes for the BU management to the HA:

- an implicit mode where the MR sends a BU as in MIPv6,

- an explicit mode where the MR sends a BU containing which mobile networks prefixes are managed by it.

From a security point of view, the signaling is secured as in MIPv6: IPsec is used between the HA and the MR. Moreover, in NEMO Basic Support, when it receives a BU in explicit mode, the HA must check in the *Prefix Table* that the prefixes contained in this BU are really owned by the MR to prevent DoS attacks.

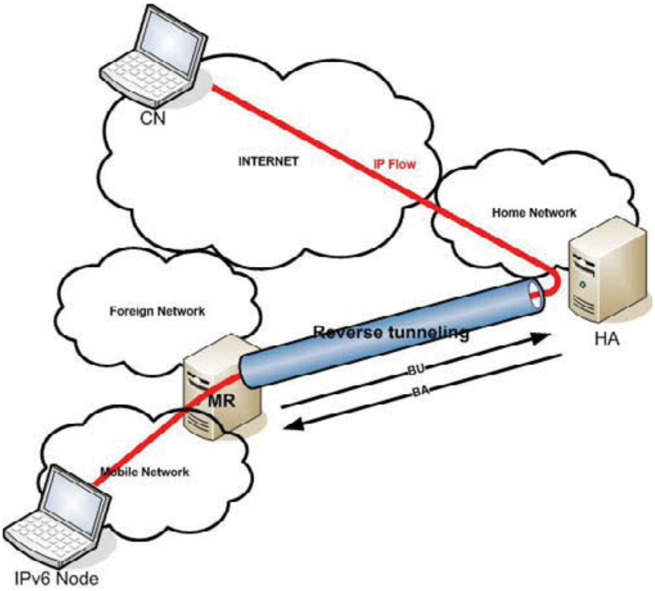


Figure 12.13. *NEMO architecture with reverse tunneling*

12.3.4. *Open security issues*

12.3.4.1. *Mobile IPv6*

The RR mechanism for the RO is actually based on “weak” security. This mechanism assumes that there is no malicious node close to the CN, which will be able to intercept the exchanged messages between the MN and the CN. In such a scenario, this malicious node would be able to generate the Kbm key and so would be able to send a fake BU to the CN instead of the MN.

A first solution, standardized and described in the RFC 4449 [RFC4449], suggests using pre-shared secrets between the MN and a CN. This secret is the Kbm key used to secure the BU and the BA. So, the RR mechanism is no longer necessary but it is recommended to check the *Sequence Number* included in the BU to avoid replay attacks. The drawback of this solution is that it is necessary for the MN and the CN to know each other before the RO process to set up the pre-shared secret.

Another solution, described in [DUP 08], suggests using IPsec to secure the BU and the BA. This makes it possible to obtain a “strong” security but, on the other hand, it is necessary to have either pre-shared secrets or a trustable infrastructure (e.g. PKI, DNSSEC).

12.3.4.2. *MIPv6 bootstrapping*

A first open issue concerns the fact that, in the “split scenario”, the HA list is stored in a DNS which is assumed to be reachable by anybody. So, with only one DNS request, people wanting to attack the MIPv6 service provided by a MSP can obtain the list of all the HAs managed by this MSP and can launch DoS attacks on them. The next point is that the MSP cannot set up a policy of HA assignment because the MN itself chooses its HA inside the HA list stored in the DNS. The IETF is actually working on a potential HA assignment solution.

The second issue concerns the solution for the “integrated scenario” which is DHCP-based. This solution “breaks” the trust link between the MN and the MSA’s AAA server for the HA assignment. Indeed, the AAA server provides the HA to the ASP’s DHCP server, which will provide it to the MN instead of providing it directly to the MN. An alternative solution has been proposed to the IETF, but not retained, and is described in [BOU 06]. This solution restored the trust link by allowing the AAA server to provide the HA in using PANA between the NAS and the MN.

12.4. Mobility with Mobile IPv4

In certain countries like the USA, but also in Europe, the deployment of IPv6 is taking longer than in some other countries like China, Japan or Korea, which arrived later to the Internet race and were the first to suffer from IPv4 address starvation. As a result, some operators, in particular mobile phone operators, found the Mobile IPv4 (MIPv4) protocol easier to deploy. Some recent works aim at adapting the NEMO protocol originally defined with MIPv6 to a MIPv4 environment.

12.4.1. *The protocol*

Like MIPv6, the MIPv4 protocol [RFC4721] aims to make a mobile reachable whatever its geographical position on the Internet. Thus, an IPv4 mobile has one home address and a CoA address in the FN, as in MIPv6. It has also been allocated a HA in its HN.

The differences with MIPv6 come from the optional support of MIPv4 in IPv4 equipment. Thus, certain improvements like RO in MIPv6 are not possible in MIPv4 because the correspondents are not required to interpret the MIPv4 messages.

Another difference holds in the IPv4 addresses starvation problem which sometimes makes the allocation of a CoA to the mobiles during their moves difficult. Thus, an entity specific to MIPv4 called an FA and localized in the FN is defined to deal with the management of the MN in the FN. Two modes for allocating IPv4 addresses are available:

– The FA CoA mode is for the FA to assign one of its own addresses to the MN through the neighbor discovery mechanism. The same address can thus be assigned to several MNs. According to the home address specified in the packets, the FA forwards the packet to the correct MN. This mode proves particularly useful in the event of IPv4 addresses starvation within the FN.

– The co-located CoA mode is for the MN to dynamically acquire an address, for example through DHCP (Dynamic Host Protocol Configuration). Each MN has its own temporary address.

In fact, the co-located CoA mode leads to a tunnel established between the HA and the MN for any data exchange, very much like MIPv6. With the FA CoA mode, the tunnel is established between the HA and the FA. The FA decapsulates the packets and relays them to MN.

The FA CoA mode assumes that the MN can discover the local FA in a foreign network. A mechanism for discovering MIPv4 agents was designed and is based on the ICMP router discovery mechanism.

The MIPv4 service is activated when the HA knows the current position of the MN (CoA). As in MIPv6, a procedure for registration of the CoA must be carried out by the MN. Two messages (of type UDP) are exchanged: *Registration Request* (equivalent to the BU) and *Registration Reply* (equivalent to the BA). If a FA is present in the FN, these messages must necessarily pass through the FA, and the FA maintains a cache of all the visited MNs and keeps control on them within its visited domain.

When the registration is completed, a correspondent can send packets to the home address of the MN and the HA and then encapsulate these packets into the tunnel towards the MN. When the MN answers, it then has two possibilities: either it transmits its packets directly to the CN by using its home address, or it sends all of its traffic through the tunnel via its HA (reverse tunneling) [RFC3024]. However, the first solution is likely to harm MN connections because it is similar to spoofing attacks with packets going out of the FN and carrying the home address of the MN as the source address. Indeed, firewalls are used to block such packets.

In MIPv4, the MN is not required to know its HA. A procedure for dynamically discovering its HA exists [RFC4433]. For the MN, the procedure consists of transmitting a *Registration Request* message towards the network (generally towards the FA) with an extension specifying that no HA is known by the MN. Then, according to the NAI *(Network Address Identifier)*, the FA allocates a HA dynamically and tunnels the message towards the selected HA. The procedure then continues normally.

12.4.2. *Security*

The MIPv4 protocol is extremely sensitive to hijacking and traffic eavesdropping of MNs. For example, a node can spoof a FA and thus can see all the traffic of local MNs; a terminal can also transmit false *Registration Request* messages in order to redirect all the traffic of a MN towards itself. It is thus essential to protect the various mechanisms related to MIPv4.

The FA agent discovery mechanism is based on IPsec and proposes that the source of agent discovery messages is authenticated by an HA SA-IPsec, but no information on how such a SA-IPsec can be established is given.

The registration mechanism to the HA requires us to authenticate all *Registration Request* and *Registration Reply* messages. For this, it is necessary that HA and MN share a mobility security association so that they can authenticate the source of these registration messages; the MIPv4 extension carrying out this authentication is called *Mobile-Home Authentication*. It is also envisaged to secure against fake FAs, but this requires that the FA and the MN, respectively the FA and the HA, share a security association and that they prove the source of the registration messages thanks to the addition of the *Mobile-Foreign Authentication*, respectively *Foreign-Home Authentication* extension. It should be noted that the authentication service is not enough to secure against possible traffic hijacking. For instance, some *Registration Request* messages can be replayed after the MN moved. The detection of replays is possible thanks to the *identification* field which must be different in each query issued by the MN and sent back as identical by the HA in its response.

The difficulty of Mobile IPv4 is to set up such a SA between the MN and the FA, and between the FA and the HA which *a priori* do not know each other and do not belong to the same administration domain. The other difficulty is to ensure that an SA is agreed between the MN and the HA in case a HA is dynamically allocated to the MN. To manage the security and SAs associated with the various MIPv4 elements, a solution based on the Diameter MIPv4 application was defined [RFC4004]. The fundamental role of this application [RFC4004] is to allow an access network operator to control the access of the MN to its network by setting up AAA functions . The access network operator can be either the operator to which the MN subscribed the MIPv4 service or any other operator which would have concluded an agreement with the access network operator.

As shown in Figure 12.14, the MN operations are not disturbed too much because MIPv4 is still used to perform exchanges with the FA. However, it must add several extensions to its Registration Request message like its NAI (*Network Access Identifier*) used as Mobile-Foreign Authentication. The FA forwards this message to its local Diameter server (AAAL) encapsulated in an AMR *(AA-Mobile-Node-Request)* Diameter message. This message follows the classical Diameter procedure, that is, to first go through AAAL which, itself, transmits to AAA Home (AAAH) server. The choice of the AAAH is made according to the NAI provided by the MN. The authenticity of the *Registration Request* message is then verified by the AAAH server which forwards this message in a HAR *(Home-Agent-MIP-request)* Diameter message to the HA which was allocated by AAAH or designated by MN (in its MIPv4 request). The HA then returns the Registration *Reply* MIPv4 message in a HAA *(Home-Agent-MIP-Answer)* Diameter message. AAAH can also allocate a home address to the MN which it will push into the Diameter messages. AAAH also has the task of participating in the establishment of a mobility SA between the MIPv4 participants (MN, HA, F). Parameters (i.e. nonces) useful for the generation of SA are communicated to the HA and the FA in HAR *(Home-Agent-MIP-Request)* and AMA *(AA-Mobile-Node-Answer)* Diameter messages. The parameters useful for MN (nonces, but also allocated home address, HA@) are encapsulated into the *Registration Reply* IPv4 message which is itself encapsulated into the AMR message. Once the parameters are received, the MIPv4 participants are able to generate a shared cryptographic key in accordance with the standard [RFC3957] and to build their SAs between the MN and the HA, the MN and the FA and the FA-HA. Note that the only constraint that needs to be provided by the architecture for establishment of these SAs is to share a SA at the origin with AAAH.

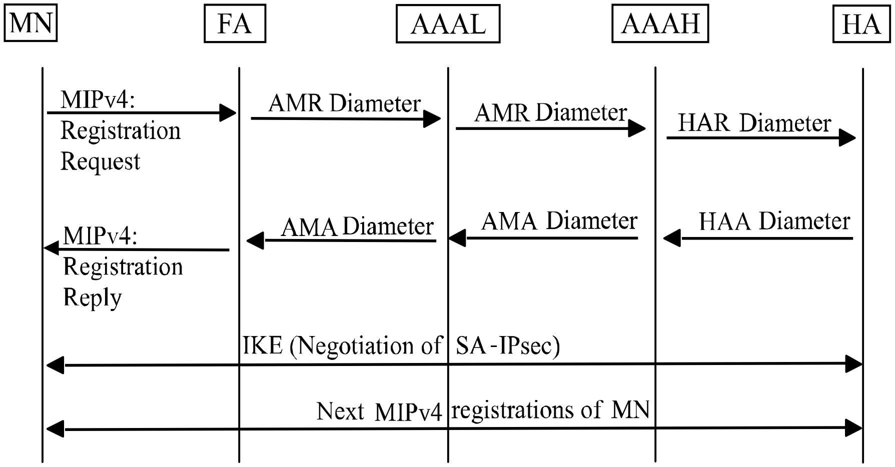

MN
FA
AAAL
AAAH
HA
MIPv4: Registration Request
AMR Diameter
AMR Diameter
HAR Diameter
MIPv4: Registration Reply
AMA Diameter
AMA Diameter
HAA Diameter
IKE (Negotiation of SA-IPsec)
Next MIPv4 registrations of MN

Figure 12.14. *Exchanges by the Diameter MIPv4 application*

12.5. Mobility with MOBIKE

It is often the case that a mobile has several IP addresses simultaneously. In the event that the MN moves from one AR to another, a new NCoA address is allocated to it by the new AR (NAR). If the MN does not move too quickly, both addresses of connectivity (NCoA and PCoA allocated by old AR: PAR) will be allocated to it during the handover period of time, which covers the time spent to discover the new FN and to switch the communications from PAR to NAR. This handover can be carried out between two networks of the same technology (802.11) or between two networks of different technologies. A mono technology handover (or horizontal handover) is often done after the mobile moves. A multi-technology handover (or vertical handover) is rather planned to reduce the costs, to counteract possible failures of one access network or to improve quality of transmission (by affording higher bitrates). For example, we can imagine a mobile connected to the WiMAX network which returns to the HN of its owner and which then connects automatically on the local 802.11 network to reduce the costs of the communications by using the fixed price wireless ADSL. Note that the vertical handover assumes that the mobile is equipped with several physical interfaces (one for each access technology: Ethernet card, WLAN interface, GPRS adapter, Bluetooth interface, etc.), each one potentially having an IP address. The property of having several of these interfaces simultaneously active is well known under the name of *multihoming*.

The challenging problem of handover is not the management of the handover itself, but its introduction into Mobile IPv4 or Mobile IPv6 environments. Indeed, with these mobility protocols, the MN is held to maintain an IPsec tunnel with its HA. However, this tunnel, which is managed by the IKE, protocol is identified partly by the IP addresses of the ends of the tunnel, i.e. the CoA address of the mobile and the address of the HA. Thus, in the original version of IKE, if one of these addresses is modified, the tunnel is destroyed and the MN connections are lost.

To introduce flexibility into the management of IPsec tunnels and to avoid any connection disruption during the reassignment of an IP address to the MN, the IETF Working Group MOBIKE *(IKEv2 Mobility and Multihoming)* defined extensions [RFC4555, RFC4621] for the IKEv2 protocol for both the IPv4 and IPv6 environments. One of the extensions makes it possible for the MN to inform, in the very first exchanges, distant VPN equipment (HA) of its capacity to manage MOBIKE (MOBIKE_SUPPORTED information), to inform it of all the IP addresses to which it is reachable (ADDITIONAL_IPv4_ADDRESSES and ADDITIONAL_IPv6_ADDRESSES), and then in the event of moves, to inform the VPN gateway when to switch on the new IP address (UPDATE_SA_ADDRESSES information).

Figure 12.15 shows the IKEv2 exchanges between the MN and the HA and how the MOBIKE extension helps to manage the security associations of IKEv2 (SA_IKE) and IPsec (SA_IPsec) levels. The MN makes itself known with the PCoA address, but it can also provide the list of its other addresses. The security associations are thus registered under PCoA and @HA identifiers. During a move and after obtaining a NCoA, the MN can notify the HA of this additional address or can directly ask the HA to update the security associations with the NCoA. Before updating associations, the HA must check that the MN is accessible through the NCoA by sending a cookie that will then be returned by the MN.

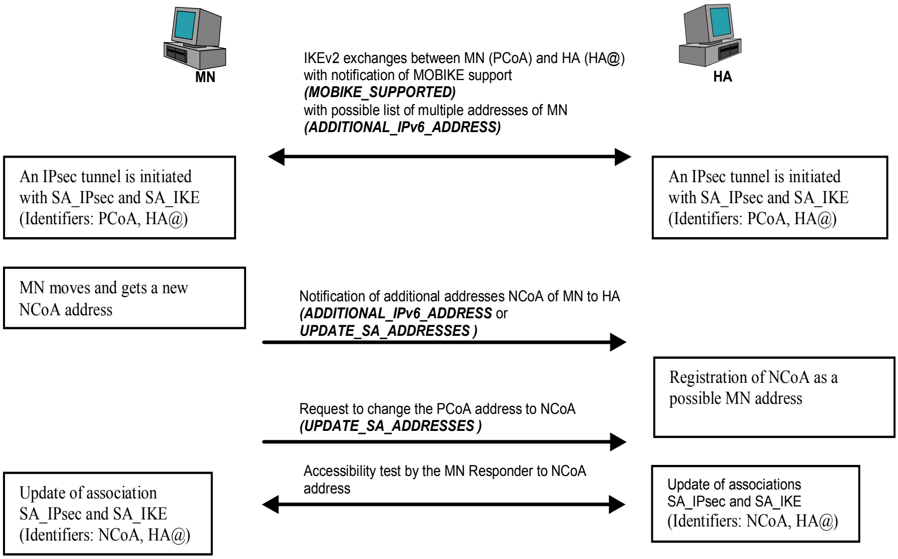

MN
HA
IKEv2 exchanges between MN (PCoA) and HA (HA@)
with notification of MOBIKE support
(MOBIKE_SUPPORTED)
with possible list of multiple addresses of MN
(ADDITIONAL_IPv6_ADDRESS)
An IPsec tunnel is initiated with SA_IPsec and SA_IKE (Identifiers: PCoA, HA@)
An IPsec tunnel is initiated with SA_IPsec and SA_IKE (Identifiers: PCoA, HA@)
MN moves and gets a new NCoA address
Notification of additional addresses NCoA of MN to HA
(ADDITIONAL_IPv6_ADDRESS or
UPDATE_SA_ADDRESSES)
Registration of NCoA as a possible MN address
Request to change the PCoA address to NCoA
(UPDATE_SA_ADDRESSES)
Accessibility test by the MN Responder to NCoA address
Update of association SA_IPsec and SA_IKE (Identifiers: NCoA, HA@)
Update of associations SA_IPsec and SA_IKE (Identifiers: NCoA, HA@)

Figure 12.15. *IKEv2 exchanges with MOBIKE*

Moreover, MOBIKE plans to detect and cope with some connectivity that might occur between two devices. New informational IKEv2 messages are defined, and one of the devices challenges the other. In the event of a connection failure, another test is launched on a previously known secondary address, and if a better connectivity is detected, an update of the address of the device is then carried out using MOBIKE.

All the IKEv2 messages benefit from the SA_IKE protection (encryption, authenticity, integrity protection and replay detection), whether their purpose is to test connectivity or to update an address. Thus, it appears highly difficult to redirect MN's traffic by sending a request to update the MN's address.

With the MOBIKE extension, the MN manages its mobility by informing the HA of its new localization in the course of its moves.

12.6. IP mobility with HIP and NetLMM

In addition to MIPv6-based protocols, the IETF is also working on alternative mobility protocols where the philosophy is different from the MIPv6 protocols. Thus, with the HIP protocol, a natively secure protocol, the IETF works on the identifier/locator split for an IP node, so that with the NetLMM protocol, the IETF studies a mechanism where the intelligence for the mobility is located in the access network. The following two sections describe these protocols.

12.6.1. *HIP*

Currently, one node on the Internet uses two different naming spaces: the IP address naming space and the naming space of the domain name, also known as the Domain Name System (DNS) [RFC1034] [RFC1035]. The DNS binds a domain name, which is a human understandable name to an IP address, which is used for routing purposes.

It appears that the IP address is used both as an identifier and as a locator. An identifier aims to distinguish one device from another, whereas a locator aims to provide the localization of such a device. If we consider a home computer, we can easily understand that a locator can be used as an identifier. The device is identified by its location. On the other hand, if we consider a mobile device, we also understand that such a device still has the same identity wherever it is. Identity is independent of localization. One way is to split the network layer and consider network Host Identifiers (HI). Such an identity must be able to communicate without considering their localization. Such a layer has been introduced between layer 3 and layer 4, and so it is called layer 3.5.

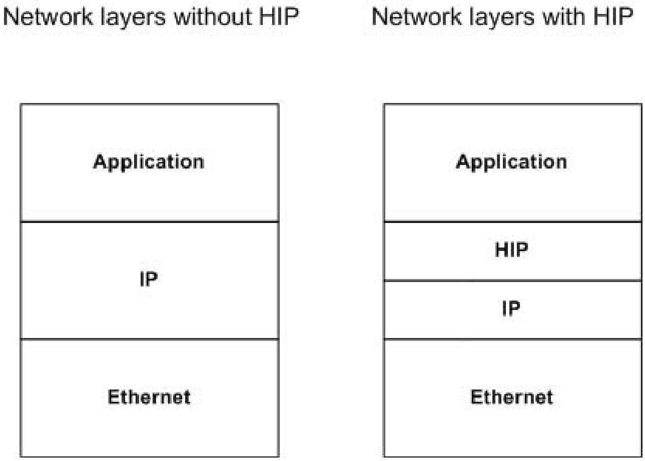


Figure 12.16. *Layer 3.5 with HIP*

On the other hand, security is one of the main preoccupations of the Internet community, and even though it was until now considered more or less as an option that required heavy configuration, the Host Identity Protocol (HIP) [RFC4423] [RFC5201], includes security in the base design of the protocol. The nature of the HI is a public key, and thus is called a cryptographic identifier. Such identifiers provide native proof-of-ownership for communication and avoid man-in-the-middle attacks. Since IPv6 network communication requires a 128 bit identifier, which is the size of an IPv6 address, the Host Identity Tag (HIT) is used as an identifier. The HIT is of a fixed size and is the hash of the public key. For IPv4, the 32 bit version is called the Local Scope Identifier (LSI).

Association between two HITs requires a four packet exchange. The Initiator initiates the communication with the Responder. The Initiator is sending an I1 packet with both the HITs of both ends. The Responder sends back a R1 packet with a puzzle, Diffie-Hellman (DH), parameters its public keys and a signature of the packet. The puzzle is only a way to avoid DoS attacks. It is a problem that requires calculation to be solved, but whose solution is easy to check. The Initiator computes the solution and sends an I2 packet, with the solution, the complementary DH parameters, its public key and a signature. The Responder checks the puzzle solution first, and then the signature, before sending a R2 packet that confirms the association.

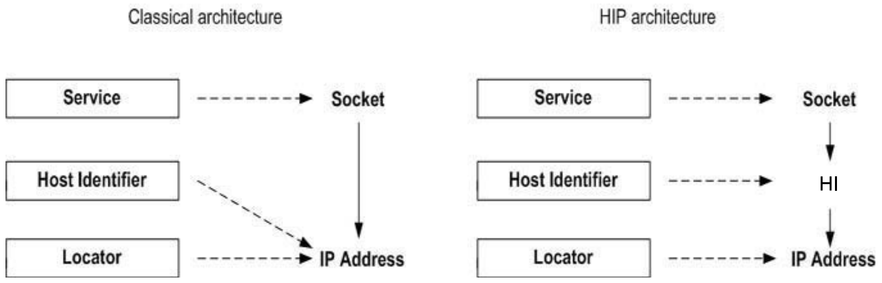


Figure 12.17. *Identifier/locator split in HIP*

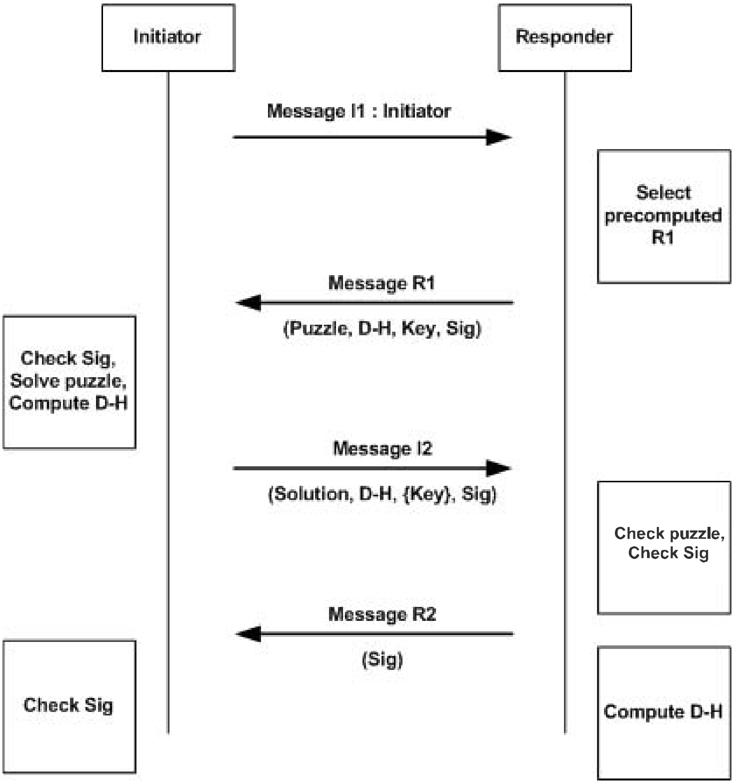


Figure 12.18. *HIP exchanges*

HIP associations consider HITs that are above the layer with locators, but packets still use locators to route packets to a destination. A binding between the HIT and the locator must be done. The DNS is used to bind domain names and IP addresses, and can be also used for the HIT to IP address binding [RFC5205]. Nevertheless, the DNS has not been designed to provide highly dynamic bindings, which is why Rendezvous Servers (RVS) [RFC5203] [RFC5204] can help. For a given HIT or HI, the DNS can redirect to a dedicated server (RVS) that hosts the binding and can send back the IP address corresponding to the HIT. By decoupling the transport layer from the internetworking layer, a node is able to change its IP address without breaking the connection. It could also manage, transparently to the transport layer, an IP address pool. The mobility and multihoming possibilities for the HIP are described in RFC 5206 [RFC5206].

The HIP is a protocol that provides a new type of communication. The HIP protects such communications against spoofing attacks by providing a proof-of-ownership. We must notice that the base HIP does not proceed to authentication of the peers. It provides protection against DoS, and security material to protect the communication. In fact, HIP provides means to establish an ESP communication between the peers [RFC5202]. The ESP protection is established between two HITs, which means that the IP packets are ESP packets as described in RFC 4303 [RFC4303], only if IP addresses in the header are replaced with HITs. The latest packet is not routable, since the HITs are only identifiers, but this is the one that is sent to/from the HIP layer from/to the IPsec layer. Considering the tunnel with implicit ends leads to a new IPsec mode: the Bound End-to-End Tunnel (BEET) mode [MEL 08]. The protection differs in that sense from IPsec whose core purpose is to negotiate security for the IP layer. One of the advantages of the HIP is that security is part of the protocol. On the other hand, IPsec provides means to authenticate the peers.

12.6.2. *NetLMM*

The IETF is also working on an IP mobility solution where the mobility management is done by the network and not by the end node. This approach avoids adding a Mobile IP stack in a terminal and this would allow a rapid deployment of such solutions. Obviously the network operator has to deploy new entities/functionalities in order to allow such mobility.

The IETF NetLMM (Network-based Localized Mobility Management) Working Group works on this approach in the micro-mobility area.

When the MN enters a NetLMM network, it obtains an IPv6 address that it will keep during its movements in the NetLMM domain. An entity called a *Local Mobility Anchor* (LMA) is in charge of redirecting packets for the MN towards the AR, called *Mobile Access Gateway* (MAG), in charge of this MN. Since the MN enters a subnetwork managed by a MAG, the MAG informs the LMA. The interface between an MN and a MAG is described in [LAG 08] while the mechanism used between a MAG and a LMA is described in the RFC 5213 [RFC5213].

The main security problem is the MN's arrival discovery by a MAG. Indeed, this discovery is based on the neighbor discovery mechanism. Thus, it is recommended that an MN compatible with NetLMM uses a SEND/CGA protection mechanism in order to guarantee the integrity and uniqueness its IPv6 address.

Moreover, it is necessary to secure the information exchanged between MAGs and LMAs of a NetLMM network. For this, the use of IPsec is recommended.

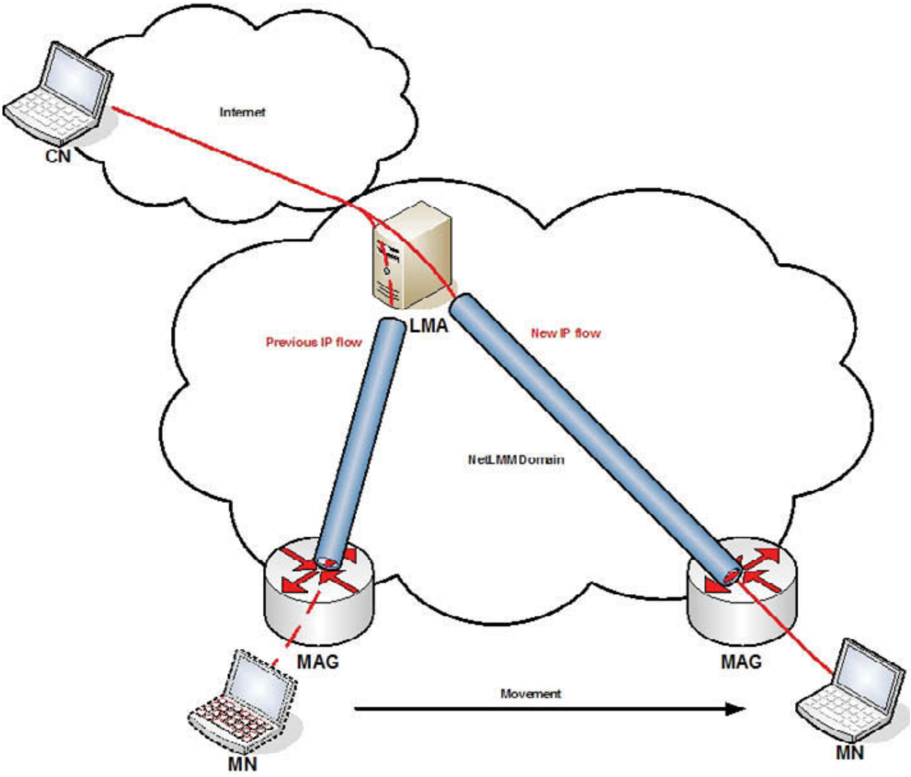


Figure 12.19. *NetLMM architecture*

12.7. Conclusions

Works on IP mobility (IPv4 and IPv6) have been strongly enriched in recent years. Several optimizations were brought to the basic protocols. FMIPv6 and HMIPv6 were defined to optimize the micro-mobility for MIPv6 (where the ARs of the successive FNs do not belong to the same administration domain). The security implementation problems of these latter solutions lie in the establishment of IPsec security associations between the MN and the FN equipment which do not know each other, do not belong to the same administration domain (they do not trust each other originally) and whose interactions are of short duration. Traditional security solutions like PKI or the DNS are not appropriate in this context. Solutions based on SEND and cryptographic identifiers are also considered, but their security levels seem insufficient.

To fulfill the requirements of MIPv6 deployment, some work was carried out on the automation of MIPv6 service bootstrapping. Several scenarios were then elaborated according to whether the mobility service is provided by the operator of the access network or a MIPv6 service provider. This procedure of automation is also planned to partly solve the problems of establishing IPsec SA between the MN and the access network.

MOBIKE appears to be a secure solution for the micro-mobility of the MN because MOBIKE does not ask the FN to manage the micro-mobility; all the management is done by the HA thanks to the IKEv2 protocol and its MOBIKE extension. On the other hand, there is still an open problem that relates to the interconnection of this mechanism to the AAA/PANA access control; indeed, at each moment, the AAA server of the FN must know the positioning of the visiting MN in order to conduct periodic re-authentication of the MN.

In other approaches, the MN does not change its address while moving in a FN (NetLMM); the management of mobility is done by the FN itself. Another solution, HIP, distinguishes the identification address and the localization address; and defines a centralized rendez-vous server to perform the binding between the two addresses.

All these complementary solutions for IP mobility have rather significant flaws. It is difficult to say which one will be the final solution adopted for 4G networks. Strategic choices must be taken, for instance, the level of MN implication in the management of their mobility, but also the effective deployment of DNSSEC, and its targeted security level. Finally, further work is still to be developed on the optimization of handover performances, and how to make the handover as transparent as possible with respect to underlying technologies in order to avoid any interruption of connection. In addition to the classical problems of performance (handover delay) and bandwidth management during handover, we will have to solve the problems of monitoring and implementation of security. The most critical case that will need to be solved is the vertical handover carried out between two operators.

12.8. Glossary

@HA	Home Agent’s Address
AAA	Authentication, Authorization, Accounting
AR	Access Router
ARP	Address Resolution Protocol
ASA	Access Service Authorizer

ASP	Access Service Provider
BA	Binding Acknowledgment
BU	Binding Update
CGA	Cryptographically Generated Addresses
CN	Correspondent Node
CoA	Care-of Address
CoTI	Care-of Test Init
CoT	Care-of Test
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name Server
DNSSEC	DNS Security
ESP	Encapsulating Security Payload
FBack	Fast Binding Acknowledgment
FBU	Fast Binding Update
FMIPv6	Fast Handovers for MIPv6
FNA	Fast Neighbor Advertisement
FQDN	Fully Qualified Domain Name
HA	Home Agent
HAck	Handover Acknowledge
HI	Handover Initiate (for FMIPv6)
HI	Host Identity (for HIP)
HIP	Host Identity Protocol
HIT	Host Identity Tag
HMIPv6	Hierarchical MIPv6 Management
HoA	Home Address
HoTI	Home Test Init
HoT	Home Test
IETF	Internet Engineering Task Force
LCoA	Local Care-of Address
LMA	Local Mobility Anchor
LSI	Local Scope Identifier
MAG	Mobile Access Gateway
MAP	Mobility Anchor Point
MIP	Mobile IP

MIPv6	Mobile IPv6
MN	Mobile Node
MR	Mobile Router
MSA	Mobility Service Authorizer
MSP	Mobility Service Provider
NAR	New Access Router
NAS	Network Access Server
NCoA	New Care-of-Address
ND	Neighbor Discovery
NEMO	Network Mobility
NetLMM	Network-based Localized Mobility Management
PANA	Protocol for carrying Authentication for Network Access
PAR	Previous Access Router
PCoA	Previous Care-of Address
PKI	Public Key Infrastructure
PrRtAdv	Proxy Router Advertisement
RCoA	Regional Care-of Address
RO	Route Optimization
RR	Return Routability
RtSolPr	Router Solicitation for Proxy Advertisement
RVS	Rendez-Vous Server
SA	Security Association
SEND	SEcure Neighbor Discovery

12.9. Bibliography

[BOU 06] BOURNELLE J. (Editor), LAURENT-MAKNAVICIUS M., COMBES JM., *Using PANA in the Mobile IPv6 Integrated Case*, draft-bournelle-pana-mip6-01.txt, August 2006 (expired).

[CHO 08] CHOWDHURY K. (Editor), YEGIN A., *MIP6-bootstrapping via DHCPv6 for the Integrated Scenario*, draft-ietf-mip6-bootstrapping-integrated-dhc-06.txt, April 2008 (work in progress).

[DUP 07] DUPONT F., *A Note about 3rd Party Bombing in Mobile IPv6*, draft-dupont-mipv6-3bombing-05.txt, January 2007 (work in progress).

[DUP 08] DUPONT F., COMBES JM., *Using IPsec between Mobile and Correspondent IPv6 Nodes*, draft-ietf-mip6-cn-ipsec-08.txt, August 2008 (work in progress).

[LAG 08] LAGANIER J., NARAYANAN S., MACCAN P., *Interface Between a Proxy MIPv6 Mobility Access Gateway and a Mobile Node*, draft-ietf-netlmm-mn-ar-if-03.txt, February 2008 (work in progress).

[MEL 08] MELEN J., NIKANDER P., *A Bound End-to-End Tunnel (BEET) Mode for ESP*, draft-nikander-esp-beet-mode-09.txt, August 2008 (work in progress).

[RFC1034] MOCKAPETRIS P., *Domain Names – Concepts and Facilities*, RFC 1034, November 1987.

[RFC1035] MOCKAPETRIS P., *Domain Names – Implementation and Specification*, RFC 1035, November 1987.

[RFC2782] GULBRANDSEN A., VIXIE P., ESIBOV L., *A DNS RR for Specifying the Location of Services (DNS SRV)*, RFC 2782, February 2000.

[RFC3024] MONTENEGRO G., *Reverse Tunneling for Mobile IP, Revised*, RFC 3024, January 2001.

[RFC3315] DROMS R. (Ed.), BOUND J., VOLZ B., LEMON T., PERKINS C., CARNEY M., *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*, RFC 3315, July 2003.

[RFC3756] NIKANDER P., KEMPF J., NORDMARK E., *IPv6 Neighbor Discovery (ND) Trust Models and Threats*, RFC 3756, May 2004.

[RFC3775] JOHNSON D., PERKINS C., ARKKO J., *Mobility Support in IPv6*, RFC 3775, June 2004.

[RFC3776] ARKKO J., DEVARAPALLI V., DUPONT D., *Using IPsec to Protect Mobile IPv6 Signaling Between Mobile Nodes and Home Agents*, RFC 3776, June 2004.

[RFC3957] CALHOUN P., *Authentication, Authorization, and Accounting (AAA) Registration Keys for Mobile IPv4*, RFC 3957, Mars 2005.

[RFC3963] DEVARAPALLI V., WAKIKAWA R., PETRESCU A., THUBERT P., *Network Mobility (NEMO) Basic Support Protocol*, RFC 3963, January 2005.

[RFC3971] ARRKO J., KEMPF J., ARRKO J., ZILL B., ARRKO J., NIKKANDER P., *SEcure Neigbhor Discovery (SEND)*, RFC 3971, March 2005.

[RFC3972] AURA T., *Cryptographically Generated Addresses (CGA)*, RFC 3972, March 2005.

[RFC4004] CALHOUN P., JOHANSSON T., PERKINS C., HILLER T., MCCANN P., *Diameter Mobile IPv4 Application*, RFC 4004, August 2005.

[RFC4033] ARENDS R., AUSTEIN R., LARSON M., MASSEY D., ROSE S., *DNS Security Introduction and Requirements*, RFC 4033, March 2005.

[RFC4301] KENT S., SEO K., *Security Architecture for the Internet Protocol*, RFC 4301, December 2005.

[RFC4303] KENT S., *IP Encapsulating Security Payload (ESP)*, RFC 4303, December 2005.

[RFC4306] KAUFMAN C. (Ed.), *Internet Key Exchange (IKEv2) Protocol*, RFC 4306, December 2005.

[RFC4423] MOSKOVITZ R., NIKANDER P., *Host Identity Protocol (HIP) Architecture*, RFC 4423, May 2006.

[RFC4433] KULKARNI M., PATEL A., LEUNG K., *IPv4 Dynamic Home Agent (HA) Assignment,* RFC 4433, March 2006.

[RFC4449] PERKINS C., *Securing Mobile IPv6 Route Optimization Using a Static Shared Key,* RFC 4449, June 2006.

[RFC4555] ERONEN P., *IKEv2 Mobility and Multihoming Protocol (MOBIKE)*, RFC 4555, June 2006.

[RFC4621] KIVINEN T., TSCHOFENIG H., *Design of the IKEv2 Mobility and Multihoming (MOBIKE) Protocol*, RFC 4621, August 2006.

[RFC4640] PATEL A., GIARETTA, G., *Problem Statement for Bootstrappping Mobile IPv6 (MIPv6),* RFC 4640, September 2006.

[RFC4721] PERKINS C., CALHOUN P., BHARATIA J., *IP Mobile IPv4 Challenge/Response Extensions (Revised)*, RFC 4721, January 2007.

[RFC4861] NARTEN T., NORDMARK E., SIMPSON W., SOLIMAN H., *Neighbor Discovery for IP Version 6 (IPv6)*, RFC 4861, September 2007.

[RFC4877] DEVARAPALLI V., DUPONT F., *Mobile IPv6 Operation with IKEv2 and the Revised IPsec Architecture*, RFC 4877, April 2007.

[RFC4941] NARTEN T., DRAVES R., KRISHNAN S. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6,* RFC 4941, September 2007.

[RFC5026] GIARETTA G., KEMPF J., DEVARAPALLI V., *Mobile IPv6 Bootstrapping in Split Scenario,* RFC 5026, October 2007.

[RFC5201] MOSKOWITZ R., NIKANDER P., JOKELA P. (Eds.), HENDERSON T., *Host Identity Protocol,* RFC 5201, April 2008.

[RFC5202] JOKELA P, MOSKOWITZ R., NIKANDER P., *Using ESP Transport with HIP*, RFC 5202, April 2008.

[RFC5203] LAGANIER J., KOPONEN T., EGGERT L., *Host Identity Protocol (HIP) Registration Extension*, RFC 5203, April 2008.

[RFC5204] LAGANIER J., EGGERT L., *Host Identity Protocol (HIP) Rendezvous extension*, RFC 5204, April 2008.

[RFC5205] NIKANDER P., LAGANIER J., *Host Identity Protocol (HIP) Domain Name System (DNS) Extensions*, RFC 5205, April 2008.

[RFC5206] HENDERSON T., NIKANDER P., VOGT C., ARKKO J., *End-Host Mobility and Multihoming*, RFC 5206, April 2008.

[RFC5213] GUNDAVELLI S., DEVARAPALLI V., CHOWDHURYK K., PATIL B., *Proxy Mobile IPv6,* RFC 5213, August 2008.

[RFC5268] KOODLI R. (Ed.), *Fast Handovers for Mobile IPv6*, RFC 5268, June 2008.

[RFC5269] KEMPF J., KOODLI R., *Distributing a Symmetric FMIPv6 Handover Key using SEND*, RFC 5269, June 2008.

[RFC5380] SOLIMAN H., CASTELLUCCIA C., EL MALKI K., BELLIER L., *Hierarchical Mobile IPv6 Mobility Management (HMIPv6)*, RFC 4140, October 2008.

This page intentionally left blank

Chapter 13

Security in Ad Hoc Networks

13.1. Introduction

Mobile ad hoc networks are a special set of wireless networks. They have very particular features such as high mobility, multi-hop routing and the absence of any fix infrastructure; these networks enable the deployment of communication networks at a low cost. However, they have a disadvantage compared with classic networks: vulnerability.

This chapter details the problems in the field of ad hoc networks. First, we present the features of these networks and we review the main approach in multi-hop routing. Then we present the different vulnerabilities that threaten the routing process and how they could be exploited, describing several specific attacks.

Later, we conclude by showing the different security solutions presented in recent years to prevent attacks and maintain an adequate security level.

13.2. Motivations and application fields

13.2.1. *Motivations*

With the apparition of mobile telephony services, wireless networks become an unprecedented success in recent years. The considerable equipment evolution of equipment and also the liberation of regulation in the use of radio bands have

Chapter written by Jean-Marie ORSET and Ana CAVALLI.

enabled the deployment of different architectures that answer the different needs of the 21st century. Thus, the development of GSM, UMTS and Wi-Fi technologies enable users to have advanced means of communication while benefiting from a comfortable mobility. Alongside the development of these new architectures, new needs and new practices have also appeared. Thus, each user now wishes to be able to reach his personal data or some other information in the location where he is. In the majority of cases (GSM, WAP, UMTS, Wi-Fi), the communication is supported by an architecture which is only partially wireless. The user becomes connected thanks to a connection without wires, but via an access point which remains fixed (Figure 13.1). Consequently, in order to be connected to the network, the user must be in the coverage area of one of these access points. Thus, the wireless structure must lay out a great number of access points to offer a sufficient density and then meet the growing needs of the users in term of bandwidth and also to maximize the network coverage. However, such a deployment often comes at a high price and can cause consequent delay times, while certain infrastructures do not have vocation to remain for a long period of time: it can be necessary in particular circumstances to offer connectivity where the lifetime is know in advance, limited in time. In the same way, it is not always technically or physically possible to deploy an access point in certain specific geographical locations (difficult access, devastated zone, etc.).

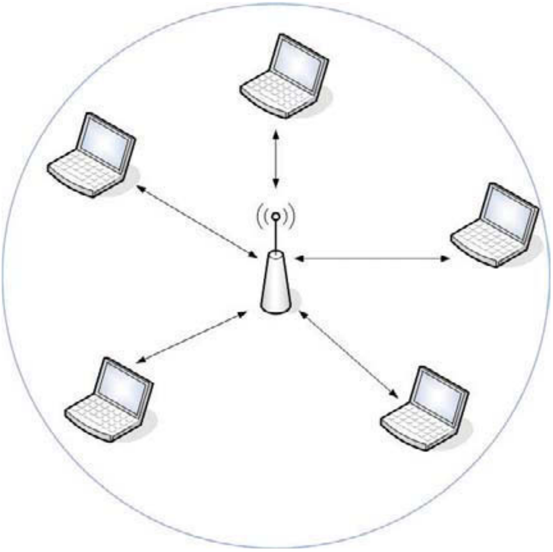


Figure 13.1. *Wireless network with fixed infrastructure*

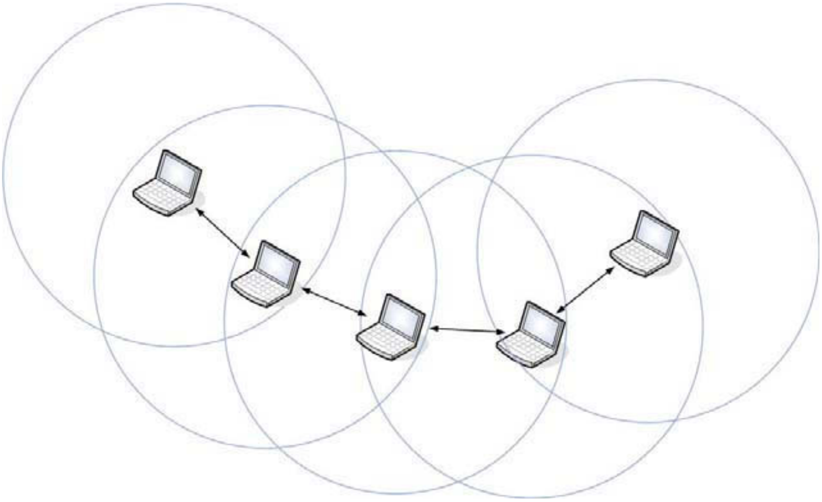


Figure 13.2. A*d hoc network with multi-hop routing*

A solution for this type of problem is to consider a wireless network where mobility is not only related to the users but is directly related to the infrastructure per se. Such a network is not made up of fixed access points but is, on the contrary, composed of entirely mobile entities. These entities (also called nodes) can communicate with each other using radio waves and when two of them are too distant to communicate directly, they use other nodes which are in charge of relaying the packets from the transmitter to the destination (Figure 13.2).

This concept can be seen as a network with a flat architecture, in the sense where all the communicating entities are equivalent and could, according to the needs, sometimes be used as customers (to emit or receive packets) and sometimes as routers (to relay packets between two other nodes). Such networks are called ad hoc networks, i.e. networks specifically dedicated to an environment without a fixed infrastructure. To be adapted to the absence of a fixed router, specific routing protocols must be employed; this is the reason why the ad hoc networks (also called MANET for *Mobile Ad Hoc Networks*) use a multi-hop routing to convey the packets. From the strong mobility of the entities, the routing protocols also have to be adapted to the important fluctuations of connectivity: links can appear or disappear constantly, become successively one-way, bidirectional and offer disparate capacities. From these various characteristics (heterogenity of the links, temporal entities, entirely distributed routing protocols), ad hoc networks can be seen as the ultimate incarnation of the concepts which justified the creation of the Internet network in the 1960s. In fact, they constitute a more flexible and light subset of wireless networks.

The advantages of ad hoc networks are numerous: they allow the installation of networks where the nodes are able in an instant, with very little human intervention and at a lower cost, to initiate communication and to exchange information.

13.2.2. *Applications*

The motivations that lead to the development of applications for ad hoc networks can be physical (impossibility of deploying a wired infrastructure) or economic. Historically, these networks were built from a military point of view. Actually, still today, a lot of research in the field is financed by the US army and navy [NRL]. In France, the army also considers the use of ad hoc networks to facilitate communication on certain operation scenarios [DGA, PLE 04].

In another field, the use of sensor networks constitutes an application that is completely adapted for ad hoc networks. Indeed, such networks must sometimes be deployed in inaccessible zones, in which any human intervention must be reduced to the bare minimum. Also, recourse to an ad hoc routing makes it possible to deal with the physical limitations imposed by the wired infrastructures (wiring of the station basic, carried radio operator field, etc.).

Beyond the purely scientific or military fields, the possibilities offered by ad hoc networks are also of interest to industry. In the automobile industry for instance, multi-hop routing is completely adapted for use within a network made up of vehicles. Considering the vehicles circulating on a highway, one of the objectives could be to prevent all the cars approaching a point of the highway with special conditions (such as traffic jam, accident), or also to disseminate information about the circulation condition on the route. Here, recourse to an ad hoc network would make it possible to avoid the expensive investment in the deployment of a complete infrastructure network over the entire highway.

From the mobile telephony operators' side, the current trend consists of extending the deployment of the 802.11 standard, in ad hoc mode, in order to make it work with simple telephones. Thus, some of the main actors of the market such as France Telecom and NTT DoCoMo now consider ad hoc networks in order to create new services with added value for their subscribers.

Finally, more and more applications are motivated by purely economic aspects. Thus, we can note an increasing use of ad hoc networks within the context of citizen's networks [LSF, SW]. Also, the low costs related to the ad hoc concept enable the reconsideration of the installation of communication networks in economically disadvantaged zones, where the deployment of fixed infrastructures is non-profitable [OLC].

13.3. Routing protocols

Because of the particular characteristics of ad hoc networks, the traditional routing protocols cannot be used in this context. Indeed, the protocols must take into account the strong mobility of the nodes and the absence of pre-configured routers. The approaches used are traditional: flooding, distance vector routing and link state. The IETF [IET] MANET [MAN] Working Group works in the standardization of ad hoc routing protocols. In general, we can distinguish two main categories of protocols, according to the way in which the nodes establish the routes: active and reactive protocols.

13.3.1. *Proactive protocols*

Proactive routing protocols permanently provide each node with information about the network topology. This information is obtained by the periodic flooding of control packets; from these each node builds its routing table. Therefore, a node that needs to establish a route towards a destination applies a path discovery algorithm to the information contained in this routing table.

The main advantage of proactive protocols is that the route establishment only cause a very slight delay. Indeed, at any moment each node already has the necessary information to establish a path towards any other node of the network. Moreover, the knowledge of the topology enables the nodes to calculate the optimal route in terms of hops. However, this knowledge also has a cost and it is the principal disadvantage of proactive protocols. Indeed, the control packets are exchanged periodically whatever the characteristics of the network (not very dynamic topology, high density of nodes, etc.). However, when the mobility of the nodes is low and there are very few changes in the topology, the emission of control packets is often useless. Moreover, the routing tables can contain routes towards nodes which will never be requested. Finally, the updates have a significant impact on the bandwidth, since they generate a considerable overhead.

Two proactive routing protocols have been standardized by the IETF: the *Topology Dissemination Based one acts Reverse-Path Forwarding* (TBRPF) [CLA 03] and the *Optimized Link State Routing* (OLSR) [OGI 04]. In the following part of this chapter, we will detail the OLSR protocol which since its standardization, has aroused much interest in the field of research as well as in the industrial field.

13.3.1.1. *The OLSR protocol*

The OLSR protocol is a link state protocol inspired by the traditional OSPF (*Open Shorts Path First*) wired routing protocol [MOY 89]. It also uses the sending of periodic control packets to inform each node of the changes which have occurred in topology. OLSR is distinguished from the traditional link state protocols by the introduction of a strategy optimization of the basic diffusion, carried out by particular nodes: the "multipoint relays" (MPRs). These MPRs are nodes entrusted with the exclusive responsibility of emitting certain routing information. Each node chooses its MPR among its one hop symmetric neighbors, in such a manner that using the MPR can reach any two hop neighbor. Each MPR itself maintains a list of all the nodes that have chosen it as a MPR; these nodes are called MPR selectors. Thereafter, the role of the MPR is to relay any message coming from their MPR selectors and to ignore messages coming from the other nodes. The discovery of neighbors and the link determination are performed by the exchange of HELLO messages. These messages are transmitted periodically (two seconds by default) by each node to all their two hop neighbors. For each node, they contain the list of all its known neighbors, as well as the type of link which connects them. This can be asymmetric (if an exchange was done in only one direction) or symmetric (when the exchange was carried out in two directions). It can also relay multipoint in the case where a node is specified as being selected as the MPR and finally lost when a link is detected as broken after a certain period of time. Thus, on receiving a HELLO message, a node examines the addresses list and the associated information to update its routing table.

In addition to the knowledge of the one hop vicinity, each node also maintains information on its two hop neighbors. The addresses of these neighbors are stored in a list and are used later in order to determine the optimal MPR that covers these nodes.

In order to update their routing tables, the nodes must be regularly informed about the changes of topology occurring in their vicinity. This is the role of the *Topology Control* (TC) message. These control packets are emitted periodically by each MPR with all the nodes of the network as a destination, in order to inform their whole set of MPR selectors. The consequence is that each node receives a partial graph of the topology conformed by all the network nodes and also the totality of the links between a node and its possible MPR selectors. Starting with this information, each node can very quickly determine the optimal path (in term of hops) towards any destination.

13.3.1.2. *The DSDV case*

There exist other proactive routing protocols, such as the DSDV (*Destination-Sequenced Outdistances Vector)* routing protocol [PER 94]. DSDV is a distance vector routing protocol based on the Bellman-Ford algorithm. As opposed to the majority of the distance vector protocols, DSDV solves the routing loop problem by associating a sequence number with each node, thus enabling it to differentiate the old routes from the new ones. Unfortunately, DSDV is also characterized by a consequent slowness, which is why it is less and less frequently used; in fact, we will not detail it here.

13.3.2. *Reactive protocols*

In large network, the proactive approach cannot be very powerful, because it is too greedy in bandwidth. That is why certain protocols are based on another approach, which is more specific to the field. It is then the case of reactive protocols which do not keep information about the topology of the network. On the contrary, they do not establish a route unless a node wishes to send a message. They are on-demand routing protocols. The advantage of these protocols is that the network is not flooded by the control packets until it is really necessary, i.e. only under a node request and not regularly, as in the case of proactive protocols. They are thus less expensive overall in terms of signaling and energy. On the other hand, the time to establish a route is somewhat longer than for proactive protocols, especially if the distance between the source and the destination is great. Moreover, all the nodes receive the requests, including those which are not concerned. The reactive approach thus also generates a certain traffic overload.

13.3.2.1. *The AODV protocol*

As in the case of DSDV, the AODV (*Ad hoc On demand Distance Vector*) protocol [PER 99] belongs to the category of distance vector routing protocols. AODV, while being based on a reactive approach, in fact constitutes an optimization of the DSDV protocol in the sense that it reduces the number of diffusions by creating routes only when needed. It also uses the sequence numbers to maintain the consistency of the routing information. Indeed, because of the mobility of the nodes, the routes frequently change and those maintained by certain nodes become invalid. The sequence numbers make it possible to then use the most updated routes.

When a node needs to establish a route towards a destination, AODV uses the concept of *Route_Request* (*RREQ*) packets. This can occur if the destination is not previously known or if the existing routes towards the destination were broken (i.e the associated metric is infinite). The node thus sends RREQ packets towards its neighbors who relay them in turn, so that all nodes of the network receive the request. When an intermediate node retransmits the request to a neighbor, it also saves the identifier of the node from which the first copy of the request was received. This information will be used later on to build the opposite route in order to answer the destination node. Thus, when this receives a RREQ packet pointing on its own address, it answers the source by emitting a *Route_Reply* (*RREP*) packet. This packet returns to the source by the opposite route, thanks to the information previously stored in the intermediate node cache. If it is not received beyond a certain fixed period, the source sends a new request. For each new diffusion, a RREQ packet field is incremented and after several unsuccessful requests, an error message is delivered with the application.

In order to maintain the route consistency, each node transmits HELLO messages periodically. If, in a given time interval, three messages are not received consecutively from a nearby node, then the link is considered to be faulty. In the case of link failures, all the routing table entries that are related to the failure are removed. This is achieved by the diffusion of an error message among the active nodes.

One of the disadvantages of AODV is that it manages symmetric links only. Indeed, since the route answer packet is sent to the source, the nodes belonging to the return path will modify their routing tables according to this packet. Also, it does not ensure the use of the best existing path between the source and the destination. However, recent performance evaluations showed that there are not huge differences (in term of optimization) between the routes established by AODV and those used by the protocols based on the shortest path research algorithms.

13.3.2.2. *The DSR protocol*

Just like AODV, the DSR (*Dynamic Source Routing*) protocol [JOH 96] uses on-demand routing but it is characterized by a routing approach from the source. Thus, it also requires a route discovery mechanism but one that is different from AODV; in fact the source node indicates in each packet header the list of all the nodes which comprise the route until the destination. When the source node does not know the route, it sends, as in the case of AODV, a RREQ packet, where it registers its address. The difference is then at the level of the intermediate nodes since these each add in turn their respective addresses. Then, when the destination receives the request, it is enough for it to reverse the list contained in the header of the packet to answer the source. The answer is also given with a RREP message. Thereafter, the source can directly register the route in the headers of the latest packets. It will be noted in addition that the intermediate nodes can store the routes that they examined (and possibly even several for each destination) in a memory zone (*route cache*) especially dedicated to this purpose, in order to avoid an expensive later discovery of neighbors or to answer a route request more quickly.

Because of the mobility of the nodes, the links can spontaneously be broken and some of the previous stored routes are then no longer valid. To solve this problem, DSR has a *route maintenance* mechanism. Thus, when a node detects a transmission problem at its link layer level, it sends a special message called *RRER* (for *Route Error*) to the transmitter of the packet. This message contains the node address which detected the error and also that of the node which normally follows it on the route. Once this message is received, the source node removes the address of the unreachable node from all the registered routes and truncates them at this point. Thereafter, a new route discovery procedure must be launched in order to determine a new route to reach the recipient.

Among the advantages of DSR over AODV, we can note the indifferent use of symmetric or asymmetric links. Indeed, a destination node can indicate in the header of its packets a route different from the one indicated by the source node. Another important advantage of DSR is the absence of routing loops. On the other hand, DSR induces some overheads in the signaling level since the route present in the packets header increases their size. We can estimate that this overhead is compensated by the absence of HELLO messages.

13.3.3. *Hybrid protocols*

In this context, certain protocols propose the combination of the two preceding approaches in order to eliminate their respective disadvantages, whilst keeping their advantages. This is the case of hybrid protocols. One of the most representative protocols of this category is the ZRP (*Zone Routing Protocol*) [HAA 02]. This divides the network into geographical areas. Thereafter, the proactive IARP (*Intra Zone Routing Protocol*) is used to communicate inside the zone while the reactive IERP (*Inter-Zone Routing Protocol*) is employed in order to allow the communication between zones.

Thanks to this hybrid approach, a protocol such as ZRP can converge much more quickly than a total reactive protocol in certain topologies.

13.3.4. *Performance*

Generally, to evaluate the performance, it is difficult to compare the proactive and reactive approaches. Indeed, all simulations carried out show that the performance vary considerably according to the network characteristics (node mobility, density, network diameter) and also with the selected mobility model. However, it seems to be the case that a protocol such as OLSR is more adapted to dense networks with a high mobility while a protocol such as DSR is more effective on not very dynamic networks with a low density. This is explained by the fact that the route do not need to be rediscovered regularly; the routes discovery mechanism is the most expensive phase of reactive protocols.

13.4. Attacks to routing protocols

Because of their particular characteristics, ad hoc networks offer vulnerabilities which do not exist in wireless architectures with access points. These vulnerabilities generate specific attacks against them where the traditional security measures are ineffective.

13.4.1. *Ad hoc network features*

In order to understand the reasons of the ad hoc model vulnerability, it is useful to examine what distinguishes them from the wired traditional networks. The RFC 2501 [COR 99] IETF MANET group defines the following characteristics as being inherent to ad hoc networks:

– *Dynamic topologies*: the network entities are free to move independently of each other. Thus, the network topology tends to change quickly and in an unforeseeable manner, forming one-way as well as bidirectional links. The vulnerability here lies in the lack of control on the components of the network. Indeed, in a wired network, the insertion of a link towards a new node can be easily detected and controlled. In the case of an ad hoc network, on the other hand, such an event can occur constantly. An attacker can then be more easily inserted into the network and move from victim to victim.

– *The absence of infrastructure*: in the absence of any fixed entity, it becomes difficult to set up a traditional public key infrastructure and establish a centralized certification authority. In addition, in the case of intrusion detection systems, this poses the problem of network supervision: the traffic is entirely distributed. Also, it poses the crucial problem of the synchronization of nodes; in the absence of satellite guidance system, it becomes very delicate to synchronize the nodes with the same clock. This functionality is vital to check the freshness of the messages in certain protocols.

– *Limited bandwidth* (capacity variable): currently, wireless connections offer much less capacity than wired connections. In addition, the output obtained in wireless communications – if we consider the effect of access to the media and the signal attenuation phenomena, noise or interference, etc. – is appreciably lower than the maximum theoretical output that a radio link allows. Another consequence of the link low capacity is congestion, which is very common in these networks (i.e. the needs of actual applications frequently exceed the capacities of the network). Also, the DoS attacks have a greater impact in ad hoc networks, because the available band-width can be easily saturated.

– *Reduced autonomy*: most nodes of these networks are light terminals and depend on batteries whose capacity is limited compared to calculations. An important need for these nodes is to save energy. Additionally, the node performance, regarding processor power or storage capacities, is more restricted than in the case of fixed networks. Here it is much more delicate to establish cryptographic protection mechanisms than in traditional networks, because of their cost in terms of calculations.

– *Need to cooperate*: in the absence of a router, each participant may have to relay packets to the other network nodes. Consequently, if one of these participants decides either with an "selfish" behavior (safe guarding the economy of its own resources) or with a voluntarily malicious goal, not to relay the packets, it is the network operation which is affected and its effectiveness is reduced.

– *Auto-configuration*: the vocation of ad hoc networks has been, by definition, to be the most autonomous possible. Auto-configuration seems to be an essential functionality, since it enables the integration of nodes into a network without requiring human intervention. On the other hand, such a mechanism constitutes a target choice for malicious nodes; in fact, a large number of attacks will be based on identity usurpations.

Because of these particular characteristics, ad hoc networks are much more vulnerable to attacks and naturally offer more faults than other types of networks when faced with a potential attacker.

13.4.2. *Description of attacks*

The term "attack" indicates an action aiming to compromise the confidentiality or the integrity of the information circulating in the network or, generally, to damage its good performance. In ad hoc networks, we generally distinguish two categories: passive attacks which are a direct consequence of wireless technology, where the attacker only listens to the traffic without influencing the routing process, and active attacks, where the nodes directly influence the routing process injecting packets in the network.

13.4.2.1. *Passive attacks*

Because of the nature of the access medium, it is very easy for an unspecified node to listen to communications without the knowledge of the participants. Indeed, in ad hoc networks, the nodes communicate sharing the air interface with "collision avoidance" access control [IE 97]. A common attack then occurs when an attacker is in listening mode (*promiscuous listening*) in order to collect everything over the air interface and thus to analyze the traffic (Figure 13.3). In this scenario, if we consider the signaling used by the protocol, a node can of course extract all kinds of strategic information like the data contents, but also the network connectivity, the localization of certain nodes, their IP addresses, MAC, etc. Regarding data protection, an IPsec coding protocol (obligatory in IPv6) would be enough to ensure the confidentiality of information. However, within the ad hoc network environment, its implementation does not constitute enough protection since the nodes cannot to start with have confidence in each other in order to exchange cryptographic keys easily. Moreover, it is not adapted to the ad hoc model because it was primarily conceived to ensure data confidentiality and not to protect signaling information. Another possible solution consists of carrying out a coding at the physical level, according to the time or the radio wavelength. However, in this case, a more dangerous and quite simple attack consists of scrambling these radio waves to make any information exchange impossible. Later in this chapter, we will study attacks concerning the routing information, i.e. on the routing protocol.

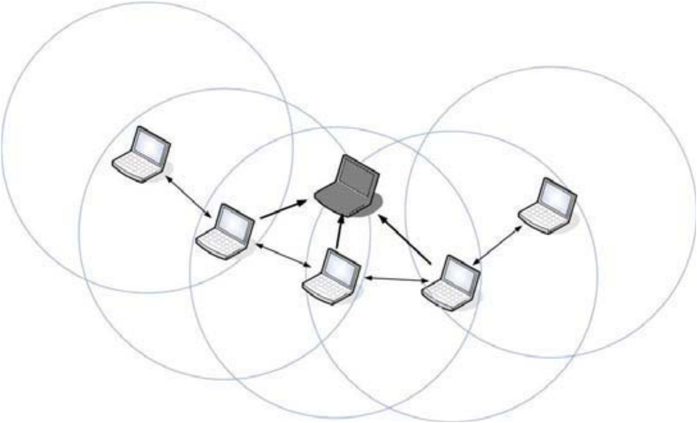


Figure 13.3. *Passive attack. The central node receives with discretion messages from other nodes without emitting any signal*

13.4.2.2. *Active attacks*

These attacks can target various layer of the OSI model (Table 13.1). Like other types of wireless networks, ad hoc networks are completely vulnerable to attacks at the level of the physical layer (e.g. jamming) or link layer (e.g. problem of the hidden station). However, because of the routing characteristics in these networks, it is the third layer which appears the most vulnerable. In the following part of this chapter, we will detail attacks related to the network layer (the other problems have already been approached in the preceding chapters).

In a completely distributed context like ad hoc networks, one of the major difficulties lies in node authentification and what information is exchanged. Indeed, without the presence of any central entity that filters the participants like in the case of networks with an infrastructure (router with firewall, access point), the nodes do not have *a priori* any means of checking the identity announced by a neighbor. It is thus very easy to usurp the identity of a legitimate node. This action can have several objectives. For instance, it allows the attacker to mask its true identity and pass for a legitimate node and then to launch more complex attacks, which affect the routing process inside the network.

Network layer	Security criteria	Attack	Target
Application	Confidentiality	Unauthorized listening	Node location, data content
Transport	Availability, integrity	Flooding, rejection	Messages
Network	Packet integrity, resources availability	Redirections, tunnels, suppression/traffic filtering, route destruction, battery exhaustion, etc.	Control packets, routing tables, autonomy
Link	Availability	Collisions, band saturation	Ethernet frames
Physical		Jamming, robbery and corruption	Terminal, radio waves, batteries

Table 13.1. *Attacks can affect all network layers*

Once the identity usurpation is done, the principal purpose of the majority of attacks will consist of deviating normal traffic from its normal route. In order to do this, the attacker asserts connections towards a maximum number of neighbors, by associating them, if necessary, with the smallest possible metrics. Thus, the neighboring nodes believe they selected the optimal route and the traffic is redirected towards the malicious node. The attacker can choose to redirect the traffic towards itself (Figure 13.4), towards a legitimate node or a distant ally node. By diverting the traffic towards itself, the attacker gives itself the possibility of analyzing or even filtering a maximum amount of information. If the entire data is then returned towards the destination, the attack is practically transparent. Moreover, if the malicious node wishes to disturb the operation of the network, it can filter the "pure" data, to let only the control packets pass. This attack (called *gray hole*) gives the illusion to other nodes that the network is functioning normally because they continue receiving the signaling data, whereas information in fact is lost, causing successive and expensive retransmition. Finally, the most brutal (*black hole*) attacks consist of purely and simply removing all the traffic and passing it by the attacker.

Thus, the connections are completely stopped, and all communications on large routes are paralyzed, which can cause real network partitions. This attack can be comparable to a non-participation attack.

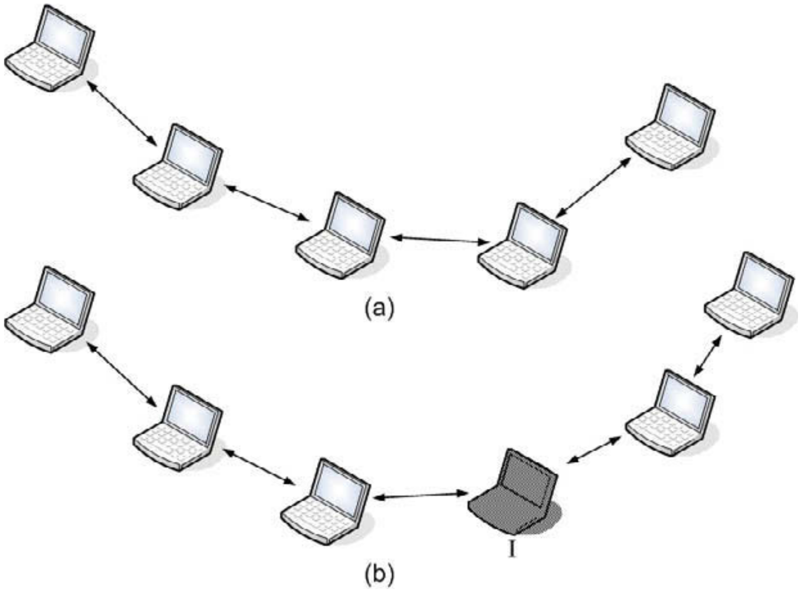


Figure 13.4. *Active attack scenario: gray node (I) is inserted in a route. (a) represents the initial route while (b) represents the route perception by the nodes after the attack*

The attacker can also choose to reorientate the traffic towards another legitimate node, in order to create routing loops in the network. Indeed, by successively usurping the identities of certain nodes, an attacker can force the nodes to redirect the traffic according to a cycle. The consequence is that the packets buckle between the same nodes without ever reaching their destination, thus consuming the available resources: energy and bandwidth. This attack requires a minimum level of information about the topology of the network, and the attacker will be satisfied to create sub-optimal routes to increase the packet routing time.

Another alternative and much more pernicious form of attack consists of creating a virtual tunnel (also called a *wormhole*) in the network by using an ally node. The traffic is all deviated by the attacker, then encapsulated in a new flow directed to the ally node, where it is decapsulated and sent to the destination node as if nothing has occurred. Intermediate nodes are not aware of the trick since the flow is encapsulated and transported as simple data; thus, it is not examined. This attack is extremely severe because is very difficult to counter. It affects the majority of ad hoc routing protocols because they are not conceived to detect this kind of anomaly. The current solutions to this attack will be described in section 13.5.

Not all attacks have the goal of perturbation of the routing process or, more generally, compromising the safety of the network. Indeed, ad hoc networks are characterized by limited resources. One of most critical in a context where the majority of the terminals are mobile is energy. Indeed, even if the batteries which currently equip the mobile terminals are increasingly powerful in terms of autonomy, the energy remains a crucial resource. Also, certain nodes could try not to participate in the routing process while refusing to relay the packets between two other distant nodes. The malicious node can then simply generate false routing informations by virtually increasing the length of all the routes passing by it. In the case of proactive protocols, it is enough for the nodes to establish only asymmetric links, in order not to be selected like multipoint relay. In the case of reactive protocols, it is even more trivial. It is enough for the malicious node to assert links with the artificially long metrics (AODV case) or even to add imaginary addresses in the route establishment packets headers (*RREP* packets in the case of DSR). In an ad hoc network context where the collaboration between the nodes is a paramount element for the good performance of the network, such selfish behavior constitutes a real problem. We will see in the following section some examples of mechanisms created to promote or force the collaboration between the nodes, as well as their disadvantages.

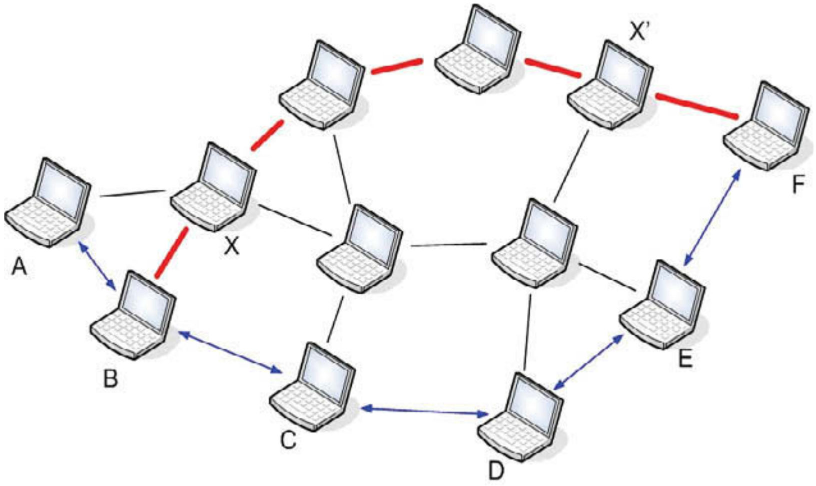


Figure 13.5. *Description of a tunneling attack*

As in the case of traditional wired networks, attackers exploit the vulnerabilities of the ad hoc protocols to conduct specific attacks. Thus, a frequent attack (called a *rushing attack*) targets the duplication suppression mechanism of on-demand reactive protocols. Indeed, in order to minimize the duplications generated by the request floods through the network, certain protocols like AODV or DSR remove received RREQ packets several consecutive times. The attack consists of a malicious node disseminating through the network and in a very short amount of time the maximum possible requests, thus forcing the honest nodes to remove all recent legitimate requests.

Generally, most protocols are vulnerable to these brutal DoS attacks. As we have seen, the previous attack uses the route discovery mechanism of reactive protocols. It is also easy to exploit the route diffusion maintenance message mechanism of the RRER network to announce route ruptures. The nodes receiving this message update their routing tables and remove existing routes, possibly causing network partitions. Proactive protocols are not protected either. Indeed, a node can regularly send HELLO messages in which it will sometimes assert bidirectional links and sometimes links ruptures. Actually, we can consider that all the attacks described previously constitute a DoS attack, because they deprive the nodes of resources like the bandwidth, access to the network, address space, or even their energy. Each protocol mechanism can constitute a potential fault and the security mechanisms must not only prevent the existing attacks, but must also take care not to offer additional vulnerabilities.

13.5. Security mechanisms

For many years, the security problems were completely ignored in the field of ad hoc networks, the majority of research focusing on improving the performance (output of the protocols, *overhead* limitation, etc.). Thereafter, several mechanisms were planned to increase the robustness of routing protocols without affecting the performance too much. Some of them simply consist of basic optimizations to the protocols, in order to prolong their use in a hostile environment. Others, on the other hand, are inspired by more advanced but also more expensive techniques such as cryptography to guarantee essential functionalities like confidentiality and authentification.

13.5.1. *Basic protections*

If the specific characteristics of ad hoc networks often constitute an obstacle to routing security, they can also be exploited assuming the opposite: to reinforce the data routing. This is the case, for example, for route redundancy. Each node in an ad hoc network could possibly act as a router. Consequently, even for a small number of nodes, it is often possible to find several different routes between two nodes. However, the majority of traditional protocols (AODV, OLSR, ZRP, etc.) have the ability to establish several routes between two nodes exchanging information. A simple solution then consists of benefiting from this multiplicity of routes to make a secure transfer [TSI 01]. If a malicious node is identified, the protocol can almost always find a route which makes it possible to evade it. Also, it becomes possible to transmit redundant information through additional routes in order to allow the recipient to check the integrity of the information sent. We can thus associate detecting error codes, error correction, or hashing of the transmitted data. For example, if there are N routes disjoined between two nodes, we can employ (N – R) channels to transmit the data and to use the R remaining channels to transmit the redundant information.

Offering a good level of security, this technique has the disadvantage of appreciably reducing the available bandwidth by increasing the control traffic. Additionally, it does not solve a certain number of problems mentioned above like identity usurpation, false signaling packet injection or route redirection.

Another approach consists of using the nature of the medium, i.e. the radio wave, in order to ensure that the information is really transmitted. Indeed, another characteristic of ad hoc networks is a completely open medium with a shared access, where all the nodes can listen to the information transmitted by their one hop neighbors. Thus, a solution [LEE 02] developed by a team of Maryland University consists of modifying the basic protocol (DSR) in a way that each answer to *Route_Request* is the object of a confirmation by a neighbor of the transmitter. Since a node receives a *Route_Request* packet which corresponds to a valid route in its route cache, it of course answers with a *Route_Reply* packet, but also sends a packet of request for confirmation (*CREQ*) near the first neighbor downstream. This examines its route cache to look for a route towards the destination. If it finds one, it answers the source with a packet (*CREP*) containing this information; in the contrary case, it does not answer. On its side, the node source compares the information sent by the first intermediate node with the confirmation received by the neighbor. If they are different or, more simply, if the neighbor downstream does not send anything, the node source does not take into account the answer of the intermediate node and looks for another route.

It should be noted that this process aims to make the route discovery secure by ensuring that the announced route really exists. However, it is too simple and restrictive so its security cannot be considered as sufficient. First, it takes into account only the *Route_Reply* security of intermediate nodes, which merely represents a technical improvement of the source caching of the DSR protocol. Then, it obligatorily requires the employment of additional tools that are able to provide packet authentification because, if not, nothing would prevent the intermediate node from falsifying a confirmation and answering the source node usurping the addresses of its neighbor. Now, let us suppose that another intermediate node decides to remove the *Route_Reply* packet confirmation that it relays; the route will never be established and the suspicion will immediately weigh on the node which was the source of the *Route_Reply*. The route confirmation must necessarily be transmitted through another route. Finally, if two malicious nodes are associated to conduct an attack, the protection process can be circumvented very well. The first node sends *Route_Reply* which is confirmed by the second node with the help of a *Route Reply Confirmation* and a false route will be established.

These mechanisms offer a security level that is higher than traditional protocols while reinforcing the packet routing process. On the contrary, it is not sufficient to meet the basic security requirements such as confidentiality and authentification. This is why the most effective protocols use more conventional mechanisms inherited from wired networks such as access control, cryptography with public keys, digital signatures, etc. These mechanisms are known to be preventive because they aim at preventing in advance, the attacks of compromised nodes in the network. In parallel, certain approaches aim at detecting in realtime the attacks within the network to support the cooperation between the nodes in order to restrict the impact of the malicious nodes. These approaches are known as reactive and can be used to complement the preventive approaches.

13.5.2. *Existing tools*

As we have seen in section 13.4.2.2, most of the attacks are characterized by control packet corruption and identity usurpations. Typically, a malicious node will alter the content of these packets in order to create routing loops or to remove routes illegitimately. Thus, a protocol to secure the routing must prevent these attacks in order to guarantee the packet integrity and authenticity. This is why the most successful routing protocols are generally based on traditional tools like the hashing functions and the symmetric/asymmetric coding mechanisms. Thus, to guarantee the message authenticity, the most effective solution consists of providing each node with secret keys, which are used to cipher and to decipher the received messages. In the case where there exists only one and single key for each possible pair of nodes, such a process guarantees the source of each message. Because of the strongly distributed character of ad hoc networks, certain protocols preferred to be directed towards the asymmetric coding mechanisms. In this case, we no longer use only one secret key to cipher and decipher a message but a duple public/key private key. The asymmetric coding mechanisms also make it possible to ensure the authenticity of the messages or their confidentiality. In the first case, the transmitter of a message ciphers it with its secret private key. The recipient can then decipher it with the public key of the transmitter, which is known by all nodes. Since each private key is associated with one and only one public key and considering that the protocol used is sufficiently reliable, this operation guarantees the authenticity of the message. If, on the other hand, the goal is confidentiality, the transmitter will prefer to cipher the message with the public key of the recipient. Thereafter, only this last one will be able to decipher the message, since it is the only one to have the corresponding private key.

Such mechanisms indirectly make it possible to also guarantee the message integrity. Indeed, if they are corrupted in the routing process, they can no longer be deciphered. Consequently, a deciphered message is a message which was not faded. However, considering only the integrity of the packets, these mechanisms are relatively expensive as they make it necessary to cipher then to decipher the whole message. A more adapted and economic solution consists of using the hashing functions in one sense.

The hashing functions are mathematical objects that, with given data provided as an entry: $\{0,1\}$ *, associate a smaller set of data – a few hundred bits of magnitude – characteristic of the start entry: $\{0,1\}$ ρ, ρ being the length in bits of the image function. We call this image a print or a digest. In practice, when the transmitter of a message wishes the destination to check the integrity, it applies a hashing function to the message and joins it with the calculated print. Thereafter, this recipient recomputes the print and the result obtained is compared with the received print. If they are different, it means that the message was altered. For the integrity to be formally checked, the hashing function must satisfy the following properties:

– it is very difficult to find the content of the message starting from the print (attacks on the first pre-image);

– from a given message and its print, it is very difficult to generate another message which gives the same signature (attacks on the second pre-image);

– it is very difficult (that is, to exceed the current capacities of calculation) to find two random messages which give the same signature (resistance to the collisions).

If we wish to also guarantee the integrity and the authenticity of a message, the cryptographic hashing function can be combined with a coding mechanism. In this case, we then speak about authentification message codes (HMAC, *keyed-hash message authentication codes* [MS 01]): the message is given to the hashing function which generates the corresponding print. Then, this is ciphered in order to prove the message authenticity to the transmitter.

Another interesting characteristic of hashing functions is their capacity to be used in a recursive way, to produce hashing chains. The construction of a hashing chain thus consists of successively applying a hashing function to the previously calculated exit. In practice, a node chooses a starting element $x \in (\{0,1\}$ and calculates a list of values h_0, $_{h1}$, …, h_n where $h_0 = x$ and $h_1 = H(h_{i-1})$ for all $i < n$. Thereafter, starting from an authenticated element of the hashing chain, we can easily check a former element by applying as many times as necessary the hashing function and by comparing the values of the elements. For example, starting from an authenticated value h_i, a node can authenticate the value h_{i-3} by calculating $H(H(H(hi–3)))$ and by checking that the value obtained is identical to h_i. A great advantage of the hashing chains is that they do not require large storage capacities and calculation. For example, Jakobsson and Coppersmith developed a mechanism of storage of hashing chains [COP 02, JAK 02] such as a chain made up of n elements requiring only O (log (N)) operations of storage and $O(log(n))$ arithmetic operations to reach one of the elements. These characteristics make it a tool that is completely adapted to use in an ad hoc network where the resources are, by definition, limited.

These processes are very effective but some of them are not completely appropriate to the ad hoc network environment. Thus, secret key distribution within a network where not all the participants are known constitutes a delicate problem. Indeed, in a context where the nodes are mobile and where it is possible to spy on the information forwarded in the vicinity, it is difficult to set up a secure channel to exchange the keys. These keys must thus be installed before the deployment of the network, during the initialization phase. Some authors recommend a manual key distribution by equipping each node with a smart card on which the key will be stored in hardware. If this solution is possible for small size networks, perfectly controlled as military networks or the terminals of a telephone operator, it does not work in the same way for open networks like citizen networks or networks with a high density such as sensor networks. Since each pair of nodes likely to communicate with each other must have a key, the total number of keys in a network of n nodes is n x $(n-1)/2$. This can represent a considerable number of keys to be managed in certain circumstances.

Asymmetric cryptography then seems to be a more suitable solution because of its flexibility. However, it also suffers from a defect penalizing its use in the ad hoc context. Thus, when we wish to send a coded message we must attach to it a certificate issued by a certification authority. The role of this certificate is to prove that it possesses the public key for a certain period. The disadvantage of this approach is that in the context of ad hoc networks which, by nature, are deprived of any infrastructure, it is not possible to go to a centralized and fixed certification authority, as can be done for traditional networks. In the following part of this chapter, we will see that in order to be able to be used effectively, these mechanisms will have to be adapted in order to satisfy the constraints of the ad hoc model.

13.5.3. *Key management architectures*

As we have seen previously, the traditional mathematical tools can be used to conceive protected routing protocols. Now let us detail the existing solutions and their limits.

13.5.3.1. *The Resurrecting Duckling*

In order to facilitate the key distribution in an ad hoc network, Franck Stajano and Ross Anderson proposed in [STA 99] a mechanism to exchange a secret key between two nodes. This model, called the Resurrecting Duckling, is based on a master/slave relationship and on the concept of impregnation. Thus, during an initialization phase (before its introduction within the network), a slave node must be "impregnated" by its main node (possibly the owner) by physical contact (e.g. electric). At the time of this contact, a secret key is exchanged confidentially. Thereafter, this key can be used to code and authenticate information, like a list of other shared keys. Although innovative, this approach leaves several questions unanswered. The first relates to the impregnation phase. If a physical contact is possible within the framework of a small network (a piconet [BEN 97] for example) with an appointed leader, it becomes less possible within the framework of an open wide-area network. The second problem is related to the key management. Indeed, the approach does not propose how to make the secret key exchange between each pair of nodes of the network. In addition, if one of the nodes is corrupted, all the other keys relating to this node can be threatened and nothing is mentioned about key repudiation. A systematic rebootstrapping appears difficult to set up.

13.5.3.2. *SUCV*

In [MON 02], Montenegro and Castellucia developed another approach called SUCV (*Statistically Unique Cryptographically Verifiable identifiers and addresses*) in which each node builds an address based on its public key. Each node generates a public/private key pair and then chooses its address, calculating it from the public key, using a cryptographic hashing function. The authors propose two mechanisms. In the first one, the IPv6 addresses of a node correspond to the complete result of the hashing function on the public key. In the other approach, only the 64 least significant bits correspond to the result of the hashing function. Thus, if an attacker wishes to compromise a given SUCV, it will have to carry out 2^{63} (roughly $4.8*10^{18}$) tests to find a key public whose print is identical to that of this SUCV. If this attacker has the option of calculating a billion prints a second, it will take him roughly 142 years to find this collision. The disadvantage of this approach is the fact that does not solve the problem of key installation. Thus, in a normal network, the problem consists of obtaining a list of couples (nodes, public keys) of confidence; here, we must despite everything determine a list of trusted nodes.

An alternative approach consists of defining one or more certification authorities. Indeed, the presence of a public key is not enough; it is still necessary that one node can check the legitimacy of the public key used by each node; this is the role of the authority. Each node of the network has a certificate which contains its IP address, its public key and, of course, a signature of the certification authority. When a node wishes to send a message, it signs it and joins its certificate there. Thereafter, the receiving node checks the certificate initially then uses the public key contained in this certificate to check the signature of the message. However, several problems arise. The first relates to the availability of the authority. Indeed, in a network free from any fixed infrastructure, the question of access to the authority is posed to check the certificate. Certain links break, the nodes can move and thus it is not certain that each node has access to the authority at any moment, and then to the certification service. The second problem relates to the mutual dependence between security and routing. Indeed, to validate a certificate from a certification authority, it is necessary, as a preliminary step, to establish a route, but for this route to be established in a secure way, access is needed to check the public keys of each intermediate node.

13.5.3.3. *Architecture of distributed certificates*

To solve the constraints induced by the absence of a centralized infrastructure, Zhou and Haas imagined taking benefits from the intrinsic characteristics of ad hoc networks in order to conceive a new certificated management approach. They imagined a system of key certification [ZHO 99] where the authority is not only given to one fixed entity but on the contrary is distributed among several nodes of the network. In this manner, the certification service obtained is defined by a distributed certification authority that has a pair of public/private keys. The public key is known by each node of the network and enables them to check with confidence any certificate signed with the private key. The private key is not known by any particular node, but is in fact is partially distributed on several nodes called contributors. Thus, a client node that wishes to obtain the public keys of the other clients or to launch updates to change its own public key has to emit a request to the certification service. To guarantee a good safety level even in a distributed context, the certification service rests on threshold cryptography. A diagram of threshold cryptography (*n, t+1*) is conceived in such a way that among *n* nodes which share the management of the keys, *t+1* will have the possibility of proceeding to the operations of coding, while *t* nodes alone will be unable, even in coalition. Thus, when the service must sign a certificate, each server node generates a partial signature by using its private key, and transmits the result to another server called an assembler which will be in charge of assembling the portions of the signature of *t* nodes. When this server receives a *t+1* correct partial signature, it is able to calculate the final signature of the certificate. It should be noted that this assembler role can be filled by any of the *n* nodes. To reinforce the robustness of the device and to thwart the possible compromising of this server, the authors recommend assigning this role simultaneously to *t+1* possible nodes (of course, the phase of signature checking is then weighed down considerably). The advantage of this model lies in the fact that *t* malicious nodes cannot create valid certificates since *t+1* valid partial signatures are necessary. Of course, we are not safe from an attacker which generates false signatures systematically, in order to lead to the creation of an invalid certificate. However, the assembler node always has the possibility of checking the validity of any signature using the public key of the service. If the checking fails, the assembler must indicate another group of *t+1* partial signatures. This procedure continues until it manages to generate a correct signature.

The negative point of the architecture suggested by Zhou and Haas is its complexity of implementation. Indeed, the security is based on the choice of the assembler nodes. If the number of malicious or corrupted nodes exceeds a certain threshold, the service becomes inoperable. Moreover, it is probable that the need to have certificates of several nodes for each coded message generates a consequent overhead on the level of the load network, where of all the assembler nodes must be sent and received.

13.5.3.4. *PGP approach*

Another solution [HUB 01] proposes using the traditional online certification model considering the concept of certificate graphs as a starting point (the tops of the graph represent the public keys of the users while the terminals represent the certificates) of the PGP (*Pretty Good Privacy*) protocol. In this model, each node signs the certificates of the participants in which it has confidence, according to its own criteria. The certificates rest on a transitive confidence, i.e. if *A* trusts *B* and *B* trusts *C*, then *A* trusts *C*. However, differing from PGP, the certificates are stored and then distributed by the nodes themselves and not by an online server. Thus, each node has a "*local certificate register*". Thereafter, when two nodes mutually wish to check their identities, they merge their respective register with the idea of finding a certificate chain which binds them in a trust relationship.

The success of this approach depends mainly on the characteristics of the certificate graphs but also on the construction of the local certificate register. In addition, before being able to generate certificates, each node must initially build its own certificate register, which constitutes a complex operation. Moreover, if the number of revoked certificates becomes too considerable, the certificate register become obsolete and the certificate chains are no longer valid.

13.5.3.5. *TESLA*

The TESLA (*Timed Efficient Stream Loss-tolerant Authentication*) protocol [PER 00] was conceived to allow an authentification of a multicast flow source, which tolerates losses. The basic TESLA principle is the following: the transmitter of a message has associated a code of message authentification (*MAC*) obtained using a secret key that will be revealed only after a certain amount of time. The mechanism begins by creating *MAC* keys. In order to do this, the transmitter generates a series of keys K_1, K_2,..., K_t using a hashing function in one sense. It first generates a random key and calculates the following keys by applying the hashing function successively: $K_i = h(K_{i+1})$. Then, it generates the *MAC* keys using another hashing function in one sense: $K'_i = h'(K_{i+1})$ (Figure 13.6). The use of two distinct hashing functions is a precaution taken by the authors to further reinforce the security (the use of the same function for all cryptographic calculations can constitute an exploitable potential vulnerability by an attacker).

$$\begin{tikzcd}
K_{i-1} \arrow[d, "h'(K_{i-1})"] & K_i \arrow[l, "h(K_i)"'] \arrow[d, "h'(K_i)"] & K_{i+1} \arrow[l, "h(K_{i+1})"'] \arrow[d, "h'(K_{i+1})"] & K_{i+2} \arrow[l, "h(K_{i+2})"'] \arrow[d] \\
K'_{i--1} & K'_i & K'_{i+1} & K'_{i+2}
\end{tikzcd}$$

Figure 13.6. *Use of hashing functions in TESLA. The hashing values appear at the top and the corresponding MAC keys appear below*

Thereafter, the transmitter attaches a calculated MAC to each packet starting from its contents and generated thanks to the one sense hashing function. It divides the time into several intervals of fixed duration. During the same interval, the transmitter can send zero or several packets. Once the sending is done and with the expiration of a preset delay, it can reveal the corresponding key which will be used to authenticate the packet (for example, the key used during interval *i* is revealed during the interval *i+3*).

On its side, when the receiver receives a packet including an index of interval *i*, it must estimate the interval in which the transmitter is using its local clock (taking account of the time of transmission of a packet and estimating the clock of the transmitter). This estimate is used to check that the transmitter did not arrive yet in the time interval where it reveals the K_i key. If this condition is not observed, the integrity is no longer formally guaranteed and the packet is rejected. In the contrary case, the receiver cannot for the moment check the authenticity of the message sent during interval i without the corresponding key K_i which will be delivered later. It thus records a triplet (containing the message, the index of the interval as well as *MAC*) in a buffer until it receives the K_i key. Once it is received, the receiver ensures its legitimacy by hashing it successively several times and comparing the obtained print with the value of a previous key. For a certain number of successive passes, the values must be identical. Thus, by noting d the time of disclosure of the keys, and Kv $(v < i\text{-}d)$ a former key, we must obtain $Kv = h^{i-d-v}K_{i-d}$.

We can note here one of the main advantages of TESLA related to the properties of the hashing chains: starting from a revealed key, we can calculate all the preceding keys, so that even if several packets of the same interval are lost, a node is always able to check them starting from a key obtained in a later interval. Thus, if the value of (i-v) is higher than 1, the receiver can check the authenticity of all the packets recorded during the intervals ranging between $v+1$ and i-1. This characterizes the capacity of tolerance to TESLA losses. Another important property is the unidirectional flow sense, i.e. a source towards one or more destinations. The source reveals the corresponding key at intervals of the packets sent independently of the number of receivers. It is this scale capacity which enables TESLA to be used within the *multicast* flow environment.

On the contrary, certain aspects of TESLA require very detailed attention. Thus, the choice of this delay is crucial. It must be sufficiently large so that the recipient received the message before the key and at the same time sufficiently small to ensure a good reactivity of the network. Indeed, if the delay is too short, the key is likely to be received before the message and the integrity of this could not be guaranteed anymore. Conversely, too great a time will significantly increase the time necessary for a node to authenticate a message, which is likely to generate a consequent delay inside the network. It should be noted on this subject that several requirements must be satisfied to make this protocol work. The first is the capacity of the nodes to be synchronized (at least roughly). The second is the need for TESLA to be started by a system making it possible to install the keys.

13.5.4. *Protections using asymmetric cryptography*

The protocols detailed in this section generally assume the existence of a management system and distribution of pre-established keys. They then use the security mechanisms described in section 13.5.2 to ensure the integrity and the authenticity of control packets. To prevent the risk of usurpation in a network, the protocol must be able to guarantee node authentification. In order to do this, the coding mechanisms seem to be most effective. The difference between the various protocols is the selection of the cryptographic process.

13.5.4.1. *SAODV*

Zapata and Asokan developed a protocol dedicated to the security of the AODV protocol, called SAODV (*Secure Ad hoc On-demand Distance Vector)* [GUE 02]. The principal idea of SAODV consists of using signatures to authenticate the majority of the fields of the *Route_Request* and *Route_Reply* packets and use hashing chains to protect the integrity of the hop counter. Thus, SAODV constitutes an extension of AODV with signatures, in order to repel "identity usurpation" attacks. SAODV requires the presence of a certification authority in order to check the signed packets, thus ensuring their authenticity. In SAODV, each *RREQ* packet includes a simple extension of the signature. The initiator *S* of the packet chooses a maximum number of hops while basing itself on an estimate of the diameter of the network and it then generates a one sense hashing chain whose length equal to the number of hops, plus one. The example of the SEAD protocol is detailed in section 13.5.6; the hashing chains in SAODV are used to authenticate the metrics in the signaling packet headers. This process is described in Table 13.2: the initiator S of *RREQ-SSE* packet includes the message (*RREQ*), an identifier (*i*), the address of the source node (destination) as well as a sequence number Seq_S (Seq_D). Moreover, this header also includes an element of the hashing chain (h_0) based on the estimate of the hop number (*N*) of header *RREQ*. This value is called the hop number authenticator. If, for example, the values of the hashing chain $h_0, h_1..., h_N$ were generated so that $h_i = \mathrm{H}[h_{i+1}]$, then the authenticator hop number h_i corresponds to a number of hops value *N-i*. Thereafter, the source node signs everything using its private key K^-_S, before adding the hop meter to the corresponding print. Before relaying a *RREQ-SSE* request, a node starts checking the authenticity of the message in order to make sure that each field is valid. It then removes possible duplications (packet coming from several nodes). It then increments the hop meter, ciphers it, adds the print and re-emits it. When the request arrives at the destination, its authenticity is checked. If the request is invalid, it is simply deleted. Otherwise, the process is similar to AODV: the destination answers with a *RREP-SSE* packet very similar to a *RREQ-SSE* request. The difference is in the presence of the field lifetime which corresponds to the exact number of nodes used to return the answer. The packet is then signed and complemented with a hop meter in an identical way.

S ?*: (($RREQ$, id, S, seq$_S$, D, oldseq$_D$, h$_0$, N)$_{K^{-}_{S}}$, 0, h_N)
A ?*: (($RREQ$, id, S, seq$_S$, D, oldseq$_D$, h$_0$, N)$_{K^{-}_{S}}$, 1, h_{N-1})
B ?*: (($RREQ$, id, S, seq$_S$, D, oldseq$_D$, h$_0$, N)$_{K^{-}_{S}}$, 2, h_{N-2})
C ?*: (($RREQ$, id, S, seq$_S$, D, oldseq$_D$, h$_0$, N)$_{K^{-}_{S}}$, 3, h_{N-3})

D ?C: (($RREP$, D, seq$D$, S, lifetime, h'_0, N)$_{K^{-}_{d}}$, 0, h'_N)
C ?B: (($RREP$, D, seq$D$, S, lifetime, h'_0, N)$_{K^{-}_{d}}$, 1, h'_{N-1})
B ?A: (($RREP$, D, seq$D$, S, lifetime, h'_0, N)$_{K^{-}_{d}}$, 2,h'_{N-2})
A ?S: (($RREP$, D, seq$D$, S, lifetime, h'_0, N)$_{K^{-}_{d}}$, 3, h'_{N-3})

Table 13.2. *Route discovery in SAODV. Here, the node S establishes a route towards the node D*

With the exception of the hop number and its authenticator, the fields contained in the *RREQ* and *RREQ-SSE* packet headers are not modifiable and can thus be easily authenticated by checking the signature in the *RREQ-SSE* extension. When it relays a *RREQ* request, a SAODV node can authenticate the *RREQ* packet to ensure that each field is valid. Then, it removes the duplicated packets in order to not retransmit more than a *RREQ* for each route exploration. The node then increments the hop number in the *RREQ* header, calculates the print which will authenticate the hops and resends the request for the *RREQ-SSE*. When the request arrives at the destination, the authenticator in the extension is verified. If the request is valid, the destination turns over a *RREP* as in AODV. As for the *RREQ*, the only modifiable field of the *RREP* is the hop number. Consequently, security is performed in the same manner.

SAODV also uses the signatures to protect the *RRER* messages during the mechanism to maintain the route (route maintenance). Thus, each node using SAODV signs the *RRER* messages that it emits. On the other hand, the nodes do not change the information concerning the number of sequences when they receive a *RRER* packet because the destination node does not authenticate the sequence number.

This protocol ensures a good authentification of control messages as well as a good integrity. However, the use of hashing chains is not effective against all attacks on the hop number. Thus, although the hashing of the hop number prevents a possible malicious node from announcing routes shorter than those existing, nothing prevents an attacker from arbitrarily increasing the length of the routes. Indeed, such a node can apply the hashing function several consecutive times before relaying a packet; the route then appears longer than it actually is.

In addition, in the case where there would be several attacker allies, a tunnel attack can always be launched and the hop number can even be decreased at the arrival, in a transparent way for the other nodes.

13.5.4.2. *ARAN*

The creators of the ARAN (*A Secure Routing Protocol for Ad Hoc Networks*) [DAH 02] also chose to use cryptography with public keys to secure the routes. ARAN is an on-demand protocol, which provides an authentification service hop by hop using a public key infrastructure. It thus supposes the existence of an authentification server T, whose role is to manage the certificates and whose public key is known by all participants. Before entering the network, each node must identify itself to the server and must request a certificate which will be used to sign the messages that it will send. This certificate contains the IP address of the node, its public key, a first stamp which gives an account of the creation date of the certificate, and a second stamp which indicates its expiry date. In a traditional way, this certificate is then signed by T and must be regularly updated.

$S \text{->}*: (\text{RDP}, D, \text{cert}_S, N, t)K^{-}_{S}$

$A\ ?\ *: ((\text{RDP}, D, \text{cert}_S, N, t)K^{-}_{S})K^{-}_{A}, \text{cert}_A$

$B\ ?\ *: ((\text{RDP}, D, \text{cert}_S, N, t)K^{-}_{S})K^{-}_{B}, \text{cert}_B$

$C\ ?\ *: ((\text{RDP}, D, \text{cert}_S, N, t)K^{-}_{S})K^{-}_{C}, \text{cert}_C$

$D\ ?\ C: (\text{REP}, D, \text{cert}_D, N, t)K^{-}_{D}$

$C\ ?B: ((\text{REP}, D, \text{cert}_D, N, t)K^{-}_{D})\ K^{-}_{C}, \text{cert}_C$

$B\ ?A: ((\text{REP}, D, \text{cert}_D, N, t)K^{-}_{D})\ K^{-}_{B}, \text{cert}_B$

$A\ ?S: ((\text{REP}, D, \text{cert}_D, N, t)K^{-}_{D})\ K^{-}_{A}, \text{cert}_A$

Table 13.3. *Route discovery mechanism in ARAN*

The goal of ARAN is to secure the route discovery mechanism from node to node. Thus, when a node wishes to send a message, it generates, signs, then sends a RDP (*Route Discover Packet*). Thereafter, each intermediate node receiving this packet checks the certificate of the preceding node, adds its own certificate and resends the packet. Once this packet arrives, the destination node checks the certificate and answers in unicast, using a REP (*Reply Packet*) message which is checked node by node. This mechanism is illustrated in Table 13.3. In this example, the source node (*S*) initiates the route discovery mechanism sending a RDP previously signed packet, which includes the node's destination address (*here, D*), its certificate (*cert$_S$*) a "nonce" (*N*) as well as a stamp (*t*) (note that the nonce and the stamp guarantee the freshness of the message, and also imply the existence of a mechanism that makes it possible to synchronize the nodes on a common clock). Thereafter, each node that has to relay this request starts by checking the signature and the freshness of the certificate, then validates this packet by adding its own signature and its own certificate. Once finished, it relays the request to the following node and this is done until the destination is reached. Thus, in our example, the node checks the data $cert_S$, signs the packet then adds its own certificate $cert_A$. Thereafter, the following node (*B*) checks the certificate $cert_A$ packet transmitted and uses this to validate the signature. It proceeds in the same way with the message encapsulated by checking the certificate $cert_S$ source, then its signature. When all the signatures are validated, node *B* removes the signature of the preceding node and adds its own. *C* proceeds in the same way until the destination is reached.

When the request finally arrives at the destination, node D generates and signs a REP packet, adds its certificate and sends the whole to the request source node (C). Thereafter, this packet is relayed until the source in the same way as the request, i.e. each node checks the signatures contained again. Thus, node C checks the certificate $cert_D$ of node D, validates its signature and in turn signs the packet. B proceeds in the same way with this new packet and the signature of C but also checks the certificate $cert_D$ D to validate the original signature of the answer. The difference compared to the RDP is that the REP is transmitted in unicast, by reversing the routed determined to receive it. In the same way, following the example of reactive protocols, each node receiving the REP establishes a new entry in its routing table which indicates the address of the next node for the packets for D.

$$B \text{ ? *: } ((\text{ERR}, D, \text{cert}_B, N, t\,)K^-_B)$$
$$A \text{ ? *: } ((\text{ERR}, D, \text{cert}_B, N, t\,)K^-_B)$$

Table 13.4. *Routing maintenance in ARAN: each node relays the packet without re-signing*

The ARAN protocol also specifies how to protect the route maintenance mechanism. When an intermediate node detects that a route is broken, it sends a *Route Error* (ERR) packet to the upstream next node (in the direction of the source). This packet includes the addresses of the source and destination nodes, the certificate of the intermediate node as well as a nonce and a timestamp (Table 13.4). The packet is then relayed without being resigned by the intermediate nodes.

Because the packets do not contain any hop meter and especially because the authentification is carried out node by node, possible malicious nodes cannot create routing loops, or redirect the traffic while inserting non-legitimate addresses in the route discovery packets. In this sense, ARAN shows great robustness against this type of attack. Also, the public use of the ciphering mechanism with public keys opens the way to DoS attacks.

Indeed, in this protocol, for each discovery route packet, it is necessary to check the certificate provided, to decipher the packet, then re-cipher it with its own key and add its certificate. When the number of packets becomes considerable, this can be extremely expensive. Also, a DoS attack will consist of flooding the network with false control packets, where the verification will monopolize the resources of the nodes. In addition, if a node cannot perform the verification in realtime, it can be threatened by an attacker that could randomly remove certain packets including valid packets.

When comparing ARAN and SAODV, it should be noted that in spite of an authentification from node to node and from end to end, ARAN does not bring a significant benefit in security terms over SAODV (which only provides an end to end authentification).

13.5.5. *Protections using symmetric cryptography*

13.5.5.1. *SRP*

Papadimitratos and Haas proposed a protected routing protocol, SRP (*Secure Routing Protocol*) [PAP 02], which is especially adapted to the characteristics of the DSR protocol and the interzone routing protocol (ZRP). Thus, they conceived SRP as an extension to the header of the *Route_Request* and *Route_Reply* packets. SRP uses sequence numbers in the interior of the requests to guarantee their freshness; however, this sequence number can only be checked at the destination. Moreover, it establishes security associations between the communicating nodes only. This association is then used to authenticate the *Route_Request* and *Route_Reply* packets through the MAC. At the destination, SRP allows the detection of modifications of *Route_Request* packets while at the level of the source; it is the *Route_Reply* integrity that will be analyzed.

Since the SRP only requires security associations between the communicating nodes, it is relatively light. On the other hand, certain defects are quite punishing, thus limiting its interest. SRP does not secure the route maintenance mechanism and delegates this task to another protocol. Moreover, SRP does not detect the modifications relating to the routing information subjected to modification at the routing time. For example, a corrupt node can easily remove the contents of the node list included inside a *Route_Request* packet. Finally, the integrity of messages is only checked at the source and destination node level, and an attacker can always corrupt packets to waste network resources in useless retransmissions.

13.5.5.2. *SAR*

The SAR (*Security-Aware ad hoc Routing*) protocol [YI 02] is also based on the symmetric encryption process. In the beginning it was elaborated to prevent "black hole" attacks which consist of removing all the packets at the malicious node level. Following the example of preceding protocols, SAR is conceived to be employed jointly with reactive protocols such as AODV or DSR. It uses the "trust level" concept to establish the route security. Thus, when a node wishes to establish a route with a certain security level, it generates a new *RREQ* packet indicating the necessary level. Thereafter, the route discovery mechanism differs slightly from the traditional diagram of reactive protocols in the sense that only the nodes satisfying the necessary security level can repeat the request with its neighbors. On the contrary, the request is rejected by the node. Once the route is established at the destination, it generates in return a *RREP* packet with the same security level. If no route guarantees in return the required security level, it can be adjusted by the node source.

Of course, this approach implies the binding of the identity of a node to a certain security level. With this intention, there is a secret key for each security level defined and this must be distributed to the entire node network that complies with this security level. The contents as well as the packet headers are then ciphered with the key so that nodes of lower levels cannot read it. Consequently, even information about the topology can be hidden from non-secure nodes.

The capacity of partitioning the network according to various security levels makes SAR an original protocol. However, it suffers from several important defects. The main defect is in the key distribution, which must be carried out before the installation of the network, using a secure channel. Then, we can imagine that the nodes with higher security levels are used to distribute the keys corresponding to the lower levels. However, this raises the possibility of severe identity usurpation attacks if a node has suddenly been corrupted. Indeed, in this case, the keys of all the lower security levels become obsolete, in fact threatening the total security of the network. In addition, ciphering and deciphering all the packets (including the headers) is a risk that can have a significant impact on the network resources, and this can be used by a malicious node to launch a DoS attack. Finally, an effect inherent in this approach is that the routes are no longer optimal in term of hops. More serious still, the route establishment rate directly depends on the number of trusted nodes but, more especially, on their layout. It is also probable that certain topologies are not adapted to this approach.

13.5.5.3. *ARIADNE*

Considering the disadvantages of the asymmetric coding process, Hu, Perrig and Johnson developed a protected routing protocol, ARIADNE [HU 02], inspired by the traditional DSR protocol and based on ciphering symmetric coding mechanisms. The idea was to propose a protocol which could be implemented on powerful portables as well as on personal assistants, which is why the authors chose to associate it with three methods of authentification, in order to adapt it to the calculation capacities of the nodes:

– use of a shared key between each pair of nodes;

– use of a shared key between each pair of communicating nodes combined with an authentification by diffusion;

– use of digital signatures.

Concerning routing, ARIADNE is very similar to DSR: the nodes establish the routes on demand through the same route discovery process. These routes are then used while the links are valid. If a rupture occurs on a route, each intermediate node can help to solve the problem performing the route maintenance procedure. The route discovery within the ARIADNE protocol can be divided into two parts. The first enables a destination node to check the authenticity of the transmitter of a route request (*RREQ* message). The second consists of using hashing techniques in order to ensure the integrity of the list of the nodes included in the request.

13.5.5.3.1. Route discovery

To explain the discovery of routes, let us suppose that a node *S* launches this procedure to establish a route towards a node *D*, which shares a secret key *K*. To prove to node *D* that each field composing a *RREQ* message is correct, node *S* includes a message authentification code (MAC, mentioned earlier) calculated using the key *K*, as well as a stamp. Thereafter, *D* can easily check the authenticity and the freshness of the message by using the secret key.

However, at the moment of route discovery, the node recipient also needs to authenticate each node included in the request before sending an answer message (*RREP* message). In order to do this, each node authenticates the new contained information in the request using its corresponding keys TESLA. Thereafter, the recipient stores the answer in a record until the nodes send the corresponding keys TESLA. The security condition related to TESLA is checked on the destination level and this includes MAC in the answer to guarantee that the condition was indeed met.

This authentification, although effective, is not enough to guarantee the total routing security. Indeed, a malicious node could very well remove an address in the list of the nodes of a request. Also, ARIADNE uses hashing functions in one sense to counter this threat. Thus, in order to add or remove a node in a list, a malicious node must either capture a request without the address of this node or be able to reverse the hashing function (which is supposed to be unfeasible). For more effectiveness, the address of the authenticating node can be included in the hashing print of the request. Table 13.5 shows an example of the route discovery procedure with ARIADNE.

13.5.5.3.2. Route maintenance

Once again, ARIADNE is inspired largely by the DSR mechanism. Thus, when a node is unable to transmit a packet to the following node on the route after several successive tests, it returns a error route packet (*RRER* message), taking care to have signed it beforehand. Then, each node relays the message normally. If the TESLA protocol is employed for the authentification, this can be slightly delayed. In this case, each intermediate node records the message until the TESLA key is available to authenticate the message.

Comparing the approaches used in ARIADNE, we can note that recourse to the TESLA protocol makes it possible to be freed from the expensive (in terms of overhead) and delicate distribution of private keys. Here, the protocol assumes an exchange of pre-established keys between the nodes. On the other hand, this profit is carried out to the detriment of the reactivity of the protocol since the use of TESLA causes an appreciable increase in the authentification time.

S:	h_0= MAC_{KSD} (REQUEST, S, D, id, ti)
S ? *:	REQUEST, S, D, id,ti, h_0, (), ()
A:	h_1= H[A, h_0]
	M_A = MAC_{Kati} REQUEST, S, D, id, ti, h_1, (A), ()
A ? *:	REQUEST, S, D, id, ti, **h_1**, (**A**), **M_A**)
B:	h_2= H[B, h_1]
	MB = MAC_{KBti} (REQUEST, S, D, id, ti, h_2, (A, B), (M_A))
B ? *:	REQUEST, S, D, id, ti, **h_2**, (A, **B**), (MA, **M_B**)
C:	h_3= H[C, h_2]
	MC = MAC_{KCti} (REQUEST, S, D, id, ti, h_3, (A, B,C), (M_A, M_B))
C ? *:	REQUEST, S, D, id, ti, **h_3**, (A, B,**C**), (M_A, M_B, **M_C**)
D:	MD = MAC_{KDS} (REPLY, D, S, ti, (A, B,C), (M_A, M_B, M_C))
D ?C:	REPLY, D, S, ti, (A, B,C), (M_A, M_B, M_C), **M_D**, ()
C ?B:	REPLY, D, S,ti, (A, B,C), (M_A, M_B, M_C), M_D, (**K_{Cti}**)
B ?A:	REPLY, D, S, ti, (A, B,C), (M_A, M_B, M_C), M_D, (K_{Cti}, **K_{Bti}**)
A ?S:	REPLY, D, S, ti, (A, B,C), (M_A, M_B, M_C), M_D, (K_{Cti}, K_{Bti}, **K_{Ati}**)

Table 13.5. *Route discovery with ARIADNE*

In spite of these protections, ARIADNE is vulnerable to a malicious node that would try to penetrate a route. Indeed, the mechanisms presented do not determine if the intermediate nodes relay the packets for which they were requested. Also, in order to avoid regularly using routes made up of malicious nodes, the authors of ARIADNE recommend choosing the routes according to their performance in terms of packet routing. A classification is thus carried out starting from the returns of the established routes. In fact, it is a matter of associating the traditional mechanisms coding with a model of confidence (detailed in the next section) by considering the reputation of the routes, according to their use. The addition of such an approach proves to be necessary because if the cryptographic mechanisms make it possible to guarantee a good authentification, they do not make it possible to guarantee the legitimacy of routing information and, consequently, the route selection. They do not even count the number of attacks by non-participation. Thus, as example, we showed that ARIADNE makes it possible to secure the routing of the route error messages; however, this protocol does not manage malicious nodes which do not transmit these messages. Such behavior has nevertheless had a very negative impact on the network. It is a limitation of the coding processes in routing security, only guaranteeing a "low level" security. To deal with attacks as complex as non-participation, we will see in section 13.5.8 that other approaches must be used as a complement.

13.5.6. *Protection against data modification*

As we saw in section 13.5.2, in order to guarantee the data integrity, the hashing chains are a very effective tool offering a very satisfactory protection with lower costs compared to the previously detailed cryptographic approaches. Thus, the SEAD protocol proposes to reinforce the DSDV protocol security by using the hashing chains. It enables the prevention of possible attack that artificially increment the hop numbering in the signaling packet header. A node generates a hashing chain and breaks it into several segments of *m* elements: ($h_0, h_1..., h_{m-1})..., (h_{km}, h_{km+1} ..., h_{km+m-1}), ..., h_n$) with $k = m/n - i$, *m* corresponding to the maximum diameter of the network and *i* being the sequence number (Figure 13.7).

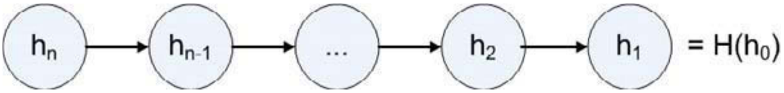


Figure 13.7. *Hashing chain in SEAD*

Since $h_i = H(h_i - 1)$, knowing h_i, it is easy to check h_j authenticity, as long as *j* remains lower than *i*. Moreover, since different hashing functions are used to differentiate diameters and metrics, an attacker can never forge a lower metric value or a greater number of sequences. Finally, the DSDV protocol specifies when a node receives a signaling message; it updates its routing table if the sequence number is larger than or identical to a lower metric. Therefore, SEAD prevents a potential attack artificially decreasing the hop number or incrementing the packet sequence number.

In addition to their work on the SRP, Papadimitratos and Haas also developed a mechanism intended to secure the link state routing protocols, called SLSP (Secure Link State Protocol) [PAP 03]. In the case of SEAD, the protocol uses the digital signatures as well as the one sense hashing chains to guarantee the integrity of the link state updates. SLSP can be used alone, in an independent manner or as an inter-zone routing protocol (IARP), which is a component of the ZRP. The SRP comprises four principal mechanisms: a neighbor monitoring protocol (NLP), a key distribution protocol (PKD), a link state update protocol (LSU) and finally a DoS type prevention attack mechanism.

Thanks to NLP, each node is authenticated near its neighbors by sending through the network a signed duple (IP addresses/MAC addresses). A node can also inform SLSP when the same physical address corresponds to two IP addresses, when two different physical addresses own the same IP address or even when another node uses the same physical address. Then, each node periodically sends a PKD packet inside a zone which contains its certified public key. The link state updates (LSU) are also signed and periodically sent inside the same zone. In order to ensure that PKD and LSU packets do not cross a large number of nodes, each of them included a hop meter. In the case of SEAD and SAODV, the hashing chains are used to protect these meters. Finally, in order to limit the DoS attacks, each node supervises its neighbors and assigns a low priority to the nodes that generate too many updates. The technique here is exactly the same as that used by the SRP protocol. On the other hand, the disadvantage is also the same: the possibility that an attacker usurps the identity of a victim and floods its vicinity with updates which will seem to be emitted by the victim. Moreover, although the victim always has the possibility of detecting the attacks, using the detections of multiple physical addresses of the NLP mechanism, it is highly probable that it cannot react. Finally, SLSP does not consider possible attackers that could forge the erroneous metrics or even create tunnels.

13.5.7. *Protection against "tunnel" attacks*

The cryptographic processes employed in the preceding diagrams effectively avoid a large number of attacks. However, none of them, having asymmetric or secret keys, can solve the tunnel (or *wormhole*) problem presented in section 13.4.2.2. Indeed, even if all the route entries seem perfectly identified, nothing prevents a node in charge of transferring a packet, from requesting a route in parallel, with an ally node and of transferring the encapsulated packet towards it, which will then be in charge of forwarding all of them to the destination. Several solutions can be planned to solve this problem. First, during the route discovery process, the *RREQ* packet is flooded through the network and since the tunnel inevitably passes by a more significant number of nodes, if the destination establishes the time like the selection criterion of the route, it is extremely difficult for the route passing by the tunnel to be selected because it will be slower. Also, we can imagine that the nodes located around the first malicious node relay the packet to the destination even before the malicious node has the time to encapsulate it in a tunnel. However, these solutions are not viable in all circumstances and in particular if the ally node is an essential node on the path. This is the reason why a Carnegie Mellon University team developed a parade based on the localization of the nodes in one part and on their temporal synchronization in another part: *Packet Leashes* [HU 03]. In the initial version, the packet transmitter includes its localization and a stamp corresponding to its clock during the emission. When the destination receives the packet, it compares these values with its own localization and its clock at the time of the reception of the packet. If the two nodes are synchronized except for a coefficient, the recipient can estimate, starting from the temporal markers, an approximation of the distance which separates them and thus check if this corresponds to the real distance. Nevertheless, there are certain circumstances for which this technique is ineffective, for example, when obstacles are involved between two close nodes. In such circumstances, a protection diagram based on the correlation between distances and transfer time could not prevent a tunnel attack. This is the reason why the researchers developed a second diagram where only the temporal metric is considered. With this intention, the nodes must be synchronized with a margin of a few microseconds, even nanoseconds; this difference must be known by all the nodes. The process is then identical: when a packet is sent, a time record (emission clock) is included. Then, the destination node compares this value with its clock at the packet reception time, and it is able to determine if the distance covered is reasonable by comparing the transfer time with the wave propagation speed. An alternative consists of including in the packet an expiration date, beyond which it the packet must be purely and simply ignored. The difference between the two approaches lies in the fact that when the geographical position of the nodes is used, the synchronization does not have to be precise. Additionally, the fact of knowing the position of the nodes makes it possible to detect a node which claims to be in several places at the same time. Of course, it will be necessary in all cases to complement the schema with a field authentification mechanism in order to ensure they were not falsified. Unfortunately, the mechanism shows several defects which could make its implementation difficult. First, it requires a means of checking that the localizations advanced by the nodes are exact. Indeed, since each node in the network must register its localization in the packet that it transmits, all the nodes have a certain time where they have a more or less precise knowledge of the topology of the network and in particular of the positions of the other participants. Consequently, a malicious node can assert a false localization very well which could justify an abnormal increase in the transfer time. To avoid this threat, we can imagine a distributed surveillance of the nodes in which their position would be confirmed by their neighbors. We can also imagine a localization service like the Global Positioning System that would also be the certification authority. This solves the most important problems regarding the relevance of a temporal synchronization within the framework of the ad hoc model. Indeed, if the protocol is based on a shared access with contention resolution support (for example, 802.11 MAC), there could be several time lapses before a packet is effectively sent. Also, the electronic signature generation with a 1,024 bit RSA key can take about 10 ms on a recent processor. Thus, according to the quantity of data to be ciphered, it is sometimes illusory to preserve the precision of a few microseconds required by the scheme.

13.5.8. *Mechanism based on reputation*

The previously detailed mechanisms prove to be effective to ensure the traditional security functionalities such as confidentiality, integrity and especially authentification. Thus, they make it possible to prevent many attacks which disturb the routing process considerably. On the other hand, they do not appear at all adapted to solving the non-participation problem of the nodes. Indeed, the very effective cryptographic mechanisms do not ensure that a node takes part in the routing process by relaying all the packets. However, in the context of ad hoc networks, the cooperation between the nodes is a vital functionality on which the networks are based. This is why, in addition to the security mechanisms, certain protocols aim more specifically at cooperation incentives. Among these, we generally distinguish two categories: those which are based on node reputation worked out over time according to the observations and those that establish a virtual payment system.

13.5.8.1. *Micro-payment mechanism*

The concept consists of charging the services which the nodes wish to reach in exchange for virtual credits. To obtain these credits, each node must provide services to the other nodes. The credits are spent then later to buy services. If a node does not have enough credits to buy the minimum service, this means that it has not participated enough in the good progress of the routing process.

The NUGLETS protocol [BUT 01] is registered in this perspective. Its objective is at the same time to incite the nodes to participate and to limit the network flooding since nodes have to pay. So, in order to secure the virtual credits, the protocol assumes the existence of inviolable materials. The principal assumption is that no attack can be launched against the virtual currency. Two models are specified by the protocol. In the first, a node wishing to send a packet must incorporate sufficient credits in it as a preliminary step. Thereafter, each intermediate node on the route takes a quantity of credits. If the number of credits is insufficient, the packet is rejected. The interest of this approach is in limiting DoS attacks, as no node can be allowed to finance a flood. On the other hand, it implies that each node knows in advance the number of nodes on the route. If the number of credits is too large, they are wasted. Conversely, the packet is lost and more credits must be spent for its re-emission. In the second model, the routing objectives are the transactions since in fact the destination nodes must pay to receive the packets which are directed to them. Indeed, each node buys the received packets of its upstream neighbor and the packet destination buys the last intermediate node. This approach suffers a disadvantage even more logical than the preceding one, since it does not prevent an attacker from flooding the network. On the contrary, a node can be tempted to relay many packets towards many nodes in order to maximize its profits at the time of the transactions.

In general, these protocols do not fit sufficiently into the ad hoc model to be effective. First, they do not consider the node mobility enough. Indeed, if an intermediate node leaves the route, the packet is lost and also the investment in term of credits, whether for the transmitter (case of the first model) or for the last intermediate node (second model). Finally, this approach poses large problems concerning the operation even of the routing protocol. Thus, in the case of a reactive protocol, the nodes can be tempted not to send *RRER* error messages during the link failure detection because they would then have to pay for that. In the case of a proactive protocol, this would be related to the control messages which would then become too expensive. Also, the protocol would also ensure that the nodes cannot steal the credits by simply spying on its neighbor’s conversations.

13.5.8.2. *Trust-based mechanism*

The aim of these protocols is to provide node classifications in order to differentiate the “good” nodes, which have a good reputation because they cooperate regularly, from the “bad” nodes which adopt a selfish behavior.

The CONFIDANT (*Cooperation Of Nodes – Fairness In ad hoc DynAmic NeTworks)* protocol [BUC 02] is included in this category. It uses an auto-organized public key infrastructure inspired by PGP protocol. The aim of CONFIDANT is to treat the malicious and selfish nodes at the same time through supervision and analyzing two routing processes knowing the information transfer and the discovery of neighbors. It is then conceived to be used jointly with a reactive protocol, typically DSR. CONFIDANT is composed of four complementary elements: the monitor, the confidence monitor, the reputation system and the route management mechanism. The role of the monitor consists of ensuring that the node neighbors to which it is attached relay the packet correctly. When the monitor detects an anomaly or an inconsistency, it informs the reputation system which on its side maintains up to date lists of notes for each node observed. The lists can possibly be exchanged between the nodes. Thus, if a list is received from a node having great confidence, the receiver can directly record the information inside its own list. In the contrary case – if the list is sent by a suspect node – the receiver can completely ignore it or can accept it but give it less importance than a list received from a secure node. Finally, the route management mechanism determines the surest routes starting from the lists of excluded and trusted nodes. Moreover, it can decide to refuse to relay the requests coming from badly noted nodes.

Concerning the management of trust, the approach is inspired by the one used in PGP. Thus, the nodes have four trust degrees: *friend, marginal, unknown* or *enemy*. Each node records its friends in a dedicated list. Later, if a node *A* has managed to detect a malicious behavior of a node *B*, node *A* will inform all the friends contained in the list using a signed alarm message. Such messages can be sent through the network. Then each node decides if the message must be taken into account, according to whether the transmitter is trusted or not. A improved version of CONFIDANT uses a Bayesian approach in order to more effectively differentiate true alarms from lies intended to decrease the reputation of a node.

One of the principal motivations for a node not to participate in the routing process is energy saving. This is sometimes a critical resource, so certain nodes can try to save energy adopting a selfish behavior. To combat this phenomenon, Michiardi and Molva developed the CORE (*a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks*) protocol [MIC 02]. The objective is not to definitively exclude the nodes but to encourage them to participate, rejecting their packets until they cooperate with the routing process. CORE assumes as a hypothesis that: node identities are unique and non-modifiable, that an adapted routing mechanism is also capable of securing the neighbor discovery phase and finally that the traffic inside the network is sufficiently dense. The operation is very similar to CONFIDANT, in fact the monitors analyze the traffic and transmit the results to a reputation management system. The reputation exchange between nodes is optional here. Moreover, the authors validated their approach at the same time by simulation and game theory.

CORE unfortunately suffers from important defects. First, it does not really solve the non-participation problem. Certainly, the selfish nodes are going to see their packets systematically rejected and in this aspect the protocol is effective. However, on the other hand, large quantities of data remain lost, decreasing the output of the network significantly. Lastly, the protocol rests on very strong assumptions (secure routing, unique and non-usurpable addresses) which still remain to be fixed. In fact, it is a common disadvantage of all the protocols based on the reputation. Indeed, this rests on the information observed on the nodes and consequently requires a strong authentification mechanism to assign the notes to the legitimate nodes. Moreover, it is difficult to avoid the problem of “false accusation” in which a malicious node generates false alarms to put honest nodes on a black list. This type of mechanism is also potentially very vulnerable to ally nodes which agree among them to give a good evaluation and then to affect their counterparts, giving bad notes to honest nodes. Finally, a common disadvantage to all the protocols based on a trust model is that they need time to be effective. The trust is established slowly between several nodes and so an attack can more easily be launched at the beginning of the installation of the network.

The following table summarizes the possible defenses offered by the different security protocols described in this chapter. It should be noted that the protocols tend to target certain attacks in particular, so none of them offers an effective protection to all the attacks described here. The conclusion we can draw is that the most promising solution is probably in the use of a protocol combining these approaches: a protocol based on cryptography to ensure the authentification of nodes and the integrity of control messages and a protocol based on trust models to detect and then ignore nodes presenting a malicious behavior.

	Indiscreet listening	**Usurpation**	**Gray hole**	**Black hole**	**Tunnel**	**Non-cooperation**
ARAN	Yes	No	Yes	Yes	Yes	Yes
ARIADNE	Yes	No	Yes	Yes	Yes	Yes
SRP	No	Yes	No	No	Yes	No
CORE						
SAODV	No	No	Yes	Yes	Yes	Yes
CONFIDANT	Yes	No	No	No	Yes	No
Packet Leashes	Yes	Yes	Yes	Yes	No	Yes

Table 13.6. *Secure protocols: attack prevention*

13.6. Auto-configuration

If we assume that the network is connected to a wired network through a bridge, then it is easy to assign a unique address to all nodes. Indeed, this can relay the address request to a DHCP server or even carry out a NAT (*Network Address Translation*) conversion to ensure a compatibility in the case of the use of a private prefix (MANET mask).

For small-sized closed networks (the terminals can be managed by the same administrator), it is perfectly possible to manually allocate the addresses, as is the case for certain domestic networks for example. However, such a procedure becomes more difficult when managing open networks (e.g. citizen networks for example, whose nodes are free to join or leave the network) or of networks containing several hundreds of nodes. In addition, the first vocation of ad hoc networks as they were originally conceived is to be autonomous. Thus, such a network must be able to be set up with the minimum amount of human intervention.

Until now, the majority of research on ad hoc networks was rather oriented towards the improvement of the performance of routing protocols. This is why the majority of the protocols standardized by the IETF do not consider the way in which nodes acquire their address within the network. However, addressing is an important stage in the operation of the network because it is the base condition of the routing reliability. In addition, according to the description of attacks in section 13.3.1, the way in which addresses are allotted can have a significant influence on the network security as most attacks use identity usurpation. This is why several approaches have recently been created to propose reliable and effective mechanisms especially automated for ad hoc networks. Since 1999, there has been a group dedicated to the problems of auto-configuration in the IETF [ZER 04]; however, it concentrates mainly on the environments of limited size such as corporate or domestic networks, as well embedded systems and not especially on ad hoc networks. However, the automatic attribution of addresses within the ad hoc network framework is much more problematic than in wired networks or in wireless networks with wired infrastructure. The strong mobility of the nodes, the absence of fixed entities and the openness of ad hoc networks make the design of an automated mechanism much more delicate. The traditional approaches (DHCP [DRO 97] and SAA [THO 98]) appear unsuited to this model. Indeed, to be reliable in an ad hoc network, such a mechanism must be able to manage new events. Thus, a node can enter a network, automatically acquire an address and then leave the network in an inopportune way, because its battery is exhausted or because it moved away from the network until it was out of the radio wave coverage of another node. We can then consider that the address is definitively lost, or on the contrary, is re-assigned to another node. In the first case, we perhaps gradually exhaust the available addresses while in the second case, we expose ourselves to the risk of conflict if the node in question moves again within range of another.

Another event, which is much more problematic, is the fusion of networks. In such a case, if two networks (independent or resulting from the previous partition of the same network) approach one another until they form one network. Then, since the attribution of the addresses is independent from one network to another, it is perfectly possible that the same address was allotted to two different nodes in their respective networks. In the routing plan, a fusion corresponds to the arrival of one or more new participants in the network. In route conflict, the nodes can simply note abrupt topology changes implying duplicated addresses (the control packets describe different routes for the same address), as they do not *a priori* have any reason to notice that a fusion occurred and generated a conflict.

Based on these considerations, the auto-configuration mechanisms dedicated to ad hoc networks can be divided into two principal categories: first of all, the mechanisms with detection and resolution of conflicts, i.e. which allocate the addresses initially then solve the possible conflicts later on; secondly, protocols with conflict avoidance.

13.6.1. *Conflict detection protocols*

In this type of approach, a new free address is *a priori* assigned to a node arriving in the network (an address conflict can occur if two new nodes perform a request almost simultaneously). In a second example, the new arrival launches a conflict detection mechanism in order to make sure that its address is not already used. If this is the case, the node selects a new address and repeats the procedure until it obtains a final address.

13.6.1.1. *Distributed Dynamic Host Configuration Protocol*

A good example to illustrate this approach is the *Distributed Dynamic Host Configuration Protocol* (DDHCP) proposed by Ramakrishnan, Thoppian and Prakash in [RAM 06]. DDHCP maintains a table of the allocation of common addresses in a distributed manner. The operation is as follows: when a new node (applicant) wishes to obtain an address, it sends a request to all the nodes in the neighborhood and waits for an answer from one or several nodes. If no answer is received, then the applicant concludes that it is the first node of a new network. It assigns its own address and can become an initiator for the next node. If it receives one or more answers, it selects one of the nodes as being its initiator. This node assigns a temporary address to it (with the test) and requires validation near the other nodes of the network. If the address is accepted, by all the other nodes of the network, it is definitively assigned. In the contrary case, another address is selected until all the nodes accept it. The initiator is thus used as a relay between the applicant and the network.

DDHCP is particularly adapted to proactive routing protocols since the periodic packet emission makes it possible to keep the allocation table consistent. Moreover, this protocol includes mechanisms to take into account the partitioning and the fusion of networks. Indeed, a network identifier is generated and sent by the smaller address node regularly. In this way, when two networks merge, they can be differentiated by their identifier. Then, an intrusion detection mechanism solves the possible conflicts.

13.6.1.2. *IP address autoconfiguration for ad hoc networks*

This mechanism, proposed by Perkins, Wakikawa, Malinen, Belding-Royer and Suan in [PER 01], schematically consists of choosing an address in a quasi-random way and then launching a Duplicate Address Detection (DAD) mechanism specific to ad hoc networks, in order to solve any possible conflict. It should be noted that this protocol is defined to function as well with IPv 4 and with IPv6; however, we describe here only the mechanism used with IPv4.

When a new node (applicant) enters the network, it selects two addresses in the IPv4 prefix dedicated to ad hoc networks (169.254/16): a temporary and an "test" address. The first is the address used by the node for the limited period during which it will launch the duplication detection mechanism. It must be taken among the first 2,048 values of the prefix, those being reserved by definition, with the temporary addresses. The second is selected from the remaining addresses; this is where the DAD mechanism acts. Thus, when the node chooses its two addresses, it sends through the network, using the temporary address like identity, an address request (AREQ) which contains the address chosen by the test. Each neighbor examines the address included in the packet then retransmits it in its turn to its neighbors, creating an entry in their routing table in order to be able to relay a possible answer. If a network node realizes that the selected address is the same one as its own, then it answers the request, in unicast, using an AREP (address reply) message. When the petitioning node receives this message, it understands that the address that it chose is already taken; it then chooses another and starts the process with this new address again. On the contrary, if no AREP message is received after a certain duration, the applicant concludes from this that its address is free. It then chooses it as a final address and releases its temporary address.

It should be noted that the mechanism is the same as that for IPv6, the only difference being the format of the message (ICMP for IPv4, request of neighbors and warning messages for IPv6).

This protocol has the advantage of being relatively simple to manage. However, it suffers from important defects. First of all, it is mainly directed towards reactive protocols, because of the use of requests. Then, the neighbor detection mechanism is not started during the address attribution, which implies that conflicts can occur later on; when a configured node leaves the network temporarily to then return for example (during its absence, it does not receive the requests and its address can be re-assigned) or in the case of a network fusion. The authors recommend starting the conflict detection procedure again. However, this makes it necessary to detect these conflicts and to choose the nodes to be reconfigured, which are not covered by this protocol. Moreover, the case where there are a great number of duplications can have a considerable impact on the network. Finally, on very dense networks, the number of attempts can be high before a free address is obtained, which has an effect on the time employed to obtain an address.

There are several other approaches dedicated to address duplication detection. We will note in particular Weniger and Al [WEN 03], Jeong and Al [JEO 05] and Vaidya [VAI 02].

Contrary to this approach, protocols with conflict avoidance assign addresses without the risk of duplication. This can be accomplished thanks to the use of pools of disjoined address which guarantee the attribution of single addresses. This way, addresses duplication detection mechanisms, such as those described in the preceding protocols, are no longer necessary.

13.6.2. *Protocols avoiding conflicts*

13.6.2.1. *Dynamic Configuration and Distribution Protocol (DCDP)*

This protocol, described by Mokhsin and Prakash in [NES 02], is based on the concept of binary trees, used at the beginning of memory management in operating systems. The principle is as follows: each node inside the network has an address pool which it can give to new arrivals. Thus, when a new arrival requires an address, the nearest network node assigns an address to it then divides its address pool into two and a give a half to it. The applicant can then in its turn configure a new applicant. Concerning the departure of nodes, two cases can occur: “soft” departures and “brutal” departures. In the first case, a node informs one of its neighbors of its imminent departure and thus gives the neighbor its address pool. Thereafter, either this node keeps this pool, or it therefore transmits it to the initiator of the node. In the case of a brutal departure, the node therefore does not have time to inform its neighbors. In this case, in order to avoid losing complete address pools, the protocol specifies a periodic update mechanism and synchronization of the address pools.

Moreover, DCDP makes it possible to manage the phenomena of network partitioning. With this intention, each network has an associated identifier. Thus, when a network is divided into two distinct entities (following a topology change for example) each node continues using only the address pools they have. If one of the sub-networks does not have any more addresses in reserve, it assigns a new identifier. Thereafter, if the two networks merge, the one which has the greatest quantity of addresses must be reconfigured.

This category of protocols has the advantage of allocating single addresses and does not need an expensive address detection mechanism. The addresses are available immediately and the protocol manages the partitioning effectively. On the other hand, the networks fusion is still a delicate problem since, on such occasions, great quantities of nodes can be forced to reconfigure it. The node synchronization can then become very complex. Another problem is related to address pool fragmentation. Indeed, when the nodes give pools, it is not guaranteed that they will be found later. Thus, after a certain time, certain nodes can have full pools while others will have almost exhausted pools. The work of Zhou [ZHO 03] and Misra [MIS 01] also constitutes original and interesting solutions for conflict avoidance.

13.6.3. *Auto-configuration and security*

Security is the great weak point of auto-configuration mechanisms. Indeed, the field has only been explored for a very short time. As in the case for research on routing protocols, research on auto-configuration is especially concentrated on the improvement of existing solutions. Thus, from now on there exist several protocols with very different approaches (without or with resolution of conflicts, with or without state, hybrids, etc.) but which do not yet manage the problems regarding security. Nevertheless, lately, certain solutions [CAV 04, WAN 05] are more interested in the security of automatic addressing. They use coding processes identical to those used by routing protocols, namely symmetric and asymmetric cryptographies. There is no doubt that some other approaches, which are surer and more powerful still, will be created in the near future.

13.7. Conclusion

We have seen in this chapter that all the traditional routing protocols in ad hoc networks (AODV, DSDV, OLSR) are particularly vulnerable to a great number of attacks; which can go from the capture of sensitive information to the complete paralysis of the network. Where the use of wireless networks has had unprecedented success (in particular thanks to Wi-Fi, WiMAX and mobile phones) and where, in parallel, the number of attacks against the computing systems are also raised considerably, network security became crucial. Thus, even if ad hoc networks constitute a very promising solution with the current problems involved in the mobility of users and networks themselves, their development is limited today by the absence of sufficiently effective security mechanisms to cover the present needs in data protection such as those required by commercial applications.

In this context, the research works which were formerly concentrated on the improvement of performance are reorientated today to the security of routing protocols. However, the processes employed are often very different from one algorithm to another and the inherent characteristics in the ad hoc model such as mobility and the absence of infrastructure, completely reconsider the traditional security used in the wired field and oblige the designers to make compromises between the security of protocols and performance constraints. Indeed, in a completely distributed context, these mechanisms must be adapted consequently, with the risk of generating a consequent overload of the network. This is the reason why none of the recent elaborate secure protocols is proven to be sufficiently satisfactory to be imposed as a standard. All appear either too expensive in terms of resources (time, flow, memory, etc.) or too complex to be established. The problems arising from the exchange of secret keys or the installation of group keys are often evaded by the creators of protocols that consider these stages as independent. We will however see in Chapter 14 that certain solutions exist and constitute an essential precondition for the use of ciphering processes.

The experience in the field of cryptography already showed that the design of protected protocols is often subject to faults that are difficult to detect, even when assuming that the code is perfect. Thus, even if the protocols detailed in this chapter make it possible to appreciably improve the security of the routing process, on the other hand they offer an increased vulnerability to DoS attacks. Now the analysis of cryptographic is complex because the configurations to be considered together are immense, even infinite: it is necessary to take into account an unspecified number of sessions, an unspecified size of the messages, session interlacing, and algebraic properties of the coding or the data structures. This is the reason why a good number of works are currently concerned with the automation of ad hoc protocol verification starting with their specifications. Thus, a current complementary method of research consists of generating methods and verification tools based on trace analysis, the exploration of symbolic models, or the generation of tests. However, still, the strong mobility of nodes which characterizes ad hoc networks constitutes a problem of size for the actual model verification approaches such as model checking [BHA 02].

Nevertheless, no protocol can avoid all the attacks detailed here; the majority are satisfied with targeting a simple threat (non-participation, identity usurpation, traffic deviation) and providing a relatively adapted solution. This is why the most probable tendency is a combined use of various approaches (symmetric/asymmetric cryptography, trust models) within the same protocol to make the network safe. Another possible tendency is the appearance of a cleavage which would see the appearance of two distinct types of network: closed networks and open networks. The first would be restricted as a group of individuals defined within the same entity (military unit, network of a supplier of access, a company, etc.). The access control on the components of the network would guarantee high security but to the detriment of flexibility (the nodes should be configured before entering the network, to allow the installation of keys and the attribution of an address, for example). The second would be characterized by completely open access to the network (as for vehicular or citizen networks, for example). On the other hand, security could not then be completely guaranteed.

13.8. Bibliography

[BEN 97] BENNETT F., CLARKE D., EVANS J., HOPPER A., JONES A. LEASK D., *Piconet: Embedded Mobile Networking*, IEEE Personal Communications, 4(5), October 1997.

[BHA 02] BHARGAVAN K., OBRADOVIC D., GUNTER C. A., "Formal verification of standards for distance vector routing protocols", *J. ACM* 49, 4 (Jul. 2002), 538-576. DOI= http://doi.acm.org/10.1145/581771.581775.

[BUC 02] BUCHEGGER S., LE BOUDEC J.Y., "Performance analysis of the confidant protocol", in *Proc. ACM 3rd International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'02)*, pages 226-236, 2002.

[BUT 01] BUTTYAN L., HUBAUX J.-P., Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks, Technical Report DSC/2001/001, Swiss Federal Institute of Technology, 2001.

[CAV 04] CAVALLI A., ORSET J.-M., *Secure Hosts Autoconfiguration in Mobile Ad Hoc Networks*, icdcsw, pp. 809-814, 2004.

[CLA 03] CLAUSEN T., JACQUET P., *Optimized Link State Routing Protocol (OLSR)*, Project, Hipercom, INRIA, www.ietf.org/rfc/rfc3626.txt, RFC-3626, 2003.

[COP 02] COPPERSMITH D., JAKOBSSON M., "Almost optimal hash sequence traversal", in *Proceedings of the Fourth Conference on Financial Cryptography (FC'02)*, Lecture Notes in Computer Science, 2002.

[COR 99] CORSON S., MACKER J., *Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*, http://www.ietf.org/rfc/rfc2501.txt, RFC2501, 1999.

[DAH 02] DAHILL B., LEVINE B., ROYER E., SHIELDS C., "A secure routing protocol for ad hoc networks", in *Proceedings of the 10th Conference on Network Protocols (ICNP)*, November 2002.

[DGA] Ministère de la Défense, Délégation générale pour l'armement, centre d'électronique de l'armement, http://www.defense.gouv.fr.

[DRO 97] DROMS R., *Dynamic Host Configuration Protocol*, Network Working Group RFC 2131, March 1997.

[GUE 02] GUERRERO ZAPATA M., ASOKAN N., "Securing ad hoc routing protocols", in *Proceedings of the 2002 ACM Workshop on Wireless Security (WiSe 2002)*, pages 1-10. September 2002.

[HAA 02] HAAS Z.J., PEARLM M.R., SAMAR P., *The Zone Routing Protocol (ZRP) for Ad Hoc Networks*, IETF MANET Internet Draft, July 2002.

[HU 02] HU Y.C., PERRIG A., JOHNSON D.B., *Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks*, MobiCom 2002, September 23-28, 2002, Atlanta, Georgia, USA.

[HU 02] HU Y.C., JOHNSON D.B., PERRIG A., “SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks”, in *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002)*, pp. 3-13, IEEE, Calicoon, NY, June 2002.

[HU 03] HU Y.C., PERRIG A., JOHNSON D.B., *Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks*, INFOCOM 2003.

[HUB 01] HUBAUX J.-P., BUTTYAN L., CAPKUN V., “The quest for security in mobile ad hoc networks”, in *The 2nd ACM Symposium on Mobile Ad Hoc Networking and Computing*, October 2001.

[IE 97] Institute of Electrical and Electronics Engineers 802.11 Working Group, http://grouper.ieee.org/groups/802/11/main.html.

[IET] The Internet Engineering Task Force, http://www.ietf.org/home.html.

[JAK 02] JAKOBSSON M., “Fractal hash sequence representation and traversal”, in *Proceedings of the 2002 IEEE International Symposium on Information Theory (ISIT ’02)*, pages 437-444, July 2002.

[JEO 05] JEONG J., *Ad Hoc IP Address Autoconfiguration*, draft-jeong-adhoc-ip-addr-autoconf-04 (work in progress), February 2005.

[JOH 96] JOHNSON D., MALTZ D., “Dynamic source routing in ad hoc wireless networks”, in *Mobile Computing* (ed. T. Imielinski and H. Korth), Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[LEE 02] Lee S., HAN B., SHIN M.. *Robust Routing in Wireless Ad Hoc Networks*, Computer Science Department, University of Maryland.

[LSF] Association LilleSansFil, “Citizen network of Lille Metropole”, http://www.lillesansfil.org/.

[MAN] IETF Mobile Ad hoc Networks Working Group, http://www.ietf.org/html.charters/manet-charter.html.

[MIC 02] MICHIARDI P., MOLVA R., Core: “A Collaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks”, in *Communication and Multimedia Security 2002 Conference*.

[MIS 01] MISRA A., DAS S., MCAULEY A., DAS S., *Autoconfiguration, Registration, and Mobility Management for Pervasive Computing*, IEEE Personal Communications, 2001.

[MS 01] MINER S., STADDON J., "Graph-based authentication of digital streams", in *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 232–246, May 2001.

[MON 02] MONTENEGRO G., CASTELLUCCIA C., "Statistically unique and cryptographically verifiable identifiers and addresses", in *Proc. ISOC Symposium on Network and Distributed System Security (NDSS 2002)*, San Diego, February 2002.

[MOY 89] MOY J., The open shortest path first (OSPF) specification, Technical Report RFC-1131, SRI Network Information Center, October 1989.

[NES 02] NESARGI S., PRAKASH R., *MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network*, InfoCom 2002, June 2002.

[NRL] Naval Research Laboratory, Networks and Communications Systems Branch, http://cs.itd.nrl.navy.mil/work/drp/index.php.

[OGI 04] OGIER R. *et al*., *Topology Dissemination Based on ReversePath Forwarding (TBRPF)*, Request for Comments 3684, February 2004. http://citeseer.ist.psu.edu /ogier04topology.html.

[OLC] One Laptop Per Child project. http://www.laptop.org/.

[PAP 02] PAPADIMITRATOS P., HAAS Z.J., "Secure routing for mobile ad hoc networks", *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, San Antonio, TX, January 27-31, 2002.

[PAP 03] PAPADIMITRATOS P. , HAAS Z. J., "Secure link state routing for mobile ad hoc networks", in *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (Saint'03 Workshops)* (January 27-31, 2003).

[PER 00] PERRIG A., *et al*., "Efficient authentication and signing of multicast streams over lossy channels", in *Proc. IEEE Symp. Security and Privac*y, IEEE Press, 2000, pp. 56–73.

[PER 01] PERKINS C., WAKIKAWA R., MALINEN J., BELDING-ROYER E., SUAN Y., *IP Address Autoconfiguration for Ad Hoc Networks*, IETF document, November 2001.

[PER 94] PERKINS C., BHAGWAT P., "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers", in *Proc. of the ACM SIGCOMM*, October 1994. http://citeseer.ist.psu.edu/perkins94highly.html.

[PER 99] PERKINS C., ROYER E.M., "Ad hoc on-demand distance vector routing", in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, New Orleans, LA, February 1999, pp. 90-100.

[PLE 04] PLESSE T., LECOMTE J., ADJIH C., BADEL M., JACQUET P., *OLSR Performance Measurement in a Military Mobile Ad hoc Network*, ICDCS Workshops 2004: 704-709.

[RAM 06] RAMAKRISHNAN THOPPIAN M., PRAKASH R., “A distributed protocol for dynamic address assignment in mobile ad hoc networks”, *IEEE Transactions on Mobile Computing*, vol. 05, no. 1, pp. 4-19, January, 2006.

[STA 99] STAJANO F., ANDERSON R.J., “The resurrecting duckling: security issues for ad hoc wireless networks”, in *Proceedings of the 7th International Workshop on Security Protocols* (April 19-21, 1999).

[SW] Seattle Wireless, community-based 802.11b peer-to-peer network, http://www.seattlewireless.net/.

[THO 98] THOMSON S., NARTEN T., *IPv6 Stateless Address Autoconfiguration*, Network Working Group RFC 2462, December 1998.

[TSI 01] TSIRIGOS A., HAAS Z., “Multipath routing in mobile ad hoc networks or how to route in the presence of topology changes”, in *Proceedings of IEEE MILCOM 2001*.

[VAI 02] VAIDYA N., “Weak duplicate address detection in mobile ad hoc networks”, in *Proc. ACM Int’l Symp. Mobile Ad Hoc Networking and Computing (MobiHoc)*, June 2002.

[WAN 05] WANG P., REEVES D. S., NING P., “Secure address auto-conguration for mobile ad hoc networks”, in *Proceedings of the the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services – Volume 00* (July 17-21, 2005).

[WEN 03] WENIGER K., “Passive duplicate address detection in mobile ad hoc networks”, in *Proc. IEEE WCNC 2003*, March 2003.

[YI 02] YI S., NALDURG P., KRAVETS R., “A security-aware ad hoc routing protocol for wireless networks”, in *The 6th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2002)*, 2002.

[ZER 04] IETF Zero Configuration Networking Group, http://www.ietf.org/html.charters /OLD/zeroconf-charter.html.

[ZHO 99] ZHOU L., HAAS Z., “Securing ad hoc networks”, *IEEE Network Magazine*, vol. 13, November/December 1999.

[ZHO 03] ZHOU H., NI L., MUTKA M., “Prophet address allocation for large scale MANETs”, *Proceedings of INFOCOM 2003*.

Chapter 14

Key Management in Ad Hoc Networks

14.1. Introduction

Spontaneous networks are networks where entities can easily connect to each other without any pre-established infrastructure or any human intervention. The development of such spontaneous networks is nowadays possible and even a reality thanks to (1) the large base of existing wireless networks, (2) the emergence of new supporting technologies and standards (e.g. 802.11[1], WiMAX[2], etc.), (3) the increasing availability and reduced cost of autonomous and advanced terminals (phones, PDAs, etc.) and (4) an ever-growing experience and success stories in large deployments of multi-hop spontaneous networks. Ad hoc networks are a perfect illustration of this concept of spontaneousness, where each node actively contributes to the network operations, including discovery, data routing, QoS maintenance or content provision. A Mobile Ad Hoc Network (MANET) consists of a large population of mobile nodes, moving within an unspecified (or sometimes well specified) area, using wireless communication channels in a hop-by-hop manner, without the aid of any fixed infrastructure or centralized administration. The equipment that is generally used within a MANET is characterized by limited capacities in terms of bandwidth, energy and computation power (both CPU and memory).

MANET networks are dynamic in both space and time. They offer a large flexibility. However, this flexibility, associated with the vulnerability of wireless

Chapter written by Mohamed SALAH BOUASSIDA, Isabelle CHRISMENT and Olivier FESTOR.

[1] http://grouper.ieee.org/groups/802/11.

[2] http://www.wimaxxed.com.

communications, makes it necessary to secure data as well as the participating entities. Indeed, the use of wireless links makes the MANETs vulnerable to passive and active attacks. Passive attacks allow malicious non-authorized entities to access confidential data, whereas active attacks can lead to the deletion or modification of messages, the injection of new malicious messages, identity usurpation and consequently the violation of the main security services, namely availability, integrity, authentication and non-repudiation.

In parallel to the development of ad hoc networks, we note over the last decade a large deployment of multicast communications, dedicated to cooperative applications like audio-video conferences as well as one-way streaming services. The deployment of ad hoc networks, associated with the availability of multicast services, raises new challenges towards the establishment of secure communication architectures. Today, the most suitable solution to ensure secure group communications within ad hoc networks is the establishment of a group key management protocol, guaranteeing data integrity and confidentiality, in addition to the authentication and the access control of the group members. These protocols are studied in this chapter.

The remainder of the chapter is structured as follows. We start with a focus on the authentication establishment between mobile ad hoc nodes, a challenging issue within this environment, and mandatory for the other security services. We then present an analysis of the state of the art on multicast communications security within MANETs.

14.2. Authentication issue within ad hoc networks

An authentication service enables a node to prove its identity to any other entity in the network. Without authentication, a malicious node can communicate with other nodes and can easily access unauthorized confidential resources.

To allow ad hoc nodes to communicate securely, mutual authentication is required between these two nodes, being a pre-requisite of the activation of any other security service like confidentiality and access control services.

The authentication service has to perform two phases: (1) authentication establishment (the initial step to define the exchange conditions) and (2) authentication management. The authentication establishment is itself divided into three steps: (1) the distribution of a common secret, (2) the establishment of a secure channel between the participating nodes using the common secret distributed previously, and (3) the exchange of encryption keys to ensure session confidentiality, if necesssary.

In this section, we present the different authentication approaches in MANETs.

14.2.1. *The threshold cryptography technique*

The approach given in [ZHO 99] aims to solve the problems induced by the absence of infrastructure within ad hoc and sensor networks, making use of a public key infrastructure (PKI) very difficult, if not impossible.

To establish secure communications between nodes within a wired network using a PKI, each node holds public and private keys, provided by the certification authority (CA). The CA holds similarly public and private keys (K, k). The CA is always available within the network, because both the public and private keys of each network node have to be updated periodically, in order to decrease the risk of malicious attacks. The CA is also in charge of revoking the public key of an untrusted node. Within an ad hoc network, having only one CA represents a vulnerability point. Indeed, if it not available, nodes are unable to prove the authenticity of the public keys of the peering nodes and consequently cannot establish secure communications between them. Attackers can also use this vulnerability to compromise the entire network.

A naive solution consists of duplicating the CA within ad hoc networks. However, this solution increases its exposure to being compromised. The threshold cryptography [ZHO 99] proposes a more flexible approach: the new key management service having the configuration (n,t+1) consists of *n* special nodes, called servers, available in the ad hoc network, and sharing the ability to generate certificates for the other nodes. t+1 valid partial signatures are required to construct a valid complete signature. Each server i holds its public and private keys (K_i, k_i), and stores the public keys of the other network members, particularly those of the other servers. This configuration allows server nodes to establish a secure link between them. [YI 02] proposes to distribute the trust to nodes having a better physical security and computation power, especially within a heterogenous environment composed of nodes having different characteristics. These nodes are called MOCAs (Mobile Certificate Authorities). In the case of our configuration (n, t+1), the n servers share the ability to sign the certificates of the other nodes. The private key k of all the certification service is divided into n shared secrets (s1, s2… sn), one secret being known by only one server. Figure 14.1 illustrates this configuration.

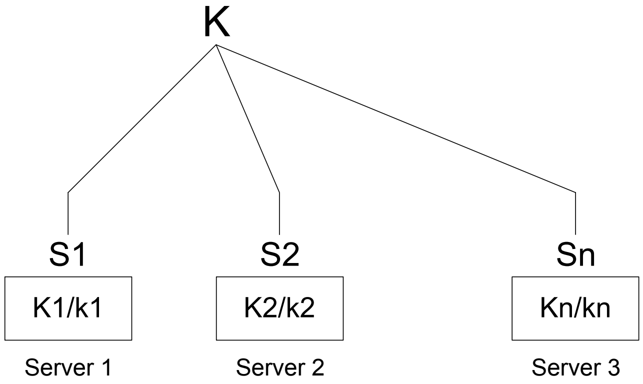


Figure 14.1. *Key management service configuration*

Each server generates a partial signature of a node's certificate and sends it to a combiner, which needs at least t+1 partial signatures to generate the complete signature. The maximum number of compromised servers at any period of time must be equal to t: with t compromised servers, the combiner is still able to generate a valid signature. Zhou *et al.* make the assumption that (n >=3t+1) [ZHO 99].

The combiner is also able to verify the validity of a partial signature (PS) sent by a server. If a PS is revealed to be erroneous, the combiner rejects it and continues collecting t+1 valid PSs. Figure 14.2 illustrates this operation of signature construction, having a (3,2) configuration in which server 2 was compromised. There, the combiner was able to generate the signature of the certificate of the node m ($Cert_m$).

The choice issue of the parameter t is detailed in [YI 02]. The higher the parameter *t*, the higher the security level against eventual malicious attacks. A high value of *t* increases the communication overhead.

The combiner, which is mandatory for the generation of node certificate signatures, can itself be compromised and consequently become a vulnerability breach of the whole network security system. [LEG 03] proposes a duplication of the combiner into several CAs: we thus obtain a cooperative architecture where local combiners can be formed around the concerned node, in order to generate its signature.

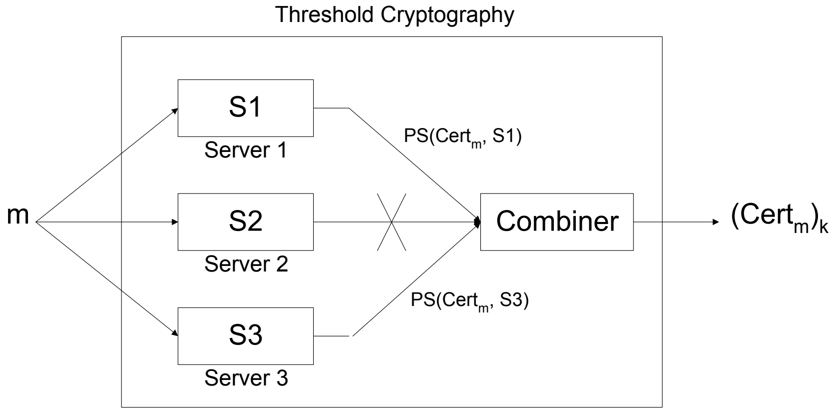


Figure 14.2. *Threshold cryptography technique with a (3,2) configuration*

Yi *et al.* present a certification protocol called MP (*MOCA Certification Protocol*) [YI 02]. According to this protocol, clients broadcast *Send Request* (SREQ) messages; each MOCA receiving this message sends a *Certif Response* (CREP) message (similarly to the AODV routing protocol), containing a partial signature. When the client node collects *t* valid CREPs, it can compute its signature. This protocol does not need a combiner, thus offering a better security level. To solve the problem of SREQ flooding (all the MPCAs receive one SREQ and send CREP messages, whereas the node needs only *t* answers), Yi *et al.* propose the B-Unicast technique. This solution allows a node to send requests by unicast to exactly *t* MOCAs if their routes are already in the routing table. Otherwise, the node has to use the more constraining solution of complete network flooding.

14.2.2. *Self-managed PKI*

Hubaux et al. propose in [HUB 01] a self-managed PKI, dedicated to operating within ad hoc networks, where each node establishes certificates for nodes it trusts. If two entities want to communicate securely, without knowing each other, they exchange their certificates lists and try to create a trust chain between them. For example, when two nodes A and B want to communicate together and they trust node C, a trust chain between A and B can be created through node C (as for the PGP protocol, which stipulates that "the friends of my friend are my friends").

Local database construction mechanisms are used in [HUB 01] to contain the node certificates, so that any pair of nodes in the network can establish a trust chain between them, with a high probability, even if the size of the local databases is small compared to the number of nodes in the ad hoc network. The relational trust model between users is represented by a graph G(V,E). V and E represent the set of vertex (users) and the set of edges (certificates) of the graph respectively. Thus, the existence of an edge between two vertices *u* and *v* in the trust graph means that node *u* generated a certificate for node *v*. The existence of a trust chain between two nodes of the MANET is thus represented by a direct route between the two vertices of the graph, representing the two concerned nodes. Figure 14.3 illustrates this process of trust chain establishment between two nodes *u* and *v*.

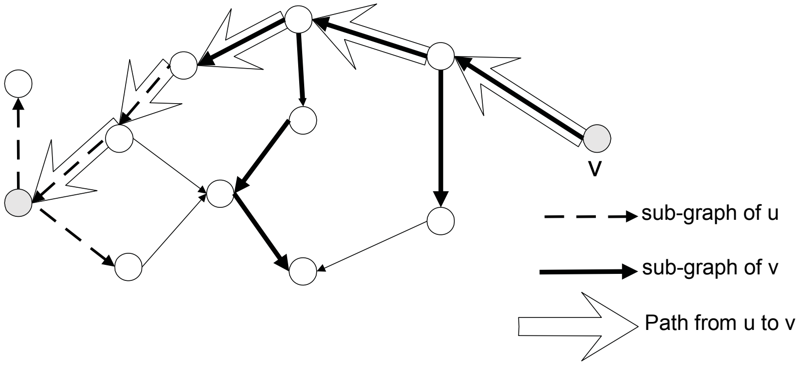


Figure 14.3. *Trust graph in [HUB 01]*

This distributed authentication technique has probabilistic guarantees, due to the fact that the existence of a trust chain between two nodes in the graph is not ensured. In addition, the distributed storage of node certificates generates a high overhead, which make the real applicability of this approach difficult on a large scale. Moreover, malicious members can generate erroneous certificates and integrate them into the trust graph. To solve this problem, Hubaux *et al.* propose the use of authentication metrics, allowing the evaluation of the authenticity of certificates and the trust chains they belong to. The number of disjoined certificates between two nodes in the trust graph is an example of an authentication metric in [HUB 01]. It is important to note that PGP-based approaches are especially suitable for small communities, because the certificate and key authenticity can be ensured, with a higher trust level.

The approach proposed by Luo *et al.* in [LUO 00] is also based on the PGP principle and consists of generating the certificate of a node by its neighbors in a cooperative manner and according to its behavior. The certification services, such as generation, renewal and revocation, are shared by all the network members. Thus, as for the threshold cryptography technique, the private key of the certification authority is shared by a defined number of the network nodes. These nodes are responsible for the generation of certificates for the "honest" nodes, and thus for the development of the trust graph of the network. Neighboring nodes, having established trust relationships, cooperate with forward packets and detect eventual malicious attacks. Note that nodes without their certificates should be considered as potential intruders.

14.2.3. *Key agreement technique within MANETs*

The context of this approach is a small group of people, participating in a conference within a room for an ad hoc meeting following an asymmetric encryption model; these people want to exchange confidential data during the meeting.

The principle of the key agreement protocol, assuming that all members trust each other, consists of sharing a weak password, from which another password will be generated and will constitute the session encryption key of the group. This protocol presented in [ASO 00] must have the following properties:

– secret: only nodes knowing the weak password should be able to deduce the session key;

– contributing agreement: the generated session key should be composed of the contributions of the participants of the secure communications session;

– tolerance to attacks: attacks taken into account are those consisting of injecting erroneous messages in the network, but not attacks which modify or delete messages sent by other nodes.

The authors of [ASO 00] present the EKE (Encrypted Key Exchange) authentication protocol; the participating entities of EKE are two nodes A and B within an ad hoc network, holding a common weak secret p. The two nodes generate a traffic encryption key K starting from the secret p, so that an intruder cannot attack the weak secret used in the first exchange (dictionary attack) or access the encryption key K. In the same proposal, the authors propose to extend the EKE protocol, so that it becomes a multi-user protocol. The only constraint is that one leader should trigger the authentication operations and the message exchanges. In addition, this protocol does not satisfy the contributing agreement property, because the leader computes the session key and distributes it to the other nodes. Asokan *et al.* [ASO 00] enhance the EKE protocol in order to obtain a multi-user protocol, allowing all the participating nodes to contribute to the session key generation process. However, this modification is very constraining because the leader should wait for all the contributions generated by the other nodes in order to compute the final session key.

The Diffie-Hellman key exchange protocol can carry out the authentication via a weak password. This protocol allows us to solve all the problems described previously. It provides a secret shared between the different participants to the secure session. Moreover, it enhances the fault-tolerance. [ASO 00] presents an enhancement of the Diffie-Hellman protocol concerning the number of communicated messages, while arranging the participant nodes on a hypercube. The basic idea of this protocol is illustrated in Figure 4.4; with four participants A, B, C and D, trying to agree on a shared secret encryption key.

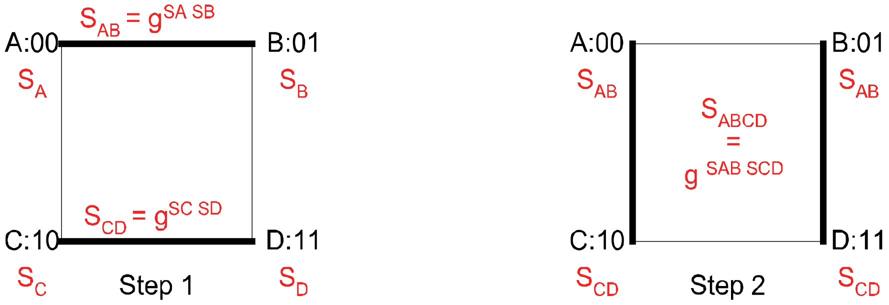


Figure 14.4. *Diffie-Hellman exchange within a 2-cube*

Each participant *i* holds a two-bit address and generates a contribution S_i. At the first step, nodes A and B execute the Diffie-Hellman key exchange for two participants, they compute thus $S_{AB}=g^{S_AS_B}$. At the same time, C and D compute $S_{CD}=g^{S_CS_D}$. The second step consists of executing the Diffie-Hellman algorithm between A and C, and B and D, while using as contributions the computed keys deduced from the first step. Thus, at the end of the second step, the four participants hold the same session key $S_{ABCD}=g^{S_{AB}S_{CD}}$.

If the number of participants is evaluated as equal to $n=2^d$ participants, each participant is attributed a vertex in a hyper-cube of d-dimension. The protocol proceeds, during d steps of key exchanges, following the same principle presented above. After d steps, all the participants will hold the same secure session key.

All the protocols presented so far solve the authentication problem within ad hoc environments, without the need for any additional infrastructure or secure physical communication channels. This matches the initial requirement of any MANET security infrastructure.

14.2.4. *Cryptographic identifiers*

Cryptographic identifiers [MON 02] are generated and held by the nodes of ad hoc networks, in order to prove their identities to nodes communicating with them, without the need of any trust administration. These identifiers are statistically unique and cryptographically verifiable, which means that it is very difficult that two entities hold the same identifier, and that it is possible to check the validity of an identifier by an entity, thanks to cryptographic techniques.

The cryptographic identifier, called CBID, is defined as:

CBID = hmac_sha1_128(sha1(imprint),sha1(PK))

where:

– PK is the public key of the identifier generator;

– imprint is a random value of 64 bits;

– hmac and sha1 are two hash functions.

The basic idea of the crypto-based identifiers is to establish a strong cryptographic relation between their components (private and public keys). A node announces its identity to the other nodes, by proving that it holds the private key associated with its public key, which is used for its CBID generation. For example, to prove its identity, a node A sends the following message to a node B:

A → B: Public_key$_A$, imprint, {CBID$_A$}Private_key$_A$

This message contains the public key of node A, the imprint value used for the generation of its CBID and the CBID encrypted with the private key of the node A. To affirm the authenticity of node A, node B computes A's CBID, using its public key and the imprint value. Then, node B decrypts A's CBID, using A's public key. The authentication process succeeds if the two computed CBIDs are equal.

CBID-based authentication does not require a centralized administration, such as a PKI or a key distribution server. So, the authentication of a new node is not possible. Only members knowing each other beforehand can identify and authenticate themselves, and consequently communicate securely.

14.2.5. *The Resurrecting Duckling technique*

This technique [FRA 99] is based on a metaphor inspired by biology, describing the behavior of a duckling emerging from its egg, and recognizing as its mother the first mobile object which emits a sound. This phenomenon is called "imprinting".

Similarly, an entity recognizes as its owner (its controller) the first entity which sends it a secret key (during the communication session). The sending of the secret key between equipment and its owner is carried out directly (via an electrical contact), thus avoiding any cryptographic operation or ambiguities concerning the identities of the intervening entities. However, at the same time, this kind of authentication makes the Resurrecting Duckling technique restricted to a specific kind of applications and not suitable for a large deployment of ad hoc networks.

The equipment controller sends it, in a secure manner, any information to determine its behavior with the other nodes of the network (security policies, access control list, etc.). The equipment can thus communicate with the other entities of the network, but cannot be controlled by them. The targeted application, detailed in [FRA 99], is a medical application on which equipment is for example a thermometer held by the patients, and the controllers are the PDAs of the doctors.

14.2.6. *Summary*

The establishment of secure communications within an ad hoc network is a challenging problem. An ad hoc network is a hostile environment, bringing several security challenges, due to its characteristics and specificities (wireless links, low capacities, etc.). In this context, we studied the various authentication approaches in these networks.

The deployment of group communications within an ad hoc network induces additional challenges towards the design of a group key management approach. Indeed, in addition to the security constraints of the ad hoc networks, the multicast IP model brings new security vulnerabilities, by eliminating any possibility of group member's identification or data confidentiality.

In the next section, we study the characteristics of the multicast communications within MANETs, and we present and discuss a state of the art concerning the group key management protocols within these networks.

14.3. Group key management within ad hoc networks

Multicast transmission is an efficient and suitable mechanism for group-oriented applications such as audio-video conferences. The IP multicast model defined by Deering [DEE 91] is an extension of the IP model. It defines the notions of group, addressing scheme and group adhesion protocol. The group is itself dynamic; one entity can join or leave the group at any time (see Figure 14.5). A multicast group is open, so an entity can send packets to a multicast group without belonging to it.

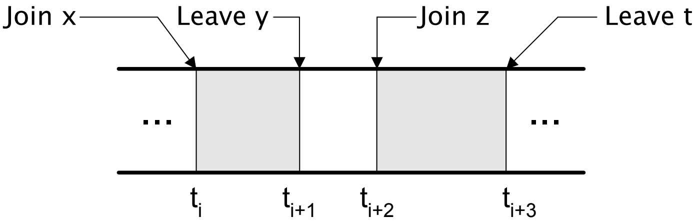


Figure 14.5. *Evolution of a group session*

Multicast group addresses form a sub-set of IP addresses (class D in IPv4 and prefix FF00::/8 in IPv6). Some multicast groups are permanent with fixed and known addresses. Other groups are temporary and thus hold dynamically allocated addresses. The group adhesion protocol IGMP (Internet Group Management Protocol) [DEE 91] operates between nodes and their multicast routers. It allows a node to inform its multicast router that it wants to receive the flow for a given multicast group. Thus, the router periodically queries its local network to detect nodes still belonging to multicast groups. Based on the IGMP, a multicast router is able to define which multicast traffic should be sent to its local network. Multicast routers use this IGMP information, associated with the multicast routing protocols (e.g. MOSPF [MOY 94], PIM [DEE 94] within wired networks, and MOLSR [LAO 03], MAODV [ROY 00] within ad hoc environments). Figure 14.6 shows the basic components of the multicast IP model.

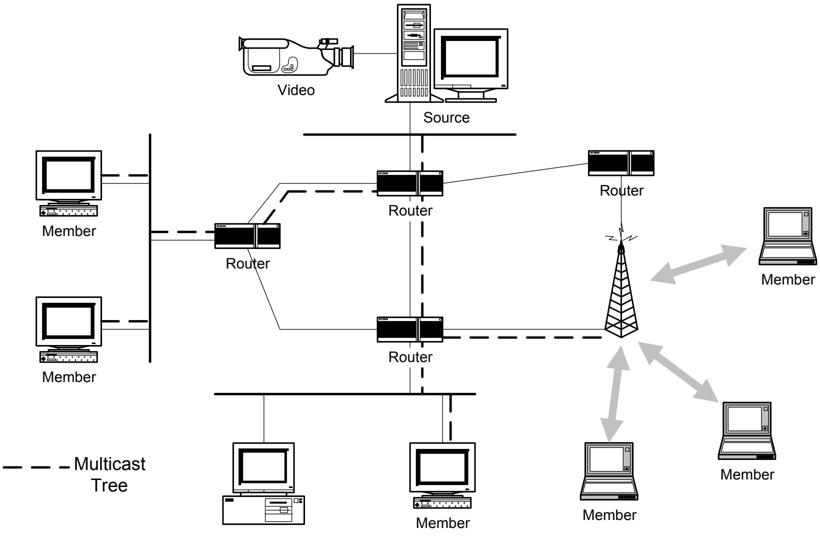


Figure 14.6. *The IP multicast model*

The lack of security within the multicast communication model is one of the factors which has limited its deployment within large-scale networks, particularly concerning business-oriented applications. This limitation is a major motivation for many research initiatives whose goal is to establish a secure architecture of group communications and avoid any malicious attack.

In this section, after describing the security services required for group communications and challenges to be considered, we compare the main key management protocols within ad hoc networks.

14.3.1. *Security services for group communications*

Security services are related to the multicast data sent by the source and to the identities of the group participants. We distinguish five main properties:

(1) Data confidentiality. This property ensures that only authorized members can access the multicast flow sent by the source. To enforce this property, a symmetric key is used by the source to encrypt data, and by the receivers to decrypt them. This key is called the Traffic Encryption Key (TEK).

(2) Forward and backward secrecies. A member having left the multicast group should no longer be able to decrypt the multicast flow sent after its departure (*Forward Secrecy*). Similarly, an entity joining a multicast group should not be able to decrypt the multicast flow sent before its group attendance (*Backward Secrecy*). It is thus mandatory to trigger a TEK renewal process after each addition or withdrawal of an entity in the multicast group. A new traffic encryption key is thus renewed and distributed to all the multicast group members (with the new member in the case of entity addition, or only the remaining members in the case of entity withdrawal). The distribution of the TEK is secured with Key Encryption Keys (KEK). Note that the forward and backward secrecies are applied according to the security policies adopted by the application: the source of the group is responsible for triggering group key renewal and activating the redistribution processes, depending on the required security level and the confidentiality of the sent data. The renewal of the traffic encryption key involves the "1 affects n" phenomenon (all the group members are affected by the renewal of a key, triggered after a join or leaving of a single member), and, in the case of entity withdrawal, the "1 does not equal n" phenomenon (the remaining members are considered individually and received unicast messages).

(3) Access control of the group members. This security service guarantees that the adhesion to the multicast group is ensured via an ACL (*Access Control List*), containing all the entities authorized to join the group.

(4) Source authentication. This security property ensures that the group members authenticate the identity of the group source for every received multicast flow. This service essentially guarantees the non-repudiation of the source.

(5) Group authentication. This security property requires the group members to check that the source of transmitted data belongs to the multicast group.

The IP multicast model is attractive, efficient and suitable for large-scale networks. However, these advantages present some vulnerabilities that security services should face to ensure secure group communications. Indeed, the simplicity and the efficiency of the IP multicast model are due to the fact that no identification of the group participants is done. Multicast group addresses are publicly known; any entity in the network can thus join the multicast group, access to the multicast flow, without any authorization or invitation. A malicious entity can also send multicast data to the group members, without belonging to their group and without authorization or access control. Such actions can cause DoS attacks and consequently affects the confidentiality and the availability of the transmitted data. Moreover, the multicast data flows are forwarded within the network via several routes, constructing the multicast group tree. This feature increases the opportunities of malicious attacks such as network sniffing.

14.3.2. *Security challenges of group communications within MANETs*

The characteristics of ad hoc networks, the security level to establish and the types of the multicast applications to secure require several constraints and challenges to be taken into account:

– the use of wireless links eases passive attacks (such as network sniffing) and active attacks (such as message alterations);

– the lack of a fixed infrastructure is one of the main characteristics of an ad hoc network. This characteristic eliminates any possibility of establishing a centralized reference that is responsible for the management of the different security services. The lack of a fixed infrastructure thus implies the inapplicability of a centralized security model, such as the one used for the PKI, which is hardly applicable within these environments;

– the size and dynamics of the multicast group can be very high within ad hoc networks. Indeed, we cannot control the number of group members or the adhesion frequency to the group. The security mechanisms should face these parameters and thus be adapted to the dynamics and scalability of MANETs;

– the mobility of ad hoc networks should be considered in the design of secure group communication architectures within these networks. When a node is moving in the network, it can lose its connectivity to its group without leaving it. Thus, it should not be obliged to re-authenticate itself every time it moves away from its multicast group. Moreover, the re-authentication mechanism should be efficient and fast, requiring a minimum of transmitted messages;

– a group key management protocol within MANETs should also consider the security requirements of multicast applications. According to the application type, different security requirements may emerge. For example, a free software distribution application follows the 1 to n multicast model. Transmitted flows are publicly available, and consequently the authentication of the source is more important than the confidentiality of the sent data. A second example is a pay service like a TV channel. Within this kind of applications, the authentication of the group members is mandatory to ensure proper access control and accounting.

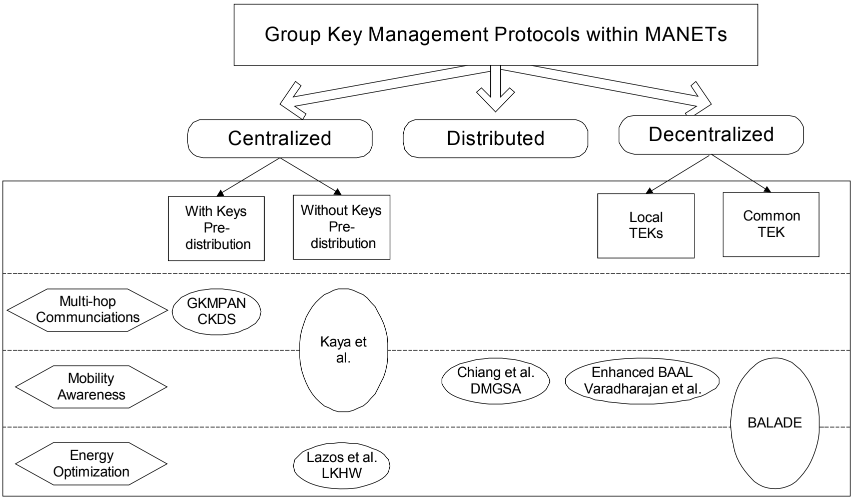


Figure 14.7. *Taxonomy of group key management protocols within MANETs*

In what follows, we present a taxonomy of group key management protocols dedicated to operate within MANETs [BOU 08] (presented in Figure 14.7). This taxonomy extends and enhances the classical taxonomy used for wired networks while integrating the characteristics and specificities of ad hoc networks (mobility support, energy optimization and multi-hop awareness). We also evaluate and discuss the presented protocols, according to a set of metrics presented in the next section.

14.3.3. *Comparison metrics*

In order to compare group key management protocols in ad hoc networks, we define the following comparison metrics: constraints and pre-requisites of the protocols, their real applicability, the supported security services (authentication, confidentiality and integrity of data, revocation of malicious nodes, etc.), scalability in terms of computation, storage and communication overheads, and finally the vulnerabilities and efficiency against bottlenecks.

14.3.4. *Centralized approach*

Within this approach, group key management is centralized around a unique entity in the network. We divide this approach into two families: with and without a key pre-distribution phase.

14.3.4.1. *Protocols with a key pre-distribution phase*

These protocols configure entities by pre-distributing a set of keys for each node off-line (before the deployment of the multicast session). These keys allow a node to decrypt the multicast flow sent by the source or to obtain the traffic encryption key sent by the source when the key renewal process will be triggered. Key pre-distribution is used within the GKMPAN [ZHU 04] and CKDS [MOH 04] protocols, because of the lack of fixed infrastructure within MANETs.

14.3.4.1.1. The GKMPAN protocol

GKMPAN [ZHU 04] is based on a phase of key pre-distribution to all the group members. It also has several key renewal phases under the responsibility of a key server.

During the key pre-distribution phase, each group member *u* obtains, off-line, before the bootstrap of the multicast session, the following keys:

– A set R_u composed of *m* keys among *l*, *l* being the total number of keys $\{k_1, k_2, ..., k_l\}$. I_u is the set of the key identifiers corresponding to the set R_u. The keys of R_u are used as KEKs. The key pre-distribution algorithm allows each node *i,* knowing the identity of a node *j*, to define the set of keys I_j and thus to determine which key to use to communicate securely with the node *j*.

– The initial group key k_g, used for securing the communications between the group members.

– A secret key, shared between the key server and each group member individually.

– The authentication of the data source is ensured via the TESLA protocol [PER 02, HAR 03] (presented below). The TESLA authentication requires the pre-distribution of a first key, called the commitment key. This key is thus pre-deployed at each group member.

New members can join the multicast group within GKMPAN, even after the key pre-distribution phase. The key server could, for example, add members in the group to compensate excluded members. To add a member u to the multicast group, the key server deploys its set R_u in addition to the current group key. Following this event, and according to the application, the key server decides whether or not to renew the group key k_g to ensure the backward secrecy, and thus to send a group key renewal message $k'_g=f_{k_g}(0)$, f being a pseudo-random function.

Distribution of the group key: the group key distribution process is initiated by the key server, which generates a new group key. It then distributes it in a hop by hop manner, encrypted using the pre-deployed KEKs. The key server delivers the group key to its immediate neighbors at one hop, which forward it to their neighbors in a recursive and secure manner. GKMPAN thus exploits the multi-hop communication property of ad hoc networks.

Group member revocation: when a malicious member is excluded, the key server broadcasts a revocation notification in the network, containing the identifier of the excluded member, the identifier of the non-compromised KEK *i*, known by the large number of group members, and the new group key encrypted with the chosen key *i.* Members not holding the KEK *i* used for the encryption of the group key will receive this key forwarded by their neighbors, encrypted with other non-compromised KEKs. The notification message is authenticated using the loss-tolerant TESLA protocol [PER 02, HAR 03].

Message authentication with TESLA: for this service, the key server and the group members are synchronized; each node knows an upper limit of the synchronization time with the server, noted Δ_t. Time is divided into intervals of T_{int} duration. To each interval I_j corresponds an authentication key k'_j. The source generates a chain of keys $k_1 \ldots k_t$ using a one way function f. In order to do this, the last key k_t is generated randomly, and the other keys are generated via the following function: $k_{j-1} = f(k_j)$. Then, the source generates authentication MAC (*Message Authentication Code*) keys such that $k'_j = g(k_j)$, g being another one-way function. Figure 14.8 illustrates key chains in TESLA.

The data source authenticates each packet P_i with the key of the current time interval j, and includes authentication information with the sent data MAC(K'_j, P_i). The source also includes the k_{j-d} key used to authenticate packets sent before d time intervals, d being the disclosure delay of TESLA.

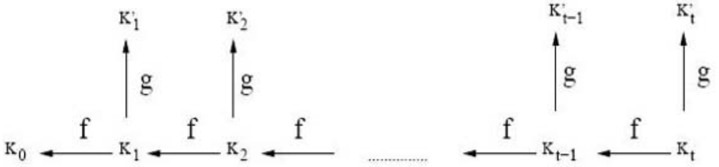


Figure 14.8. *MAC key chains in TESLA*

The receiver group members verify the authenticity of messages sent by the source by verifying that the revealed key (after *d* intervals) matches the result of the one-way function f: $k_0 = f^j (k_j)$.

Renewal of the compromised keys: the KEKs held by an excluded member are compromised and should be renewed by the other members holding these keys, in the following way:

– *M* is the identifier of the non-compromised key, known by the large number of group members;

– the key server generates an intermediary key $k_{im} = f_{kM}(k_g)$, where k_g is the group key and k_M is the key of identifier M;

– the k_i keys held by the excluded member *u* (R_u) are renewed by the k'_i keys as follows: $k'_i=f_{kim}(f_{ki}(0))$.

14.3.4.1.2. The CKDS protocol

CKDS (*Combinatorial Key Distribution Scheme*) [MOH 04] is an applicative-layer group key management protocol within MANETs. The key distribution in CKDS is based on the combinatory based system EBS (*Exclusion Basis System*) [MOR 03], associated with the CAN (*Content Addressable Network*) [RAT 01].

During the key pre-distribution phase, each node in CKDS holds *k* keys (known keys) and does not know *m* keys (unknown keys). Figure 14.9 shows an example of an EBS matrix, with 10 members U1 to U10, *k=3* and *m=2*. A case (i,j) is equal to 1 if the member Uj knows the Ki key. This example is presented in [MOH 04].

CAN is a distributed hash table used to carry out repartition of all the group members in a m-dimensional space. Thus, each node in a quadrant of the space is localized according to its unknown keys in the EBS system.

	U1	**U2**	**U3**	**U4**	**U5**	**U6**	**U7**	**U8**	**U9**	**U10**
K1	1	1	1	1	1	1	0	0	0	0
K2	1	1	1	0	0	0	1	1	1	0
K3	1	0	0	1	1	0	1	1	0	1
K4	0	1	0	1	0	1	1	0	1	1
K5	0	0	1	0	1	1	0	1	1	1

Figure 14.9. *EBS matrix in CKDS (10 nodes U1 to U10 and 5 keys K1 to K5)*

To distribute and renew keys, a centralized entity, called a global controller, is assumed to be available in the network and is responsible for the generation of the group key and the construction of the key renewal messages. The key messages distribution task is delegated to group members, which perform it using two possible methods.

The first method of key distribution is called "*m*-dimensional multicast". When a member is excluded, keys held by this member are compromised and should be renewed. The key renewal process is triggered by a diagonal node in the partitioned space (the node that holds all the unknown keys of the excluded member). This node is called the IGD (*Initial Global Distributor*). The IGD receives from the global controller key renewal messages to forward to the other non-compromised group members. In Figure 14.9, if node U1 is compromised, U6, U9 and U10 can perform the key renewal process because they know the unknown keys of U1 (K4 and K5). The selected IGD starts by localizing the central members in each quadrant of the m-dimensional space. These central nodes are called the LQD (*Local Quadrant Distributor*). Then, the IGD sends the suitable key renewal message, via a direct flooding technique. The LQDs forward, in a multicast manner, the received messages to their local members. Thus, as in GKMPAN, CKDS exploits the multi-hop communications property of the ad hoc networks.

The second key distribution method is called "2D-multicast" and, also based on initial and local distributors (IGD and LQDs), aims to decrease the overhead due to communication and encryption of the first scheme, presented above. Indeed, within the *m*-dimensional scheme, key renewal messages can reach members who need only the renewed keys and not all the distributed keys. Moreover, at the sending of the key renewal, the IGD and the LQDs must carry out re-encryption operations. Final group members should thus achieve two constraining decryption operations. The 2D-multicast scheme thus proposes to target the key renewal only to the interested members. The adopted solution thus consists of sending only one renewed key within a key renewal message. In addition, to avoid the double decrypting operations, the renewed keys are encrypted with a new KEK, established via the compromised key and another key Ki, not held by the malicious excluded member, due to a hashing function. A renewal message for a key Kj to K'j, called Rij, has the following form: Rij = Ki|Kj (K'j), with Ki|Kj being the encryption key generated via Ki and Kj.

14.3.4.2. *Protocols without the key pre-distribution phase*

This family of protocols does not need a key pre-distribution phase. Three protocols presented hereafter belong to this approach: Kaya *et al.* [KAY 03], Lazos *et al.* [LAZ 03] and LKHW [PIE 03].

14.3.4.2.1. The Kaya *et al.* protocol

Kaya *et al.* [KAY 03] propose a group key management protocol within MANETs, taking both node mobility and the multi-hop nature of ad hoc communications into account. Members join the group via the nearest neighbor, already belonging to the multicast group, using GPS information. Join requests are distributed, in anycast (only the nearest neighbor answers this request), with a limited range (TTL field), to reach the first member of the group. Consequently, in addition to the communication overhead optimization, this method allows the establishment of the multicast tree with the shortest paths, facilitating and optimizing the key distribution process.

A certification service is provided by this protocol to ensure the access control of members and the revocation of malicious nodes. Only nodes holding valid certificates are able to access the multicast flow. A node wanting to join the group should obtain a valid certificate, off-line, encrypted with a trusted certification authority (TTP: *Trusted Third Party*).

If the authentication of a new member by a group participant succeeds, the two entities generate and share a secret key. Then, the access control of the new member is verified according to its certificate. In case of successful access control, this member can access to the multicast flow sent by the source encrypted with the secret key obtained at its authentication. Excluded nodes, with revoked certificates, should not be able access to the multicast flow. To do this, the source sends periodically, in multicast, a message containing the list of all the revoked certificates. The group members store this list and use it to authenticate and control access control of new potential members.

14.3.4.2.2. The Lazos *et al.* protocol

The proposal of Lazos *et al.* [LAZ 03] adopts the centralized key management architecture, taking into account the energy constraint within ad hoc networks. It enhances the LKH (*Logical Key Hierarchy*) distribution [WON 98] and adapts it to the context of static ad hoc networks, by optimizing the energy consumption via the use of the geographical localization of group members (obtained with GPS).

A multicast group is defined in LKH by a triplet (U, K, R), corresponding to an oriented and acyclic graph (key distribution tree). U defines the set of members of the group, K is composed of the set of group keys and R defines the relations between U and K (set of keys held by each member). The root of the LKH tree corresponds to the group key, while leaves correspond to the group members. The intermediary nodes are constituted by logical keys.

A member knows all the keys of its path to the tree root. After a join or leave event of an entity, a key renewal process is triggered and consists of renewing all the keys from the joining or leaving node respectively to the root of the tree (group key). Several key distribution processes can be used (user-oriented, key-oriented or group-oriented), but all suffer from the "1 affects n" phenomenon.

The basic idea of the protocol of Lazos *et al.* is that geographically close members can potentially be reached by one broadcast message or use the same path to access the multicast flow. The ad hoc network is represented by a two-dimensional space, and the K-means clustering algorithm [MAC 67] is used to form sub-groups (called clusters) of high correlation and then establish the key distribution tree.

The key distribution process, based on the K-means algorithm, is composed of several steps. First, the group members are allocated to one cluster. Then, each cluster is divided into two sub-clusters via the K-means algorithm. A refinement procedure is used to balance the number of members per cluster. These steps are iterated, until clusters are formed by one or two members. Clusters formed by only one member are merged when possible. The final step of the process consists of mapping the cluster hierarchy to a logical hierarchy of LKH key distribution. Figure 14.10 illustrates an execution of this algorithm. In this example, members M4 and M6 are geographically close and consequently they are "brothers" in the LKH key distribution tree.

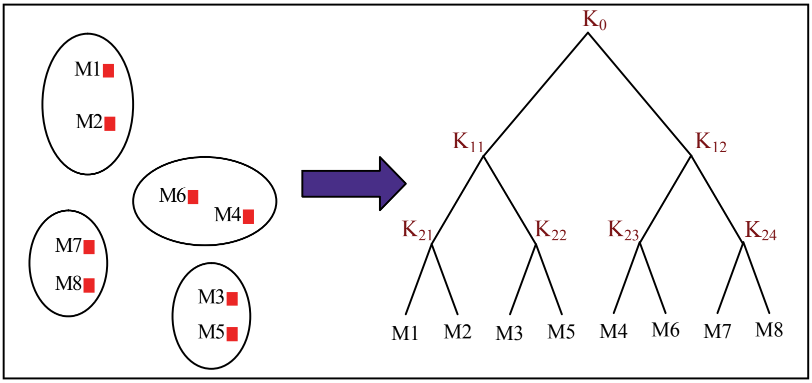


Figure 14.10. *Key distribution process based on the K-means algorithm*

14.3.4.2.3. The LKHW protocol

LKHW [PIE 03] is a secure multicast communication protocol, based on the LKH key distribution protocol [WON 98] associated with the direct diffusion technique. LKHW is dedicated to operating within wireless sensor networks (WSNs). LKHW actors are the source of the group and the sensors. The sensors can provide data required by the source, which is responsible for their collection. Sensors have low physical capacities, in terms of both communication and computation. The key distribution process is based on LKH, and the key renewal uses the direct diffusion technique, optimizing energy consumption. The security services ensured by LKHW are confidentiality, integrity and data authentication. Both backward and forward secrecies are ensured in LKHW. The main phases of LKHW are group initialization and key renewal processes triggered after each join or leave event.

At group initialization, the establishment of the secure communications starts when the source builds the logical hierarchy of keys. Initially, the source sends an exploratory message to all group members to find nodes able to provide the data it needs. The interested members answer this message by declaring tasks they can accomplish. The source then collects these answers and sends its identifier to each participating sensor, and the set of keys corresponding to its localization within the LKH tree. At this step, the secure group communications can start.

The key renewal processes are triggered at each join or leave event. When a member would like to join the group, the source starts by sending to it the set of keys corresponding to its localization in the LKH tree. In addition, all group members should update their key sets to guarantee the backward secrecy. Similarly, when a node leaves the group, the LKH keys from its position to the tree root are updated to guarantee the forward secrecy. The direct distribution technique used in LKHW is optimized thanks to the use of caches, the removal of duplicated messages and the prevention against cycles.

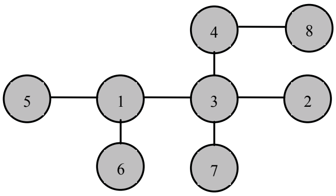
14.3.5. *Distributed approach*

Group key management within a distributed approach is under the responsibility of all group members, which cooperate to share a secret group key. Protocols belonging to this approach are those defined by Chiang *et al.* [CHI 03] and DMGSA [KON 06], both presented hereafter.

14.3.5.1. *The Chiang et al. protocol*

Chiang *et al.* propose a distributed group key management protocol within MANETs [CHI 03], based on the GPS measures (latitude, longitude and altitude) associated with the GDH (*Group Diffie Hellman*) key exchange protocol [ING 82]. At protocol initialization, each ad hoc node generates its public key K_{pubA} as follows: $K_{pubA} = \alpha^{a} \mod p$, with α an integer, p a large prime number (α and p are known by all the participants of the multicast group) and a a random private integer. Then, each node distributes its GPS localization and its public key to all the group entities.

Due to the exchanged information, each group node knows the topology of the entire network. When a source aims to send multicast data to all the group members, it builds the minimal multicast tree, using the Prüfer algorithm [PRU 18]. This algorithm computes a Prüfer number, suitable to code a multicast tree, basing itself on the degrees[3] of the group members. Indeed, a node of d degree appears exactly d-1 times in the Prüfer number. Figure 14.11 gives an example of a multicast tree and its Prüfer number. The degree of the nodes 3 is four; it thus appears three times within the Prüfer number.



Prüfer number	3	3	3	4	1	1

Figure 14.11. *A multicast tree and its Prüfer number*

[3] The degree of a member within a multicast tree is equal to the number of its links within the multicast tree.

The group key is generated by all the group members, via the GHD key exchange protocol, and is built via a combination of their public key. The principle of the GDH protocol is to extend the key DH agreement protocol to the context of group communications; with *n* participants M_1, M_2 … M_n. *n* steps are necessary to generate the group key. The first n-1 steps correspond to the collection of the contributions of group members, carried out by the last node M_n. At the last step, M_n issues the intermediary values to the group members, allowing them to compute the group key. Figure 14.12 illustrates four members 1, 2, 3 and 4. The generation of the group key is carried out as follows:

Step 1 - 1 → 2 : α^{r1} mod p

Step 2 - 2 → 3 : α^{r1}, α^{r2}, $\alpha^{r1\,r2}$ mod p

Step 3 - 3 → 4 : $\alpha^{r1\,r2}$, $\alpha^{r1\,r3}$, $\alpha^{r2\,r3}$, $\alpha^{r1\,r2\,r3}$ mod p

Step 4 - 4 → All : $\alpha^{r1\,r2\,r4}$, $\alpha^{r1\,r3\,r4}$, $\alpha^{r2\,r3\,r4}$ mod p

The source of the group then sends the Prüfer sequence to all the group members, in multicast, encrypted with the group key. After receiving this Prüfer sequence, each member will decode the multicast tree built by the source and will know if it must or must not forward packets to other group members. A secured group is thus represented by a key graph, composed of two types of node, leaves representing group members (*U*), and intermediary nodes representing their public keys (*K*). The root of the tree, called k_p, indicates the Prüfer key (*P*). The secure multicast group is noted (*U, K, P*).

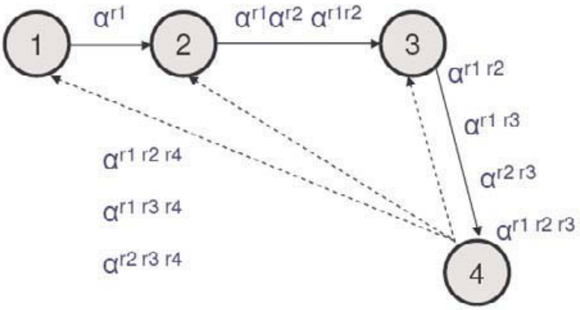


Figure 14.12. *Group key generation within GDH (with 4 participants)*

The key distribution graph can be extended to ensure secure communications between several multicast groups. The key of the merged groups can be built, in a hierarchical manner, starting from the initial group keys.

14.3.5.2. *The DMGSA protocol*

DMGSA (*Distributed Multicast Group Security Architecture*) [KON 06] is distributed and clusterized multicast security architecture. It takes into account mobility and density of nodes at the creation of clusters. The group key management is carried out through specific entities in the network, called GCKSs (*Group Control Key Servers*), acting as cluster heads, and together forming the backbone of the multicast group. Within each *k*-hop neighborhood, a GCKS is elected at each change or modification of the topology. The GCKS election is carried out in a distributed manner, following two steps: a phase of clusters formation and a phase of clusters maintenance.

The distributed phase of cluster formation is initiated by a node which does not belong to a cluster yet. This node issues the election messages, claiming itself as a cluster head (GCKS). The distribution of these messages is carried out in broadcast within the *k*-hop neighborhood (the TTL field of the packet is positioned to *k*). The choice of *k* is based on an estimation of the local density of the initiator node within its neighborhood. This estimation is computed using a neighbor's detection algorithm. In the case of concurrency between two entities, the node holding the smaller value of *k* and the smaller identifier is elected as the GCKS.

During the phase of cluster maintenance, each cluster head periodically sends a message to claim itself as the GCKS within *k*-hops, thus keeping in its cluster members which it receives. When a member does not receive a periodic message sent by its cluster head, during a defined period of time, it joins another cluster.

The key management within DMGSA consists of sharing a group TEK, managed by the group of GCKSs. Each group member receives the TEK sent by its GCKS (the nearest to its geographical location at maximum *k*-hops). In order to distribute the TEK in a secure manner to its members, each GCKS authenticates its local members when they join the group and controls their access to the group through their pre-deployed certificates. In the event of success, the GCKS establishes with each local member of its cluster a secret key, called KEK, that it will use to encrypt the TEK of the multicast group.

The TEK renewal is triggered when the join and leave events frequency exceeds a defined threshold. In this case, the GCKS generates a new TEK, sends it to its local members encrypted with their respective KEKs, and also forwards it to the other GCKSs. The encryption issue of the exchanged messages between the different cluster heads (GCKSs) is not considered in [KON 06].

14.3.6. *Decentralized approach*

The decentralized approach divides the multicast group into sub-groups or clusters. Each cluster is managed separately by a local controller responsible for the management and the security of members of its sub-group. Two families of protocols can be distinguished in this decentralized approach.

The first family of decentralized protocols uses a local traffic encryption key for each cluster. We call this protocol family *local TEK protocols*. Local controllers generate and distribute the local TEKs to their local members. Upon receiving the multicast flow sent by the source, the local controller decrypts it with the appropriate key, re-encrypts it with the local keys corresponding to their clusters, and forwards it to their local members. The advantage of this approach is that it ensures forward and backward secrecies, while attenuating the "1 affects n" phenomenon. The renewal of a local key of a cluster, triggered after an event of join or leave event, affect only members of its cluster and does not affect the other clusters. However, the double operation of decryption and re-encryption at the side of the local controllers is a problematic disadvantage.

The second family of decentralized protocols uses only one traffic encryption key for all the group members. We call this protocol family *common TEK protocols*. The source of the group uses the TEK to encrypt the multicast flow and the members to decrypt it. Thus, the intermediary encryption and decryption operations of the multicast flow are not required. The principal issues of this family are to send the TEK securely and without delay to all the group members, and to define the TEK renewal period for all the group members. A vulnerability period corresponds to the case when a node leaves the multicast group and continues to access the multicast flow, until the next TEK renewal process, or a member joins the group and can access the past sent data encrypted with the TEK that it holds. This vulnerability period should be controlled by the source of the group, according to the importance and the confidentiality of the sent data.

14.3.6.1. *Local TEK protocols*

The protocols defined by Varadharajan *et al.* [VAR 01] and Enhanced BAAL [BOU 04] adopted the local TEK approach. We present them below.

14.3.6.1.1. The Varadharajan *et al.* protocol

The group key management protocol proposed in [VAR 01] operates within NTDR (*Near Term Digital Radio*) networks. The architecture of a NTDR network is composed of a set of clusters, each one containing a cluster head. The set of cluster heads forms the backbone of the network routing. Inter-cluster communications are restricted to the cluster heads (see Figure 14.13), which share a symmetric encryption key noted CHG_K (*Cluster Heads Group Key*). A cluster is composed of local nodes, at one hop from their cluster head. All the group members of a NTDR network hold certificates, received off-line, generated by a certification authority.

Node mobility is considered within this protocol, at the setting up of the clusters and at the election of the clusters heads. Indeed, each node behaves as a cluster head if it does not detect any other cluster head within its neighborhood. Dedicated mechanisms are used to limit the number of members behaving as cluster heads simultaneously. As soon as a node is elected to be cluster head, it immediately notifies all its local members about its new state.

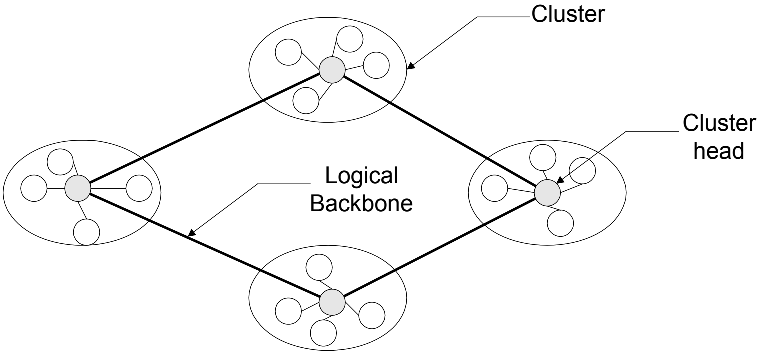


Figure 14.13. *Architecture of a NTDR network*

The functions carried out by a cluster head within its cluster are principally the maintenance of the list of its neighbors, the acceptance or refusal of a join request of a new member (through its certificate) and the forwarding of inter- and intra-packets. A notification procedure is proposed in [VAR 01], preceding a movement or a leaving of a cluster head, thus anticipating a re-election phase of another cluster head within the cluster.

The confidentiality of multicast communications is achieved via two types of keys:

– a local key for each cluster (GCK), used for the encryption of intra-cluster data;

– KEKs, shared between a cluster head and each member of its cluster. This key is a combination of a shared secret *s* and the IP address of the member, as follows: KEK= f(*s*, @IP).

The head of a cluster encrypts the GCK by the KEKs, and sends it to its local members respectively. Thus, all the group members can encrypt and decrypt data within their clusters.

14.3.6.1.2. The enhanced BAAL protocol

The enhanced BAAL protocol [BOU 04] is based on a combination of the BAAL protocol [CHA 02] (group key management protocol within wired networks) associated with the dynamic support of the AKMP (*Adaptive Key Management Protocol*) [BET 02]. The authentication and the generation of keys are carried out using the threshold cryptography technique [ZHO 99]. Each entity of the group holds its public and private keys generated by the server nodes of the threshold cryptography. The principal actors of the enhanced BAAL protocol are the global controller (GC), the local controllers (LCs) and the members of the multicast group. The GC is the source of the multicast group, and is responsible for the generation, the distribution and the periodic renewal of the TEK. In order to generate the TEK, the GC sends a request (Key-Request) to a defined number of server nodes of the threshold cryptography, which answer by sending their contributions. The GC then builds the TEK as a combination of these contributions, and distributes it to the members of its group. This key generation distribution is secure. It ensures the authenticity of the generated keys. In addition, it reduces the responsibility of the global controller which is characterized by limited capacities. Figure 14.14 illustrates this process. An LC is a member of the multicast tree, forming a cluster with its local members. The LC manages a local traffic encryption key within its cluster and is responsible for the forwarding of the multicast flow to its members. The renewal of the local encryption key is carried out after each join or leave event within a cluster, thus guaranteeing backward and forward secrecies. A member of the multicast tree can switch to the local controller state, according to an evaluation function which measures two metrics: the join and leave event frequency and the number of local members. This function is an extension of the one presented in [BET 02] and takes into account the mobility of nodes in the evaluation.

14.3.6.2. *Common TEK protocols*

The BALADE protocol [BOU 05a] uses only one TEK. BALADE is a group key management protocol, dedicated to multicast communications within MANETs, following a sequential multi-source model. According to this model, at each moment t, there is only one source which issues data, and when it finishes another source takes over. Several applications follow this model, like audio-video conferences, cooperative jukebox applications, etc.

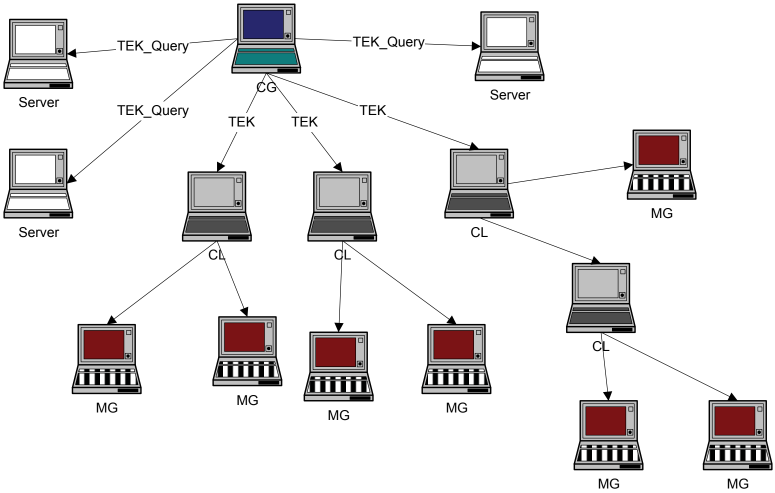


Figure 14.14. *Generation and distribution of the TEK in the enhanced BAAL protocol*

The security operations carried out by BALADE are data confidentiality and the authentication and access control of group members. The identification of the entities in the networks is done though the cryptographic identifiers CBIDs [MON 02]. The basic idea of BALADE is to divide the multicast group dynamically into clusters. Each cluster is managed and supervised by a local controller which shares a cluster key with its local members. Figure 14.15 presents the hierarchical structure of the BALADE protocol. The multicast flow is encrypted by the source using the TEK key, and sent in multicast to all the group members. The source sends the TEK to the local controllers, encrypted with a KEK. These local controllers then forward the TEK to their local members, encrypted with their respective cluster keys. Consequently, only the TEK is decrypted and re-encrypted by the local controllers while the multicast sent data flow remains unaffected. The TEK is renewed at each data unit sent by the source, according to the semantics of the multicast flow.

BALADE proposes to manage the mobility and the dynamic of the multicast groups, adapted to the nature of the ad hoc networks. To do this, a dynamic clustering algorithm, called OMCT (*Optimized Multicast Cluster Tree*), is used [BOU 05b, BOU 05c]. This algorithm considers the geographical locations and the mobility of nodes, while optimizing energy and bandwidth consumption.

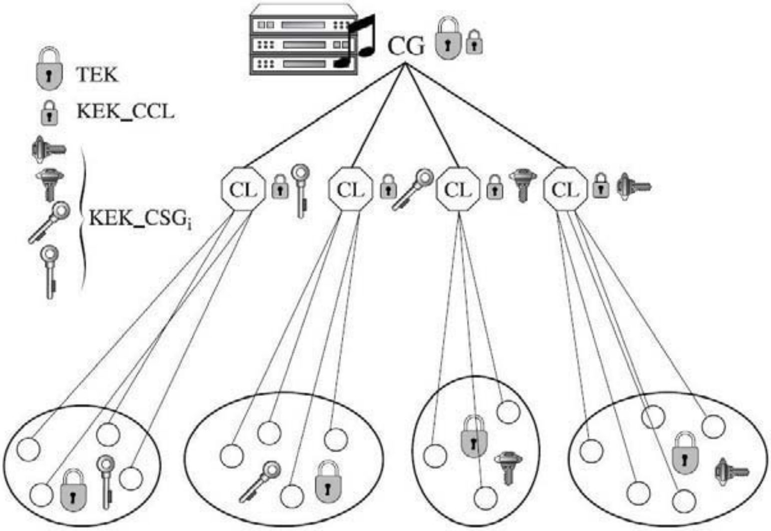


Figure 14.15. *Group member management in BALADE*

The source of the group starts by encrypting the multicast flow by the TEK. Then, it sends it to the group members following the multicast data transmission tree. At the initialization of the application, all group members receive a session key, called CSG_0 (key of the sub-group 0), sent by the source of the group. Then, dynamically, new clusters will be created according to the OMCT algorithm. Each cluster *i* has a local controller LC_i and shares a cluster key CSG_i. To send the TEK to all group members, the source encrypts it with the CSG_0 key and sends it to the members of its cluster. Then, it sends the TEK to the group formed by the LCs (this group shares a group key called K_{CCL}), encrypted with the K_{CCL} key. The local controllers belonging to this group decrypt the received message, extract the TEK, re-encrypt it with their respective cluster keys and send the new formed message to their local members. When a source finishes sending its multicast flow and another source takes over, the key distribution tree still remains unchanged. An illustration of the TEK distribution process is presented in Figure 14.16.

Access control in BALADE is ensured through an access control list (ACL) containing the CBIDs of the authorized members to join the multicast group. The ACL list is managed in a cooperative and distributed manner by all the local controllers responsible for its maintenance, its availability, its accessibility and its coherence. The redundancy of the ACL is also proposed by the BALADE protocol, in order to avoid the possible loss of stored data.

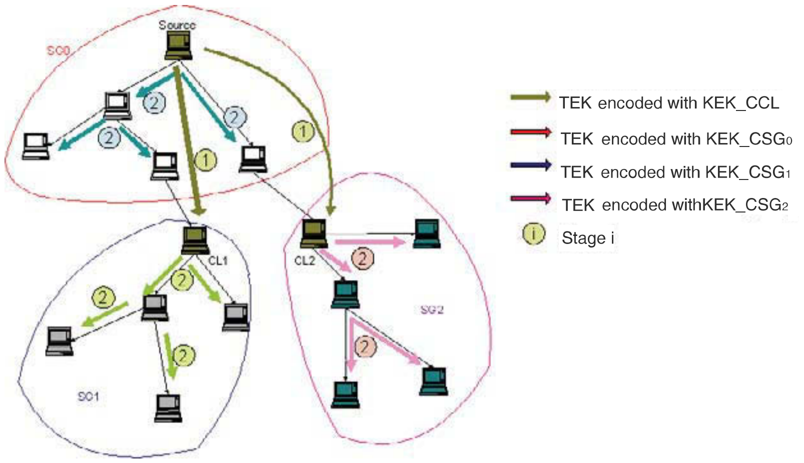


Figure 14.16. *TEK distribution within BALADE*

14.4. Discussions

In this section, we evaluate and compare the presented group key management protocols and evaluate their performance and their security properties (the comparison metrics we use are presented in section 14.3.3). Table 14.1 summarizes these comparisons and analysis results.

14.4.1. *Constraints and pre-requisites*

The proposals of Kaya *et al.*, Chiang *et al.*, Lazos *et al.* and BALADE require a GPS localization system to take into consideration the geographical positions of the group members. The GPS information is used in both Kaya *et al.* and Lazos *et al.* to efficiently build paths between the group members. However, in Chiang *et al.*, the GPS information is flooded within the network, allowing each node to know the entire topology of the network. This flooding operation is very constraining within MANETs, which makes the effective applicability of the protocol difficult.

In addition to the clustering algorithms used in enhanced BAAL and Varadharajan *et al.*, enhanced BAAL requires the availability of the threshold cryptography technique, which needs an initial configuration of the network, in order to divide the private secret of the certification authority to the server nodes.

All the proposed protocols that require a public key for each member [KAY 03, VAR 01] assume the availability of a certification authority within an ad hoc network able to provide proof of the member identities. This constraint is very difficult to satisfy within an environment without a fixed infrastructure, where links are transitory and dynamic. BALADE uses the CBIDs to ensure the identification of the group members. This technique assumes the knowledge of a public and private key by each member of the group, allowing them to compute their CBIDs respectively. The availability of a certification authority is not required within this protocol. Indeed, a node can create its public and private keys, and compute its unique CBID to cryptographically bind its created keys, and thus to be identified within the network.

The validation of the list of keys in Kaya *et al.* and GKMPAN requires the TESLA authentication, the temporal synchronization between members of the group, and the buffering of the received messages at the receiver node side. These requirements are difficult to achieve within an ad hoc network, in which links between nodes are not fixed and storage capacity is limited.

14.4.2. *Security services*

The security services ensured by group key management protocols presented in this chapter include data confidentiality, carried out via encrypting the multicast flow by the source of the group, and decrypting it by the receivers. Authentication and access control are only provided by Kaya *et al.* [KAY 03], enhanced BAAL [BOU 04] and BALADE.

In Kaya *et al.*, the certification authority offers security certificates to all group members off-line, allowing them to authenticate themselves, prove their identities and join the multicast group on-line. The certification management in enhanced BAAL is realized via the threshold cryptography, suitable for ad hoc networks. The cryptographic identifiers technique used in BALADE allows the identification of the group members registered within the access control list, since it ensures a strong cryptographical connection between the public and private keys of the CBID holder.

The revocation of malicious nodes is ensured with the key pre-distribution process of GKMPAN [ZHU 04] and CKDS [MOH 04]. Within these two protocols, keys of an excluded node are also compromised and isolated and will no longer be used for the key renewal processes by the other group members. However, the addition procedure of a new member to the group is difficult to deploy within these protocols, because new members should hold pre-deployed keys.

			Constraints and pre-requisites	Security Services	Computation Overhead	Storage Overhead	Communications Overheads	Vulnerabilities
Centralized	With key pre-distribution	**GKMPAN**	- Key pre-distribution -Synchronization	- Node revocation - Data confidentiality	- TEK decryption and encryption - TESLA	- m + 3 pre-distributed keys - TESLA bufferization	- Key routing, hop by hop, depending of *m* and *l*	Key Server
		CKDS	- Key pre-distribution - Global controller - EBS and CAN	- Node revocation - Data confidentiality	- m-dimensional scheme: TEK decryption and encryption - 2D-multicast: n	- K pre-distributed keys - EBS matrix at the source of the group (N * (k+m))	- M-dimensional scheme: - Flooding of key renewal messages sent by IGD	Global controller
	Without key pre-distribution	**Kaya *et al.***	- GPS -Certification authority - Synchronization	-Authentication & access control - Node revocation - Data confidentiality and integrity	- Revocation list management - Multicast data decrypted and re-encrypted	- Revocation list - Certificates - Packet bufferization for TESLA	- Maintenance messages and optimization of the multicast tree	- Revocation list updating
		Lazos *et al.*	- GPS - K-means algorithm	- Data confidentiality	- $O(n^2)$ clustering algorithm	- LKH tree key controller: O(N) Members: O(log N)	- Group initialization (n messages of LKH)	- Source
		LKHW	Direct diffusion algorithm	- Integrity, authentication and data confidentiality	- Message hashing and key generation	- LKH tree keys: O(N) Members: O(log N)	- Initialization (3*n) - Message flooding	- Source
Distributed		**Chiang *et al.***	- GPS - GDH protocol	- Data confidentiality	- Public key computation - Prüfer coding O(n)	- Prüfer sequence O(N)	- Flooding of the GPS localization and the public keys	- GPS flooding - High overheads
		DMGSA	- Clustering algorithm	- Data confidentiality	- Decryption and re-encryption of the TEK - Neighborhood detection	- Shared KEKs between a GCKS and each member of its cluster	- Clustering initialization O(n) - Cluster messages maintenance	- Cluster heads
Decentralized	Local TEKs	**Enhanced BAAL**	- Clustering function - Threshold cryptography	- Authentication & access control - Data confidentiality	- Multicast data decrypted and re-encrypted by LCs	- Multicast data decrypted and re-encrypted by LCs - Public and private keys	- Key generation messages - Notifications to parent LC	- Global controller
		Varadharajan *et al.*	- Clustering algorithm - Certificates and public keys	- Data confidentiality	- Multicast data decrypted and re-encrypted by cluster heads	- Multicast data decrypted and re-encrypted by cluster heads	Routing of all messages by the cluster heads	- Cluster heads
	Common TEK	**BALADE**	- Clustering algorithm	- Authentication & access control - Data confidentiality	- Decryption and re-encryption of the TEK by LCs -OMCT $O(c^2)$	- CBID - KEK per cluster - Distributed ACL (f*n/k)	- OMCT: CG: n+k messages CL: c messages	- Current global controller

Table 14.1. *Evaluations of group key management protocols within MANETs*

14.4.3. *Computation overhead*

The metric of intermediary encryption and decryption of the multicast flow is very important within ad hoc networks, because of the generally limited capacities of equipment and entities of the network. A suitable group key management solution dedicated to operate within MANETs should not require intermediary operations of either encryption or decryption of the multicast flow. Thus, transmitted data should only be decrypted by the final receivers, as for the protocols of Kaya *et al.*, Lazos *et al.*, LKHW, Chiang *et al.* and the 2D-multicast version of CKDS. These protocols suffer from the fact that they are centralized around only one entity of the network responsible for the generation and the distribution of the traffic encryption key, in addition to the sending of the encrypted multicast flow. This centralization around only one key server increases the "1 affects n" phenomenon, consisting of affecting all the group members at any change of a state of only one member (particularly after each addition or withdrawal of an entity within the multicast group). To reduce this phenomenon and avoid the use of intermediary operations of flow encryption and decryption, several protocols use the clustering approach and choose to delegate the key management task to special entities of the network other than the key server. These entities are the local controllers in BALADE and the cluster heads within the DMGSA protocol.

In order to forward the traffic encryption key to their local members, the local BALADE controllers should decrypt it, re-encrypt it with their local keys and send it in multicast to their local members. However, in the DMGSA protocol, the sending of the TEK to the local members of a cluster is carried out individually (in unicast) between a cluster head and each member of its cluster, which induces non-negligible overhead communications in ad hoc networks. The protocols proposed in [BOU 04], [VAR 01] and [KAY 03] are not well suited to low-computation capacities equipment, since intermediary encryption and decryption operations are required. In addition, these operations are carried out by the local controllers or the cluster heads, which consequently become vulnerability points and bottlenecks.

14.4.4. *Storage overhead*

The control of storage overhead is mandatory within ad hoc networks. The protocols belonging to the decentralized approach with local TEKs (enhanced BAAL and Varadharajan *et al.*) induce a high storage overhead because of the intermediary encryption and decryption operations of the transmitted multicast flow. The Prüfer algorithm used in Chiang *et al.* also requires a large memory space, especially for a large number of group participants. Note that any change in the topology of the network affects the Prüfer sequence and consequently the corresponding multicast tree. A high mobility of the nodes has a large impact on the storage overhead in the Chiang *et al.* protocol.

The storage in the Lazos *et al.* protocol and LKHW concerns the keys of the LKH tree. Their number depends on the total number of members in the group, whereas GKMPAN and CKDS store the pre-distributed keys for each node independently of the total number of group participants.

For the GKMPAN protocol, increasing the number *m* of pre-distributed keys or diminishing the number *l* of initially available keys will increase the number of direct paths between the participants. The number of common keys that two members know is evaluated as m^2/l. For example, for m = 100 and l = 2,000, 0.5% of the members will receive the renewal messages in an indirect manner (forwarded by their neighbors). However, it is preferable from both a security and storage overhead point of view to diminish *m*. The smaller *m* and the larger *l*, the smaller the risk of coalition between malicious members. Consequently, the security level is higher and the risk of attacks is smaller. The choice of *m* and *l* should thus consider the security policies and choices of the concerned application.

Within the CKDS protocol, the storage of the EBS matrix at the side of the global controller is very constraining, because its size is equal to $N * (k+m)$, *N* being the number of members of the group, and k and m the number of known (respectively unknown) keys, by a group member in the EBS system.

Being certificate-based, the approach of Kaya *et al.* implies that each member of the group stores its certificate and the revocation list sent and updated by the source of the group. To prevent this list from reaching too great a size, an entry removal technique is used periodically, at the risk that excluded members can join the group after a certain delay.

The distributed management of the access control list in the BALADE protocol implies storage overhead at the local controllers' side. If *n* is the number of authorized members to join the multicast group, *k* is the number of local controllers of the group and *f* is the redundancy number required by the security policies. The number of ACL fields that each local controller must store is thus $f * n/k$.

14.4.5. *Communication overhead*

Protocols without a key pre-distribution phase are not scalable because of their centralized architecture ("1 affects n" phenomenon). The protocol proposed by Chiang *et al.* also has a scalability problem in terms of communication overhead, due to the GPS information flooding to all group members, and to the constraining execution of the Prüfer algorithm for a large number of participants in the multicast group.

The DMGSA protocol is limited by the number of members by cluster, because each cluster head shares with each member of its cluster a secret key to encrypt the traffic encryption key and send it in a secure manner. In addition, the distributed maintenance of the clusters requires the sending of periodic messages, thus implying an important communication overhead.

In the *m*-dimensional scheme of the CKDS protocol, the IGD entity floods the network with messages containing the new group keys, sent to the LQDs. These flooding operations are very constraining in term of communications and bandwidth overheads, and require additional intermediary decryption and re-encryption operations of the sent keys. Moreover, members receiving these messages are merely interested in a subset of the distributed keys, and not in all the proposed updated keys. The 2D-multicast CKDS scheme solves this problem by sending key distribution messages in multicast only to members interested in these renewals.

14.4.6. *Vulnerabilities and weaknesses*

Centralized protocols [ZHU 04, MOH 04, KAY 03, LAZ 03, PIE 03] are based on only one entity of the ad hoc network responsible for the management of keys and certificates of the group members. This centralized entity constitutes a vulnerability point in terms of security. In addition, a centralized server represents a bottleneck and can be the target of several malicious DoS attacks. Although the centralized entities are always chosen so that they have better capacities and performance, they cannot be available in the network due to a battery problem or because of their moving.

In the protocol presented in [VAR 01], the cluster heads form the backbone of the network routing. In addition, they assume the key management task. These entities represent weakness and vulnerability points and can be targeted by several malicious attacks. The same issue is present within the enhanced BAAL protocol, where local controllers are responsible for the key management within their clusters, in addition to the forwarding of the secure multicast data flow sent by the source of the group to their local members.

The communications model adopted by the BALADE protocol is the sequential multi-source model; at any moment *t*, only one source acts as a global controller and is thus responsible for the diffusion of the secure data, in addition to the TEK distribution to the group members. The source can consequently represent a security vulnerability point. However, it is only temporary, as the source changes over time.

14.5. Conclusions

During the last few years, several research works were interested by the authentication issue within ad hoc networks. The lack of fixed infrastructure of these networks makes the applicability of a centralized architecture difficult. Some approaches such as [ZHO 99] and [ASO 00] tried to solve this problem by duplicating the certification authority within MANETs or by delegating the key management task to all group members in a distributed manner. These new approaches consequently allow the establishment of secure multicast communications within ad hoc networks, while adopting the specific context of these environments.

Securing group communications within ad hoc networks requires the deployment of a group key management protocol. This protocol should ensure data confidentiality by encrypting the multicast flow at the source of the group and decrypting it at the receivers with a symmetric TEK. In addition, authentication and access control should be ensured; only members holding the traffic encryption key should be able to access the multicast flow.

However, the design of a group key management protocol within MANETs needs to be adapted to the characteristics and specificities of such environments, such as the mobility and dynamics of nodes, the limited resources in terms of energy, bandwidth, storage and computation, in addition to the lack of fixed infrastructure. Security services provided by a group key management protocol are also highly dependent of the nature of the multicast application to secure, associated with the security level required by the established security policies to face possible malicious attacks.

In a military application for example, transmitted data is highly confidential, thus requiring a high security level. Forward and backward secrecies should consequently be ensured, during the session of a multicast group. If the transmitted data is not of large size and is not sent in a burst manner, a centralized group key manager could be suitable. However, if the group is formed by a large number of members, and to avoid the “1 affects n phenomenon”, the decentralized approach is the most appropriate. On the other hand, to secure multicast communications of a small group of users (e.g. ten people in a meeting room), the choice of using a distributed group key management protocol will be judicious, because it allows the collaboration and the cooperation of all the group entities in an equitable and equivalent manner. Finally, the decentralized approach with common TEK (BALADE) is the most suitable for multicast data streaming within ad hoc networks to a large number of users because this protocol takes into consideration of the semantics of data, while being adapted to the nature of MANETs.

The choice of a group key management protocol within MANETs proves to be dependent on the required services by the concerned multicast-oriented application, in addition to the constraints and challenges imposed by the nature of the ad hoc networks.

14.6. Bibliography

[ASO 00] Asokan N. and Ginzboorg P., "Key agreement in ad hoc networks", *Computer Communications* 23(17), pp. 1627-1637, 2000.

[BET 02] Bettahar H., Bouabdallah A. and Challal Y., "An adaptive key management protocol for secure multicast", in *11th International Conference on Computer Communications and Networks (ICCCN)*, Florida, USA, October 2002.

[BOU 04] Bouassida M.S., Chrisment I. and Festor O., "An enhanced hybrid key management protocol for secure multicast in ad hoc networks", in *Networking 2004, Third International IFIP TC6 Networking Conference*, Athens, Greece, May 2004, volume 3042 of Lecture Notes in Computer Science (LNCS), pp. 725-742, Springer.

[BOU 05a] Bouassida M.S., Chrisment I. and Festor O., "BALADE : Diffusion multicast sécurisée d'un flux multimédia multi-sources séquentielles dans un environnement ad hoc", in *CFIP 2005*, Bordeaux, France, March 2005.

[BOU 05b] Bouassida M.S., Chrisment I. and Festor O., "Efficient clustering for multicast key distribution in MANETs", in *Networking 2005, International IFIP TC6 Networking Conference*, Waterloo, Canada, May 2005, Volume 3462 of Lecture Notes in Computer Science (LNCS), pp. 138-153, Springer.

[BOU 05c] Bouassida M.S., Chrisment I. and Festor O., "Prise en compte de la mobilité dans le protocole de gestion de clé de groupe BALADE", in *SAR Sécurité et architecture des réseaux*, 2005.

[BOU 08] Bouassida M.S., Chrisment I. and Festor O., "Group key management in MANETs", *International Journal of Network Security (IJNS)*, 6(1): 67-79, 2008.

[CHA 02] Chaddoud G., Chrisment I. and Schaff A., "BAAL : Sécurisation des communications de groupes dynamiques", in *8th Colloque Francophone sur l'Ingénierie des Protocoles CFIP'2000*, Toulouse, France, October 2000.

[CHI 03] Chiang T. and Huang Y., "Group keys and the multicast security in ad hoc networks", in *Proceedings of the International Conference on Parallel Processing Workshops (ICPP 2003 Workshops)*, 2003.

[DEE 91]Deering S., Multicast Routing in a Datagram Internetwork, PhD Thesis, Stanford University, December 1991.

[DEE 94]Deering S., Estrin D. and Farinacci D., "An architecture for wide-area multicast routing", in *ACM SIGCOMM*, pp. 126-135, August 1994.

[ELL 99] Ellison C., Frantz B., Lampson B., Rivest R., Thomas B. and Ylonen T., *RFC 2693 - SPKI Certificate Theory,* September 1999.

[FRA 99] STAJANO F. and ANDERSON R., "The Resurrecting Duckling: security issues for ad hoc wireless networks", in *Security Protocols, 7th International Workshop Proceedings*, Lecture Notes in Computer Science, 1999.

[HAR 03] HARDJONO T. and DONDETI L., *Multicast and Group Security*, Computer Security Series, Artech House, 2003.

[HOU 99] HOUSLEY R., FORD W., POLK W. and SOLO D., *RFC 2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, January 1999.

[HUB 01] HUBAUX J., BUTTYAN L. and CAPKUN S., "The quest for security in mobile ad hoc networks", in *ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*. 2001.

[ING 82] INGEMARSON I., TANG D. and WONG C., "A conference key distribution system", in *IEEE Transactions on Information Theory*, September 1982.

[KAY 03] KAYA T., LIN G., NOUBIR G. and YILMAZ A., "Secure multicast groups on ad hoc networks", in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, Virginia, pages 94-102. ACM Press, 2003.

[KON 06] KONG J., LEE Y. and GERLA M., "Distributed multicast group security architecture for mobile ad hoc networks", in *IEEE Wireless Communications and Networking Conference (WCNC)*, Las Vegas, USA, April 2006.

[LAO 03] LAOUITI A., JAQUET P., MINET P., VIENNOT L., CLAUSEN T. and ADJIH C., *Multicast Optimized Link State Routing.* Research Report 4721, INRIA, February 2003.

[LAZ 03] LAZOS L. and POOVENDRAM R., "Energy-aware secure multicast communication in ad hoc networks using geographical location information", in *IEEE International Conference on Acoustics Speech and Signal Processing*, 2003.

[LUO 00] LUO H. and LU S., *Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks*, Technical report TR-200030, Department of Computer Science, UCLA, 2000.

[LEG 03] LEGRAND V., *Etablissement de la Confiance et Réseaux Ad Hoc - Le Germe de Confiance*, DEA report, EDIIS, CITI Laboratory, INRIA ARES, July 2003.

[MAC 67] MACQUEEN J.. "Some methods for classification and analysis of multivariate observations", in *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281-297, Berkeley, University of California Press, 1967.

[MOH 04] MOHARRUN M. and MUKKALAMALA R. and ELTOWEISSY M., "CKDS: an efficient combinatorial key distribution scheme for wireless ad hoc networks", in *IEEE International Conference on Performance, Computing and Communications (IPCCC'04)*, Arizona, April 2004.

[MON 02] MONTENEGRO G. and CASTELLUCCIA C., "Statistically unique and cryptographically verifiable identifiers and addresses", in *ISOC Network and Distributed System Security Symposium (NDSS)*, February 2002.

[MOR 03] MORALES L., SUDBOROUGH I., ELTOWEISSY M. and HEYDARI M.H., "Combinatorial Optimization of Multicast Key Management", in *IEEE International Conference on System Sciences*, Hawaii, January 2003.

[MOY 94] MOY M., "Multicast routing extension for OSPF", *ACM*, 37(8): 61-66, August 1994.

[PER 02] PERRIG A., CANETTI R., TYGAR D. and SONG D., "The TESLA broadcast authentication protocol, *RSA Laboratories Cryptobytes*, 5(2), 2002.

[PIE 03] DI PIETRO R., MANCINI L., LAW Y., ETALLE D. and HAVINGA P., "LKHW: a directed diffusion based secure multicast scheme for wireless sensor networks", in *International Conference on Parallel Processing Workshops (ICPPW'03)*, Taiwan, October 2003.

[PRU 18] PRÜFER H., "Neuer Beweis eines satzes über Permutationen", in *Archive der Mathematik und Physik*, volume 27, pp. 742-744, 1918.

[RAT 01] RATNASAMY S., FRANCIS P., HANDLEY M., KARP R. and SCHENKER S., "A scalable content-addressable network", in *SIGCOMM'01 Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications,* pp. 161-172, New York, USA, ACM Press, 2001.

[ROY 00] ROYER E. and PERKINS C., *Multicast Ad Hoc On-Demand Distance Vector (MAODV) Routing,* IETF Internet Draft, 2000.

[VAR 01] VARADHARAJAN V., HITCHENS M. and SHANKARAN R., "Securing NTDR ad hoc networks", in *IASTED International Conference on Parallel and Distributed Computing and Systems*, Anaheim, California, pp. 593-598, August 2001.

[WON 98] WONG C., GOUDA M. and LAM S., "Secure group communications using key graphs", in *ACM SIGCOMM*, pp. 68-79, 1998.

[YI 02] YI S. and KRAVETS R., *Key Agreement for Heterogeneous Ad Hoc Networks*. Technical Report. University of Illinois at Urbana-Champaign, Department of Computer Science, July 2002.

[ZHO 99] ZHOU L. and HAAS J., "Securing ad hoc networks", *IEEE Network,* 13(6): 24-30, 1999.

[ZHU 04] ZHU S., SETIA S., XU S. and JAJODIA S., *GKMPAN: An Efficient Group Rekeying Scheme for Secure Multicast in Ad Hoc Networks,* Technical report, February 2004.

This page intentionally left blank

Chapter 15

Wireless Sensor Network Security

15.1. Introduction

Wireless sensor networks (WSNs) can be compared to ad hoc networks, but they are characterized by a large number of sensor devices called nodes with severe restrictions in terms of energy, processing and communication capabilities. Typically, sensors operate in remote hostile environments and with no possibility for recharging their batteries. The WSNs collect the monitoring data from the sensors and make decisions on the environment in which sensors are deployed. Data are usually collected by a base station (BS) for subsequent analysis. A network being composed of hundreds or even thousands of sensor nodes can generate a large amount of data, so the challenge is to extend the lifetime of sensors by designing the least resource-consuming communication mechanisms. One of these designed mechanisms is the aggregation of data or messages that serves to reduce the time transmission.

WSN are vulnerable to various types of attacks [KAR 03], [WOO 02], due to the nature of wireless communications, the physically unprotected environments where sensors are deployed and the nature of the sensors themselves that are small and low-cost.

Chapter written by José-Marcos NOGUEIRA, Hao-Chi WONG, Antonio A.F. LOUREIRO, Chakib BEKARA, Maryline LAURENT-MAKNAVICIUS, Ana Paula RIBEIRO DA SILVA, Sérgio de OLIVEIRA and Fernando A. TEIXEIRA.

Preventive mechanisms can be used to protect against certain types of WSN attacks [KAR 04], [PER 02]. Section 15.2 details one of them: the protocols that ensure the confidentiality, integrity, freshness and non-repudiation of data exchanged and authentication of their origin.

However, these prevention methods are sometimes ineffective against some attacks, such as the wormhole attack [KAR 03], [HU 06]. In addition, there is no assurance that the preventive methods are able to prevent intrusions. As a consequence, other strategies are advocated, such as intruder tolerance and intruder detection. In the first strategy, the network aims to protect itself or reduce the effects of an ongoing attack. In the second strategy, the intrusion is detected and appropriate measures to exclude the intruders are adopted. The second strategy of intruder detection is also interesting because it helps to acquire information on the attack techniques, and thus improves the prevention systems.

The hypothesis for intruder detection is that the intruder's behavior can be quantified as different from the behavior of the legitimate user [STA 98]. The behaviors of the user are modeled and compared with the observed behavior of the system; the probability of the system to behave as a victim of an intrusion is then evaluated.

Intruder detection in WSNs needs to address several scientific challenges. WSNs are application oriented, i.e. they have very specific characteristics that are depending on the application they are addressing. The various WSN configurations make it difficult to model the "normal" or "expected" behavior of the system. Moreover, the methods developed for traditional networks are not applicable, because of the availability of resources in these networks that are much larger than in WSNs.

In the context of this chapter, an application is a set of programs that execute tasks for the benefit of users, like acquisition of temperature data or chemical composition of the environment. Normally an application runs in both the sensor nodes and the BS, as well as in computers outside the network.

The preventive mechanisms may not be sufficient to prevent all types of attacks. In some cases, the attacks may be played despite active preventive mechanisms. In these cases, the strategy of tolerance to intruders is adopted, in which the network takes measures to protect itself or reduce the effects of the attack. Tolerance is a current research topic that raises several problems. A network tolerating intruders adds the ability to survive intrusions to a network focused on prevention. In this case, the network is said to evolve from prevention to complete resilience. Some techniques of intrusion tolerance involve changing the routing of networks, by introducing additional routes for each message's source-destination pair.

In this chapter, the main types of attacks against WSNs are presented as well as various types of counter-measures that can be adopted to protect networks against these attacks (section 15.2). Section 15.3 presents all the prevention systems that are based on the traffic protection in WSNs. The remainder of the chapter focuses on the mechanisms for intruder tolerance and intruder detection. Three case studies (sections 15.4 to 15.6) illustrate the different strategies to deal with intruders in the network. Each study proposes a mechanism, discusses its advantages and disadvantages and presents experimental data on the efficiency of these mechanisms. Finally, section 15.6 gives the conclusions.

15.2. Attacks on wireless sensor networks and counter-measures

Various types of attacks against wireless sensor networks are documented in the literature. To cope with these attacks, counter-measures have been proposed. The following sections introduce the main attacks (section 15.2.1) and the main available counter-measures (sections 15.2.2 to 15.2.4). These counter-measures are described in more detail in sections 15.3 to 15.5.

15.2.1. *Various forms of attacks*

A large number of attacks can be performed over a WSN with different objectives. For example, one of the attacks can target the integrity of the messages passing through the network, while others aim to reduce the availability of the network or its components. The attacks often occur by injecting some intrusive elements into the network. Other attacks acting on the external environment itself can indirectly cause deterioration or interference with transmitted signals. A good classification of attacks is presented in [WOO 02].

The best-known attacks against the WSN are the following:

– Jamming: the intruder floods the radio frequencies used by the network with noise and can prevent any exchange of messages. The network can be strongly disrupted if the radio coverage of the intruder is large. The consequence of this attack is a denial of service (DoS).

– Eavesdropping: no access control to the network is possible because the communications are broadcast through radio waves, and moreover the network might be deployed in an open environment that is accessible to everyone. As such, it is very easy to intercept data exchanged over a sensor network and to access their content if no confidentiality service is provided.

– Physical violation (tampering): WSNs are often deployed in unprotected areas, so an intruder may have physical access to the nodes, and may violate the hardware of the nodes. The objective might be to extract secret information, such as cryptographic keys, or to disrupt voluntarily the network and application, thus causing abnormal behavior of the node.

– Neglect and greed: the intruder totally or partially removes data messages generated by the node that is subject to the attack.

– Blackhole or sinkhole: the intruder is positioned at a routing strategic point of the network and it deletes all the messages instead of forwarding them. Thus, the routing service is suspended for all the routes that go through the intruder's node.

– Selective forwarding: the intruder's node does not route the message, as required. The selection of deleted messages is done according to certain criteria or randomly.

– Wormhole: the intruder captures a message and redirects it to a remote node of the WSN through a low latency channel. As a consequence, a channel is created and messages go through some nodes that should have never seen the messages or that should have seen the messages but with a greater latency. This attack has a significant influence on routing.

– Replay, delay and data corruption: the intruder replays, delays or alters the content of messages in transit. The messages might contain collected data and configuration or routing data. The objective is to create loops, attract or repel the traffic, increase or decrease the number of routes, generate false errors, partitioning the network, and increase the latency for the data distribution.

– Exhaustion of the battery: this DoS attack is critical as exhaustion of battery of the nodes composing the network highly affects the lifetime of the network. Battery exhaustion can be conducted by injecting many messages into the network so that the nodes are wasting their energy in unnecessary retransmissions.

15.2.2. *Preventive mechanisms*

Prevention must remain the major concern of any network administrator anxious to protect a system. WSN should be protected against tapping and against the intrusion of some nodes that could spoof the identity of a legitimate sensor, disrupt routing or strongly encourage sensors to overconsume their energy and reduce their lifetime, etc.

Preventive mechanisms make use of cryptographic primitives to guarantee confidentiality, authenticity, integrity and freshness of information in transit over the network. They protect all the exchanges between the nodes and the BS which is responsible for collecting data from sensors [PER 02], or between two neighboring nodes. In the latter case, the messages are protected hop-by-hop between any pair of nodes [PER 04] and it is very difficult for intruders to interfere with the network using its own hardware. However, whatever the robustness of these cryptographic primitives, the intruder will still be able to take physical control of a legitimate node, to insert malicious code in it and thus change that node into an intruder. The physical security of nodes might be strengthened, but no effective and low cost technique is known so far.

All these mechanisms are described in section 15.3 with their consumption in energy and memory, and their advantages and disadvantages.

As preventive mechanisms are insufficient to guarantee the security of a WSN, there is a need to introduce intrusion tolerance mechanisms and deploy new tools for detecting and revoking intruders. This will help increasing the network security.

15.2.3. *Intruder detection*

The intruder detection is a very active research topic, even in traditional networks. The main motivation for developing intrusion detection systems is based on the fact that it is not possible to create a totally infallible defensive mechanism. After detecting an intrusion, it is possible to check whether a defensive mechanism has been violated, and then to launch an automatic reaction and to let the network administrator take a decision. In addition, the information provided by an intrusion detection system can be used to improve the defensive mechanisms of the network.

In an intrusion detection system, the behavior of the target under protection is controlled and analyzed. Analysis of it assumes that the behavior of the intruders, the normal behavior of the system or the behavior expected from the system are known. According to the class of behaviors under consideration, there are two strategies for detection [AMO 04]:

– Anomaly detection [GHO 98], [KO 97], [LAN 99]: the observed behavior of the target system is compared to normal and expected behavior. If the behavior of the system is significantly different from the normal or expected behavior, the system is encountering anomalies and is victim of an intrusion.

– Misuse detection [ILG 95], [PAX 98], [LIN 99]: the actions undertaken in the target system are compared to the actions usually carried out by intruders and listed in the form of signatures. An intrusion is detected when we succeed in identifying a signature from the actions under analysis.

The detection of intruders in WSNs requires a very different approach from that of conventional networks because models, attacks and resources are different. In conventional networks, the role of the user normally exists; the user is the one who uses the network and who generates his traffic profile. In a sensor network, events are monitored by sensor nodes that generate data and send them to a place where a user or an observer can proceed in the analysis of them. The behavior of the user, in an intruder detection context, is not interesting because the user has no influence on the behavior of the network, except in some rare situations when the user interacts with the network to perform configuration or stimulation of it.

Two alternatives for intruder detection are traditionally possible. In the centralized approach, the BS extracts from the network the information produced by the nodes and is responsible for detecting intruders. In the decentralized approach, all the nodes of the network or a subset of them watch their respective neighbors and perform simple intruder detection operations. Both approaches are presented in the following chapter in the form of case studies.

15.2.4. *Intrusion tolerance*

The intrusion tolerance is a third approach to security. In this approach, the idea is to make critical functions of the system as resistant as possible to any compromising attacks by an intruder.

In the context of WSNs, routing is at the heart of the majority of the works on intrusion tolerance. Several works define multiple routes for simultaneous or alternative usages, in order to guarantee full or partial delivery of messages [DEN 03, KAR 02, GAN 01]. Some other works attempt to establish new routes once communication problems are detected [STD 02].

Some intrusion tolerance techniques modify the routing of networks by defining additional routes for each source-destination pair of any messages. Designing routing with multiple routes enables total or partial continuity of operation in the network, even in the presence of intruders acting on routing. In this chapter, one of these proposals based on alternative routes [OLI 06] will be shown.

15.3. Prevention mechanisms: authentication and traffic protection

In order to limit the impact of the attacks on WSNs, several security protocols have been proposed in the literature since 2002. These protocols define mechanisms to protect data exchanges between sensors and between sensors and the BS. Offered security services include data confidentiality, integrity and freshness and authentication of data origin.

Before discussing in detail the SNEP, µTESLA and TinySec security protocols as well as [ZHU 04], section 15.3.1 gives the notations and section 15.3.2 presents a first analysis of the resources consumed by the security procedures in the sensors.

Note that this section does not address the fundamental issue of key distribution into sensors and BSs. This issue, which is also raised in ad hoc networks, is presented in Chapter 16 and will not be discussed further here.

15.3.1. *Notations of security protocols*

The description of security protocols refers to the following notations:

– BS: the base station serving as a gateway between the sensor network and external networks (other sensor networks, the Internet, etc.). The BS is regarded as a trusted entity in the network;

– A = {1, ..., n}: all the nodes forming the network of sensors;

– i: a sensor contributing to the sensors network;

– Ki: the master symmetric key shared between the BS and the node i;

– Kij: the master symmetric key shared between two nodes i and j;

– KEi = MAC (Ki, 1): a shared encryption key deduced from the key Ki;

– KAi = MAC (Ki, 2): a shared authentication key deduced from the key Ki;

– {M}<KEi, P>: the message M encrypted with the encryption key KEi and the parameter P;

– MAC (KAi, M): the message M authenticated with the authentication key KAi;

– CPTi: the counter shared between BS and the node i;

– K_g^k: a group key shared between BS and all the nodes forming the network.

15.3.2. *Cost of security protocols in sensors*

The introduction of security protocols in a sensor network can have devastating effects on the sensors. Since security is very energy consuming, it can strongly affect the lifetime of the sensors.

On the one hand, part of the energy is consumed by the processing being performed by sensors implementing the security functions. These functions must be selected carefully so that the associated code must be small (ROM), and the processing must be light on CPU consumption. Fulfilling these requirements will help to integrate new security functions into sensors without disrupting their basic operations. As such, it is better to avoid public key cryptography that is too CPU and memory consuming, and to make use of symmetric algorithms like RC5 (Rivest Cipher 5) or Skipjack because of the small size of their source code, their short running time and the small memory size (RAM) needed during their execution.

One idea to limit the size of the code in sensors is generally to use the same cryptographic tools to encrypt data (e.g. RC5), and to generate the MAC (for data integrity support). The MAC is named CBC-MAC as it serves to fragment the cleartext data into several blocks (see Figure 5.1), and to make the encryption of a block xi dependent on the previously encrypted block Hi-1 (xor operation). Likely, the final MAC is the last encrypted block. It depends on all the blocks of the data requiring protection and it constitutes a fingerprint over the data.

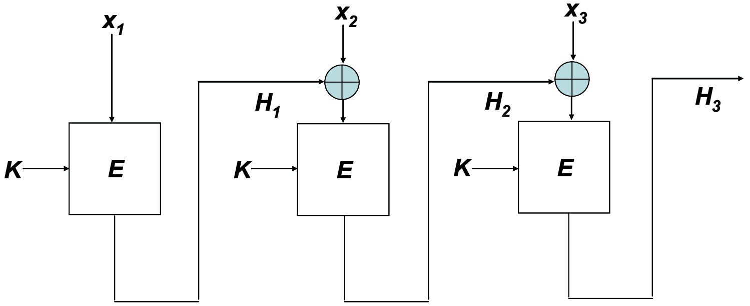


Figure 15.1. *CBC-MAC authentication with XOR operation*

On the other hand, as shown in Figure 15.2 (from [PER 02]), the computing operations performed by the sensors are not the most energy consuming activity, representing only 3 to 4% of the total energy consumed. However, the transmission operations represent more than 95% of the total energy consumed. Therefore, the longer the security information elements are injected into a packet, the more energy consuming the security solution is. In the example of Figure 15.2 [PER 02], if the integrity protection is activated, a 6-byte MAC is appended to the packets and transmission of this extra 6-byte MAC consumes 20% of the battery. Therefore, the lifetime of the sensor is reduced by more than 27% by the mere introduction of the security mechanisms: MAC and freshness.

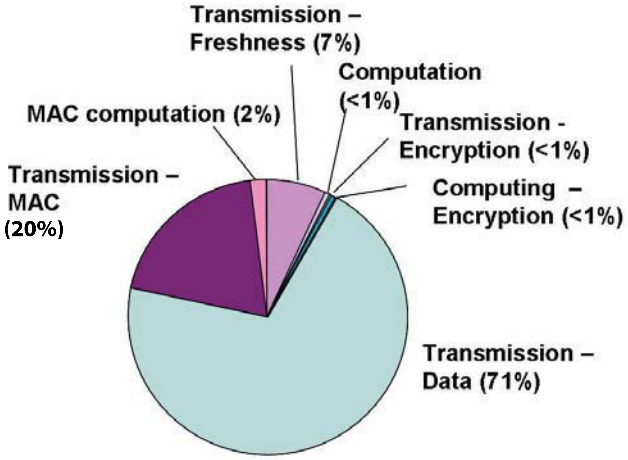


Figure 15.2. *Energy consumed by the SNEP solution (see section 15.3.3)*

Thus, the solutions presented below are analyzed under the following criteria:

– Storage overhead: we must distinguish ROM and RAM memories required for the implementation of security solutions. (Non-volatile) ROM memory is intended to contain the operating system of the sensor (usually TinyOS) and any other codes (programs) associated with security and communication management. RAM is used to contain all the data being processed in the sensor, like temporary or intermediary results (e.g. results of cryptographic operations).

– Energy overhead: previous explanations show that energy consumption is crucial in sensor networks. We must remember that the data transmission is extremely greedy in terms of energy and that any addition of the MAC, sequence number, initialization vector, etc., in data packets is costly in terms of energy and will greatly affect the lifetime of sensors.

– Residual security vulnerabilities: security protocols do not solve all the security problems, especially attacks by battery depletion. Therefore, it is interesting to identify the most important vulnerabilities that will persist, even with the introduction of security services.

– Functionalities: some of the functions typically performed in sensor networks are not compatible with certain security solutions. For example, the aggregation aims to reduce the volume of data transmitted by a sensor, but is only possible if the sensor is able to access to the content of data packets and modify these packets. This condition cannot be satisfied in case confidentiality or integrity protection is activated.

15.3.3. *SNEP security protocol*

The SNEP (Secure Network Encryption Protocol) [PER 02] focuses on the protection of communications between a sensor and a BS or between two sensor nodes of the network. Next, the communications between the BS and a sensor are first described, and then between the sensors.

15.3.3.1. *Prerequisites for the SNEP*

Each node i of the network is expected to initially share a symmetric master key K_i with the BS, which will serve to derive the keys KE_i and KA_i. In addition, each node i shares a counter CPTi with the BS. The use of the counter avoids sending an IV (initialization vector) for each message sent between the BS and the node i; it helps to preserve the energy of the nodes and guarantees the receiver that packets are received in order. Finally, sensors initially do no share any secrets in between.

15.3.3.2. *Communications protected between the BS and sensors*

Assume that a BS is sending a request R to a sensor i. The following message is then issued:

BS → i: R, MAC (KA_i, CPTi|R) (see section 15.3.1 for the notations)

The use of CPTi protects the sensor i against packet replays, because the counter is incremented on both sides at each transmitted packet. The MAC guarantees the destination i of the integrity and origin (from the BS) of the packet.

Assuming that confidentiality is required, the sensor sends the following response Ri:

i → BS: {Ri}<$KE_{i,}$ CPTi>, MAC (KA_i, CPTi|{Ri}<KE_i, CPTi>)

The use of the counter CPTi when performing Ri encryption provides security semantics and makes it more difficult for attackers to perform a brute-force attack by finding a cleartext from a ciphertext. Indeed, integrating the counter into the calculation of the MAC helps the BS to detect any packet replay attacks, as the same text being ciphered at two different times will lead to two different ciphertexts.

If, in addition, the BS makes it necessary to test the freshness of the result, i.e. that the result returned by a sensor comes in response to its own request, then it is possible to integrate a random number N generated by the BS in the request R; the BS then has to test that the returned response Ri takes into account the same number N. Due to the randomness of N, a response issued by a sensor that takes into account N proves that the response has been generated after receiving the request Ri, and the freshness property is thus guaranteed. The exchanges are the following:

BS → i: N, R, MAC (KA$_i$, N|CPTi|R)

i → BS: {Ri}<KE$_i$, CPTi>, MAC (KA$_i$, N|CPTi|{Ri}<KE$_i$, CPTi>)

15.3.3.3. *Communication between sensors with establishment of shared keys*

When two sensors i and j want to communicate securely, it is first necessary to establish a shared master secret between the two sensors. The BS plays the role of a trusted third party by generating a key Kij and by communicating this key securely to each of the sensors.

15.3.3.4. *Costs incurred by the SNEP*

Assessment of the SNEP solution is performed using several criteria:

– Storage overhead: the SNEP requires 1,594 bytes of ROM memory and that code is partly to implement the RC5 encryption algorithm for data encryption (RC5 in counter mode of blocks) and CBC-MAC calculation (MAC in block chaining mode). The introduction of SNEP has a cost of 80 bytes of RAM because of the RC5 algorithm.

– Energy overhead: in order to limit energy consumption, it is important not to increase the size of the packets to be transmitted. RC5 in counter mode of blocks offers such a property since the ciphertext is the same size as the cleartext. The extra cost of security in the SNEP is the transmission of the MAC which increases the size of a packet by 20% and therefore causes extra energy consumption of 20%. With an extra 7% energy overhead for freshness data transmission, freshness is usually optional compared to authentication which is one of the more basic needs.

– Residual security vulnerabilities: (1) because SNEP provides an end-to-end protection from the sender node to the recipient node, there is a risk that intermediate sensors transmit illegitimate packets that will be rejected by the recipient, but that will also deplete the battery of the intermediate sensors. (2) The BS through which most of the communications are go can also be subject to a DoS attack, thus leading to the network being fully paralyzed. (3) The size of the counter CPTi must be large enough to avoid its repetition, otherwise there is a risk that an attacker deduces information about the plaintext from the ciphertext or even discovers the plaintext from the ciphertext.

– Functionalities: the SNEP does not support the protection of data aggregation. First, data authentication is not done hop-by-hop but end-to-end and, as such, the aggregation can be made on erroneous data. Second, if data are encrypted, the aggregation cannot take place.

15.3.4. *µTESLA protocol*

The µTESLA (micro Timed Efficient Streaming Loss-tolerant Authentication) protocol [PER 02] is based on the TESLA protocol [PER 00] developed for ad hoc networks and is one adaptation of it to the limited resources of sensors. µTESLA supports the authentication of the packets broadcasted by the BS on the sensor network.

15.3.4.1. *Prerequisites for µTESLA*

The BS shares a group key K_g with all sensor nodes. However, with the objective to authenticate the origin of packets delivered by the BS and to prevent any malicious node from spoofing the BS while issuing messages, µTESLA introduces an asymmetry. A list of chained keys K_g^n, K_g^{n-1}…; K_g^1, K_g^0 is generated at the very beginning so that $K_g^{k-1}=F(K_g^k)$ where F is an irreversible hashing function. Each sensor is initialized with the key K_g^0 before any deployment of the network. This key K_g^0 is known as the "commitment key".

In addition, each sensor i shares a symmetric master key K_i with the BS, which allows them to authenticate each other (with key KA_i).

15.3.4.2. *Authentication of the origin of the packets and disclosure of the keys*

In µTESLA, the sensors can authenticate the origin of the packets broadcast by the BS. Two steps are necessary, as shown in Figure 15.3, and the time is divided into equal time intervals T. In the first step, the BS broadcasts the packets P1, P2 ... authenticated with the key K_g^k (k is the time interval chosen for transmission); these packets are buffered by the sensors which cannot yet verify their origin because they do not know the key K_g^k; they only know the key K_g^{k-1} and due to the irreversible property of function F, they cannot deduce K_g^k.

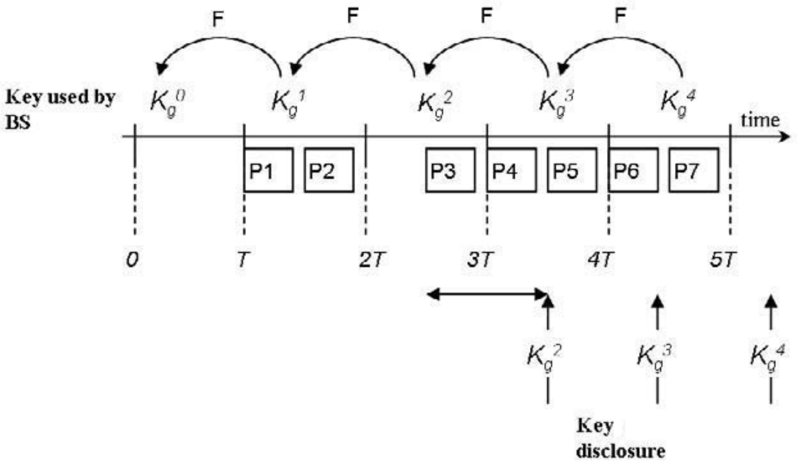


Figure 15.3. *The μTESLA protocol (δ = 1)*

In the second step, the BS broadcasts the key K_g^k in the time interval k+δ (δ≥1); the sensors then check that $K_g^{k-1}=F(K_g^k)$ and that packets previously arrived at time interval k are properly authenticated. Note that the BS should be sure that all the packets have been received by the sensors before disclosing the key, otherwise, a malicious node well positioned on the network might forge packets signed with this key before flooding the network, and sensors would have no way of distinguishing the information from the BS from those forged by the malicious node.

Due to some moves, it may be the case that some keys K_g^k are not received by the sensor for certain periods of time. It is still possible for the sensor to check the authenticity of a key K_g^l from a key K_g^k that it previously received from the BS, by verifying that $K_g^k=F^{l-k}(K_g^l)$. Once the key K_g^l is verified, the sensor can easily check any authenticated packets previously received by recalculating the missing keys.

15.3.4.3. *Communications between sensors*

As for the SNEP, the solutions for protecting the exchanges between sensors are based on the existing trust relationship between the BS and each sensor. One solution is to transmit the data to the BS which has to broadcast them in the network as described above; this solution is energy consuming, as the sensors are highly sought after. A second solution incorporates the principle of a chained list of keys that is here associated with a sensor, and serves to broadcast data in the network. The BS is assumed to know the chained list of keys and does periodically broadcast one of them.

15.3.4.4. *Costs incurred by the μTESLA protocol*

The costs are identical to those mentioned for the SNEP with the following features:

– storage overhead: the μTESLA protocol requires 574 bytes of ROM memory and 120 bytes of RAM;

– energy overhead: this includes both the cost of broadcasting packets by the BS (which is identical to the SNEP) and an extra cost due to the broadcasting of the key.

15.3.5. *TinySec protocol*

The TinySec protocol [KAR 04] of Karlof *et al.* is implemented in the TinyOS kernel (radio layer) and makes the cryptographic operations independent of the applications. It is the role of the application to specify a 2-bit level of protection expected for some data and of TinySec to apply the appropriate protection. To do this, TinySec participates in the scheduling of the processes within the sensor, and prioritizes the processes associated with cryptographic operations when data with protection should be issued.

Like the SNEP, TinySec proposes two security services: authentication only and authentication with confidentiality. Like the SNEP and μTESLA, TinySec defines an end-to-end authentication service (between source and destination) at application level, but additionally it offers a link level authentication between neighboring nodes (both types of authentication are not activated simultaneously). Link level authentication offers the advantage of rapidly detecting any falsified packet and thus avoiding energy consuming retransmissions for intermediate sensors. In addition, it helps to protect the aggregation of data.

TinySec selected the RC5 or Skipjack algorithms in chaining mode (CBC). They are both used for data encryption and CBC-MAC calculation. Data encryption defines an 8-byte initialization vector. To limit the size of the delivered packets, this vector includes several basic fields of the packet like the destination address and the length. Only 4 bytes more are introduced in a packet including a counter that helps to produce different initialization vector values.

15.3.5.1. *Prerequisites for TinySec*

Each sensor is initialized with a secret key that is shared with the BS and is used to derive the encryption and authentication keys for protected exchanges. TinySec also defines a group key shared between all sensors or a subset of sensors or even some symmetric keys shared between two sensors, but it does not specify the modalities for distributing these keys.

15.3.5.2. *Costs induced by the TinySec protocol*

The induced costs are as follows:

– Storage overhead: a TinySec implementation required 728 bytes of RAM and 7,146 bytes of ROM. To ensure the encryption and MAC-CBC calculation, only one of the RC5 or Skipjack encryption algorithms needs to be implemented. Unlike SNEP, TinySec implements an encryption module that is different from the decryption module, and this makes TinySec more ROM-consuming than SNEP.

– Energy overhead: because of the addition of 1 byte (for authentication only) and 5 bytes (for authentication with confidentiality), the time for packet transmission is longer and causes an extra energy consumption of 3% and 10% respectively.

– Residual security vulnerabilities: TinySec implements a hop-by-hop security, and as such enables intermediate sensors to eliminate falsified packets and thus to save their batteries. However, if a node is compromised on the path between source and destination, this node can falsify data and can remain undetected if no end-to-end protection applies.

– Functionalities: due to the protection of packets between neighboring nodes, it is possible to secure the data aggregation.

15.3.6. *Zhu et al. protocol*

This protocol [ZHU 04] defines an authentication service between pairs of non-neighboring nodes (partners) to detect early illegitimate packets and avoid battery depletion attacks. This protocol is still efficient up to *t* compromised nodes in the network.

15.3.6.1. *Prerequisites for the Zhu et al. solution*

Before deployment, each node is initialized with a symmetric key shared with the BS and possesses information that enables the calculation of a local secret to be shared with other nodes.

Some of the sensors can track a phenomenon in one area of interest and are then defined as a cluster. A sensor called a cluster head (CH) is responsible for all the communications with external nodes and the aggregation of data of the cluster. Other sensors are used only as relays with the BS.

15.3.6.2. *Establishment of associations between nodes*

In addition to the establishment of secret keys between neighboring nodes, all nodes on the path between a cluster and the BS and t+1 hops from each other can associate with each other by initiating a shared secret key. The BS initiates the process in two steps.

During the phase-down (from the BS to the cluster), each node discovers the path of the node that is t+1 hops away (towards the BS) and that is known as its upper associated node. To do this, the BS broadcasts a message that is enriched by the identifier of each of the relay nodes. In this way, the nodes can discover their upper associated node and calculate a secret key to be shared with it. In Figure 15.4, the node u4 discovers that u8 is its upper associate, and then it creates a secret key $K_{u8,u4}$ locally.

During the phase-up (from the cluster to the BS), each node finds its t+1-hop lower associated node and calculates the same key as its lower associate did in the previous phase. The associations are thus established.

15.3.6.3. *Protection against falsified packets*

After establishment of the associations, it is possible to protect against forged packet injection and the compromising of nodes, whether these nodes are inside the cluster (including the cluster-head) or on the path between the cluster and the BS.

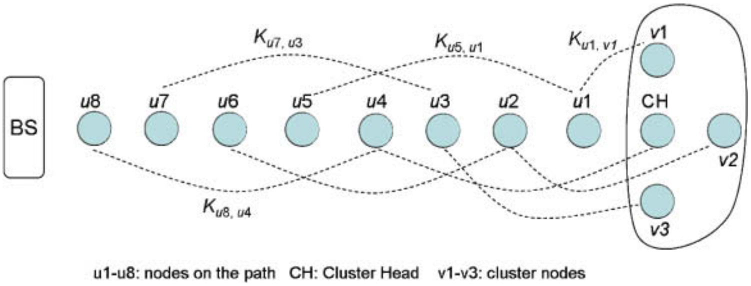


Figure 15.4. *Upper/lower association relations (t = 3)*

To achieve aggregation, each node of the cluster generates a message containing the value E of the observed event and two MACs, one generated with the secret key shared with the BS and the other one processed with the secret key shared with the upper associated node. The CH node verifies that all the nodes returned the value E and then generates the following message:

E, C_i, {CH, v3, v2, v1}, MAC($KA_{u4,CH}$, E), MAC($KA_{u3,v3}$, E),

MAC($KA_{u2,v2}$, E)

$MAC(KA_{u1,v1}, E)$, $\Delta=XOR(MAC(KA_{v1}, E), MAC(KA_{v2},E), MAC(KA_{v3},E), MAC(KA_{CH},E)$

Thus, each node can verify the authenticity of the data received from its lower associated node. In the event of failure, the node destroys the message. Otherwise, it generates another MAC over the value E so the upper associated node can verify the authenticity of the message. This procedure is repeated from node to node up to the BS. The BS then calculates the MAC associated with the nodes of the cluster, verifies that the XOR operation leads to the same Δ and concludes that E is successfully authenticated.

15.3.6.4. *Costs incurred by the Zhu et al. protocol*

The costs for Zhu *et al.* are not quantified, but it is clear that the protocol is costly in terms of computing time, bandwidth and therefore energy for transmission:

– Storage overhead: each node maintains the list of nodes on the path and on average 4 different symmetric keys, including one with its neighbors, one with each of its associate nodes, and one with the BS. Thus, the storage overhead is important.

– Energy overhead: appending t+1 MAC in the message has a very high cost in energy for the nodes of the path that are performing the transmission.

– Residual security vulnerabilities: the advantage of this protocol is in eliminating illegitimate packets at the earliest point on the path.

– Functionalities: this protocol only fits applications that are considering aggregation to be done over the same value that was agreed unanimously by all the nodes of the cluster. It can be an average, a minimum/maximum, etc.

15.3.7. *Summary of security protocols*

The SNEP and µTESLA support end-to-end security (authentication and confidentiality) and BS broadcast source authentication; both of them overconsume energy by 20%. However, the SNEP does not efficiently protect against injection of network packets by an intruder outside the network, as µTESLA does. As a consequence, a false injected packet is not detected en route, but at the recipient, and this might lead to depletion of batteries of the network nodes. TinySec proposes a node-by-node security (data link level authentication and confidentiality), which offers a better protection against false packet injections, and battery depletion attacks, but TinySec increases the energy consumed by 10% and does not protect networks against internal attacks from compromised nodes (corruption of data attacks and identity spoofing by a compromised node). The Zhu *et al.* protocol detects any falsification of data, whether accidental or due to compromised nodes, so a false packet sent even by an authenticated node is rejected at the earliest point in the network, thus preserving the total energy of the network. In return, the protocol introduces a high cost in transmission because of the use of multiple MACs per message. All these solutions have the prerequisite of sharing a secret with at least one entity (BS) and are said to rely on a central trusted entity (BS). Other solutions that are much more easily scalable and more convenient for use are described in Chapter 16.

15.4. Case study: centralized and passive intruder detection

This section presents a WSN's centralized intrusion detection system, a detailed description of which can be found in [TEI 06]. It is said that the system is centralized because surveillance and detection tasks are accomplished at the BS. The system is also non-invasive or passive, in the sense that it does not impose changes on the software or the network element equipment. In the rest of this chapter, we will refer to the system presented here as the CPIDS *(Centralized and Passive Intrusion Detection System)*.

In the CPIDS, the target network is homogenous, flat, symmetric, static and continuous, according to the classification proposed by [RUI 03]. The network has at least one BS and dozens or hundreds of sensor nodes. The hardware of the BS is different from that of the sensor nodes. The BS is typically present in the form of a usual computer with Windows or Unix/Linux operating systems. The sensor nodes are low power and low cost devices; Mica Motes sensor nodes are possible examples [CRO 04]. The nodes are individually identified, which allows the BS to determine which nodes create the information.

15.4.1. *Strategy for intrusion detection*

The centralized and non-intrusive nature of the CPIDS intrusion detection system gives it many advantages. First, the BS has more resources than the sensor nodes, which allows it to implement detection methods similar to those used in traditional IDSs. In addition, IDSs that treat messages arriving at the BS acquire a global vision of network; it is thus possible to make a correlation of events. Finally, the establishment and maintenance of the IDS is very simple, due to the fact that the latter is running only in the BS. Centralized and passive IDSs are of most interest in cases where the sensor nodes are not able to participate directly in the IDS, or when we do not want to modify their configuration.

The CPIDS system observes the messages in transit on the BS, organizes them in an information model, and uses Bayesian networks to compare the observed behavior with the expected behavior. From this comparison, the CPIDS defines the probability of occurrence of an intrusion.

15.4.2. *Information model*

The CPIDS uses an information model based on maps [RUI 03]. These maps are used to represent both the normal and real behaviors of the network.

15.4.2.1. *Information model structure*

The CPIDS proposes an object-oriented information model, represented in Figure 15.5. In this model, the main object is the sensor node, which can provide one or several types of information. For example, it can provide information about the temperature of the environment, its level of energy, etc., according to the type of node and network application.

Different types of maps are obtained from a set of nodes (see Figure 15.5, right side). For example, routing maps are obtained from the routing information collected by the nodes. In addition, each map has a timestamp attribute that indicates the moment when it was built.

Maps of various types are ordered along the axis of time and the sequence obtained is used to represent the behavior of the network (see Figure 15.5, left side). In the CPIDS, the behavior of network is defined by the maps of faults, production, consumption of energy and batteries.

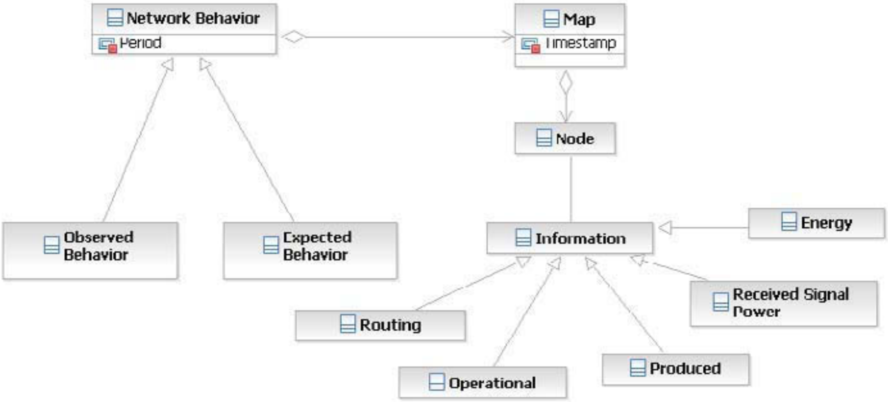


Figure 15.5. *Information model representation*

15.4.2.2. *Map construction*

To detect intruders, the CPIDS uses three types of maps that are implemented in the BS: the production map, the operational state map and the routing map.

The production map helps distinguish the nodes that have "produced" a sensing value – that is, who have made a data acquisition and have forwarded this value to the network – from those who have produced nothing. It relies on the messages received from sensors by the BS from which it extracts the following information: source of message, value of the data collected and frequency of sending the messages.

The operational status map indicates the nodes suitable to produce information. In the map, each sensor node is associated with a probability distribution that indicates its probability to produce information. This information is calculated based on the expected behavior of the node.

The routing map contains information about routes that the nodes use to communicate with the BS. The routing map is built from the information usually contained in the headers of the messages sent by the nodes. In the TinyOS Beaconing protocol, for example, each message includes the identification of the origin node and the identification of the destination node [GAY 03]. The CPIDS uses this information to build the routing map.

15.4.3. *Information analysis strategies*

Maps are combined to indicate whether the observed behavior differs from expected behavior, considering the degree of uncertainty contained in the operational state maps. In this case study, Bayesian networks have been used for the analysis of information [RUS 03].

In the Bayesian network used by the CPIDS (see Figure 15.6), the information of interest is modeled by a variable that may have the following states:

- Production: production or absence of production.
- Route toward the BS: existent or non-existent route.
- Operation: node capable of producing or node unfit to produce.
- Intruder: presence or absence of intruders.

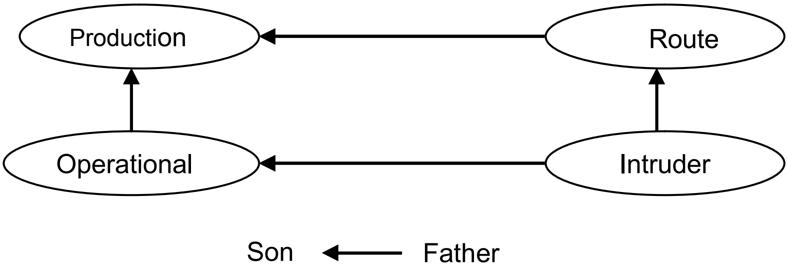


Figure 15.6. *Bayesian network for intrusion detection in the WSN*

The arc that binds the "Operation" state to the "Production" state indicates that the production of a node depends on the operational status of a node. The arc that binds the "Route" state to the "Production" state indicates the influence on a possible existence of an information production route; in effect, without a path between the node and the BS, the data produced may not arrive at the BS and may therefore not be observable. The existence of an intruder, on the other hand, affects the route and the operational state of a node. This, according to the type of attack, can then become non-operational. In this way, indirectly, the intruder affects the production of the node, either by influencing the route or by influencing the operational state of the node.

Prior to the use of Bayesian networks model focusing on intruder detection, it is necessary to establish the values of *a priori* and conditional probabilities. In effect, in the CPIDS, these values have been defined in an arbitrary way and must be graded according to the target network. The initial probability of existence of an intruder in the network has been defined as 50% or 0.5 in a scale from 0 to 1. The probability that a node is operational given the existence of an intruder is defined as 0.2 and the probability that a node is operational without intruders in the network is 0.8. Finally, the conditional probability to have a route between the node and the BS is 0.5 in the event of intrusion and 0.8 otherwise. Once the Bayesian network and the *a priori* and conditional probabilities are defined, the probability of the existence of an intruder may be estimated by analysis of events observed in the network. For example, if we know that there is a route available and that the node has not produced information, even if there was no certainty on the operational state of the node, the probability that there is an intruder is 0.7143. In other words, if we calibrate the probabilities of each variable and if we collect the production and routing maps, we can deduce the presence of an intruder by applying the concepts of Bayesian networks.

15.4.4. *Architecture of the intrusion detection system*

The intrusion detection system is structured into four parts: data source, maps, knowledge base and strategy of intrusion analysis (see Figure 15.7).

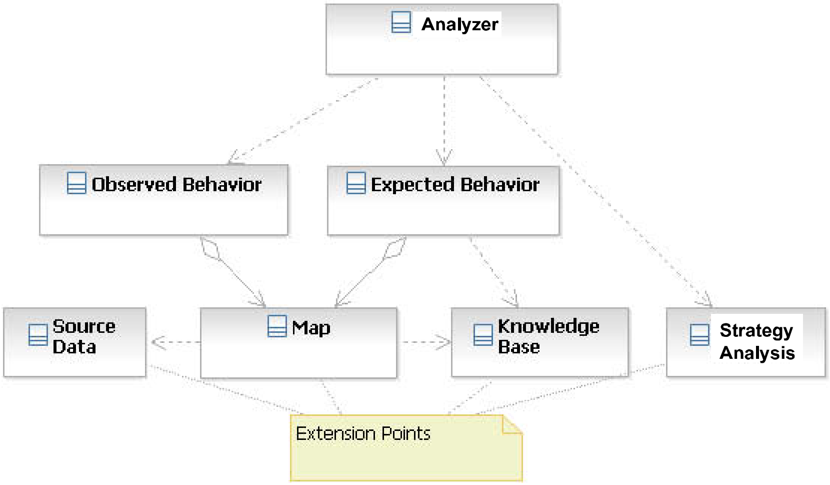


Figure 15.7. *Logical view of the IDS architecture (in UML)*

The data are obtained from the BS and come from log files or received messages. The data are organized in time stamped maps according to the type of information of interest. The maps are grouped in order to define the observed behavior over a certain period of time.

The system uses an abstraction that represents the knowledge base. The knowledge base houses all the knowledge that defines the normal behavior of the network, by considering the selected maps. The whole set of knowledge may be formed by axioms, assertions and models of prediction, such as the pattern of energy consumption, battery model, radio link model, sensing model and routing model. To define the expected behavior for a period of time, the CPIDS observes the information coming to the BS and compares it with the data contained in the knowledge base.

The strategy of analysis is another axis of the architecture, which states that the strategy can be reviewed according to the target network and the available information. For example, the CPIDS uses Bayesian networks to compare the expected behavior with the observed behavior.

15.4.5. *An IDS prototype*

A WSN has been simulated using various scenarios of intrusion in order to assess the effectiveness of the system.

In the IDS prototype built in Java, the maps and the knowledge base have been defined and a probability of intrusion has been calculated. In addition, an anomaly analyzer was built, which uses the information contained in the maps to calculate the conditional probabilities of the knowledge base.

For example, assuming that the production map indicates that a node has not produced, the routing map indicates the existence of a path between the node and the BS, and the operational map indicates a high probability that the node is operational; the anomaly analyzer would use this information as parameters of a conditional probability rule contained in the knowledge base and would calculate a high probability of intrusion.

15.4.5.1. *Experiments*

To quantitatively assess the solution, the network and the attacks against it were simulated using the simulator presented in [SIL 05]. This program simulates nodes that generate data continuously and also some of the attacks described in [KAR 03]. A program has been developed by us to analyze false negatives generated by the IDS prototype. The program summarizes the results of each experiment by calculating the average and the standard deviation. The program compares the output of the IDS with the release of the simulator to check the number of false negatives obtained. For each non-detected attack, the program counts a false negative.

The effectiveness of the IDS was tested in a fault-free network as proposed by [SIL 05]. It is a flat and static network with 100 sensor nodes randomly distributed in a grid of 20x20 square meters; data messages are sent at regular intervals, after every set of 40 iterations. Each iteration corresponds to a simulation cycle. The nodes are individually identified and have fixed radio coverage. Three types of nodes were used: common, BS and intruder nodes.

The experiments were repeated at least 35 times each and the average values were calculated. The simulations were carried out in a virtual time corresponding to 4,000 iterations and by making the attack rate vary from 0 to 100%, by intervals of 5%. The attack rate indicates the frequency at which the intruder performs its attacks. A rate of 40%, for example, indicates that an intrusion is simulated at 40% of the iterations. The experiments were carried out by simulating the blackhole, selective forwarding, negligence, wormhole and jamming attacks. Figure 15.8 illustrates the results obtained in the attempt to detect each one of these attacks. For the selective forwarding attack, experiments have been carried out by keeping the attack rate fixed at 70%. The probability of deleted messages by the attacker, in each attack, has varied from 0 to 100%.

15.4.5.2. *Result analysis*

The effectiveness of the detection is measured by the detection rate and quantity of false alarms generated by the CPIDS. When an attack occurs during a time interval, it is checked whether the attack has been correctly discovered; if yes, it is a success, otherwise it is a failure (false negative). The detection rate is determined by the ratio between the quantity of false negatives and the total number of attacks carried out during the simulation. If an attack is detected in case of the absence of intrusion, then a false alarm (false positive) is recognized.

As illustrated in Figure 15.8, the detection rate remained above 88% for four of the five attacks analyzed. Only the wormhole attack gave a detection rate above 80% and less than 88%. The maximum number of false alarms per experiment has varied from 69 to 405 for a total of 4,000 events analyzed, as shown in Table 15.1.

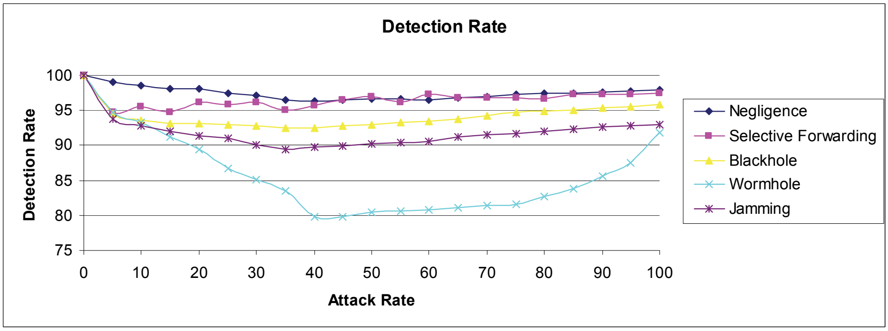


Figure 15.8. *Detection rate according to the intensity and type of attack*

The results are satisfactory compared to those presented in [LIP 00] and [AXIS 99] which, for conventional IDS systems, obtained detection rate results between 63 to 93% according to the quantity of false alarms per day. The results are also satisfactory if compared with the results obtained by [SIL 05], where the detection rate remained close to or above 75%.

Experiment	**False alarms**
Negligence	75
Selective Forwarding	69
Blackhole	405
Wormhole	181
Jamming	374

Table 15.1. *Quantity of false alarms (false positives) compared to 4,000 events analyzed*

15.5. Case study: decentralized intrusion detection

This section presents a decentralized intrusion detection system that takes into account the restrictions and peculiarities of WSNs. This IDS is based on changes in behavior of the network obtained from analysis of events detected by the monitor node where the IDS program is installed. This section includes an assessment of the efficiency and the accuracy of the IDS to detect seven types of attacks. It also includes an assessment of costs for the use of the IDS in terms of energy consumption. It also presents the sketch of a methodology to build IDSs specific to a target WSN (with its own applications), as well as the development of a simplified simulator capable of simulating the main characteristics of a WSN and IDS proposed. The details of this system can be obtained in [SIL 05].

The distributed intrusion detection systems are robust and scalable. As the monitors (nodes that have an IDS inside) spread over the network, it is more difficult for an intruder to hide itself. In addition, as the IDS is closer to the intruder, that is to say, one hop distance in the present case, the detection of attacks is fast.

The IDS was developed applying a specification-based technique [BAL 03], [TSE 03], [KB 97] because the configuration of WSNs vary greatly according to the applications that they intend to be run. The solution provides the distribution of the IDSs over the network and its installation in nodes called monitors. Information gathering and processing are also made in a distributed way, based primarily on listening to all network exchanged messages by monitor nodes (promiscuous listening).

The developments were carried out by trying to use the minimum account of memory and processing possible by storing only the information useful to the application of pre-defined rules. In addition to the control of energy consumption, these choices make it possible to obtain good performance and real-time detection.

15.5.1. *Distributed IDS modeling for different WSN configurations*

A solution has been designed to be able to adapt the IDS to a variety of WSNs and different applications. The general idea is to define possible rules from the knowledge of the characteristics of a specific WSN, and to choose the rules that may be implemented with the best cost from the network available data.

To acquire knowledge about the target WSN, it is necessary that its designer gives the details about its characteristics and behavior. For example, Table 15.2 shows the characteristics of a specific network defined by its designer and rules defined from these characteristics.

Once these rules are defined, the type of available data in the network and the cost of its implementation must be verified. For example, if a message can be clearly identified, this makes it possible to apply Rule 3 – Repetition. If the nodes do not have information about the identity of their neighbors, a supplementary implementation must be made to enable the application of Rule 4 – Coverage. However, the cost of this implementation may make the rule inapplicable.

Characteristics defined by the network designer	Rules defined from the characteristics
Characteristic 1: Multihop message distribution	**Rule 1 – Retransmission**: If a node receives a message not aimed at it, is must retransmit the message
Characteristic 2: No fusion or data aggregation before transmission	**Rule 2 – Integrity**: The message received by a common node has to be forwarded without modifications
Characteristic 3: No provision for acknowledgement or message retransmission mechanisms	**Rule 3 – Repetition**: Nodes cannot retransmit the same message
Characteristic 4: Limited node radio coverage	**Rule 4 – Coverage**: A node is able to receive messages only from neighborhood nodes (nodes under its radio coverage area)
Characteristic 5: It is possible to estimate the maximum time required for a node to retransmit a message	**Rule 5 – Delay**: Nodes have to retransmit a received message in a previously defined maximum time interval
Characteristic 6: It is possible to estimate the number of expected collisions in the network	**Rule 6 – Jamming**: The number of observed collisions must be less than or equal to the maximum number of network expected collisions

Table 15.2. *Network characteristics and defined rules*

15.5.2. *Applied algorithm*

Once the choice of rules to be used by the IDSs has been made, the IDSs can be installed in a distributed way among the network nodes, which begin to play the role of monitors. The algorithm used by the monitor consists of the following three phases:

– Phase 1 – data acquisition: the monitor nodes listen to the network and collect messages in transit to analyze them later. Only those message fields used by the rules are stored and messages on which it is impossible to implement rules are ignored. This first treatment makes it possible to decrease the space occupied in the memory and to reduce the processing time of the monitor node. Messages are stored in a vector until it is completely fulfilled. At that stage, phase 2 is launched. For economy of energy purposes, listening is disabled in phases 2 and 3. Consequently, the monitor loses a few messages and may cease to detect some attacks. Despite this, the harm is considered to be relatively low: in effect, monitor nodes are not synchronized, and therefore the listening is not deactivated among all monitors at the same time. Thus, while a monitor will have its listening disabled, a second monitor can detect the attack. In addition, the attack will probably take longer than the time during which the listening is off and thus the monitor will still have the time to detect this attack. Here a compromise between economy of energy and detection effectiveness the must be found.

– Phase 2 – application of the rules: in this phase, thanks to the data stored and the application of rules installed in the IDS, suspicious activities are identified. In cases where the data stored and associated with a message does not match some of the rules, an error occurs and the message is abandoned. No other rule is then applied to the message. This makes sense, since a message not complying with one of the rules is an indicator of an abnormal behavior in the network. This strategy has been adopted to save monitor node processing and consequently to save energy, but it also reduces the detection time since messages are processed more quickly. A compromise is to be found between the precision in detection, the processing cost, and the execution time. The sequence of the applied rules is chosen in such a way that the most simple rules are tested first. In case of error in the simplest test, the more complex tests will not be executed. Once again, the strategy has been chosen because of its gain in processing and therefore in energy.

– Phase 3 – detection of indicators: in this phase, the faults that occurred in phase 2 are analyzed and compared with the model of natural faults of the network, in case they are defined. If the produced fault corresponds to an abnormal behavior included in the model, an alarm indicating an intrusion is generated.

Figure 15.9 shows the architecture of a monitor node. In addition to the functions of the monitor, the node still performs its regular duties, such as data acquisition, sending of messages and retransmission. The IDS installed on the node has three software modules, each one responsible for one of the phases described above.

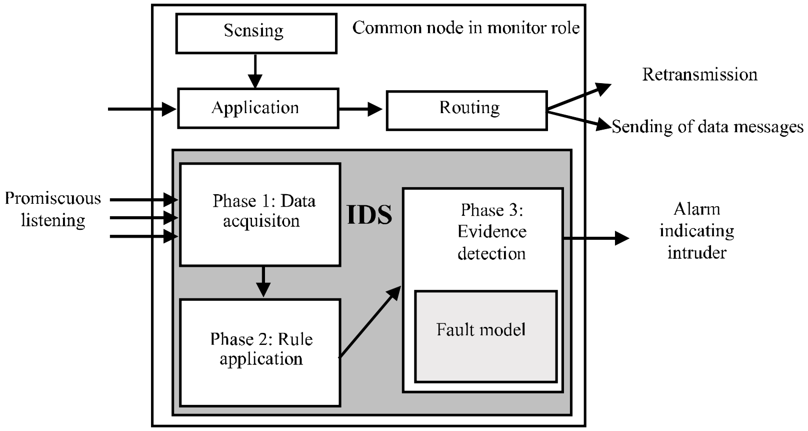


Figure 15.9. *Monitor node architecture*

15.5.3. *Prototype used for the validation*

For the sake of validation of the solution, a flat and fixed network [RUI 03] was simulated with a random distribution of nodes. The nodes are uniquely identified and have fixed radio coverage. The network includes 100 randomly distributed nodes, as shown in Figure 15.6, and the data messages are sent at regular time intervals. The set of characteristics presented in [SIL 05] has helped to define all rules identified in section 15.5.1 and used by the IDSs located in the monitors. 28 monitors were distributed in order to cover all the common nodes of the network. Most of nodes are therefore covered by more than one monitor and each node has its own vision of the network.

15.5.4. *The simulator*

To validate the system, a simulator has been developed. The simulator was implemented in the C language with three objectives: performance, modularity and extensibility. A model of discrete events has been implemented. In this model, the objects of analysis, that is, the BS, common nodes, monitors and intruders, change the state at the time of the occurrence of certain events, for example, reception, sending of a message, data acquisition and the achievement of an attack. The network-sensing events are generated at random and nodes are not synchronized in their attempt to approximate the behavior of the simulator to that of a real network. More details on the simulator can be found in [MAR 05].

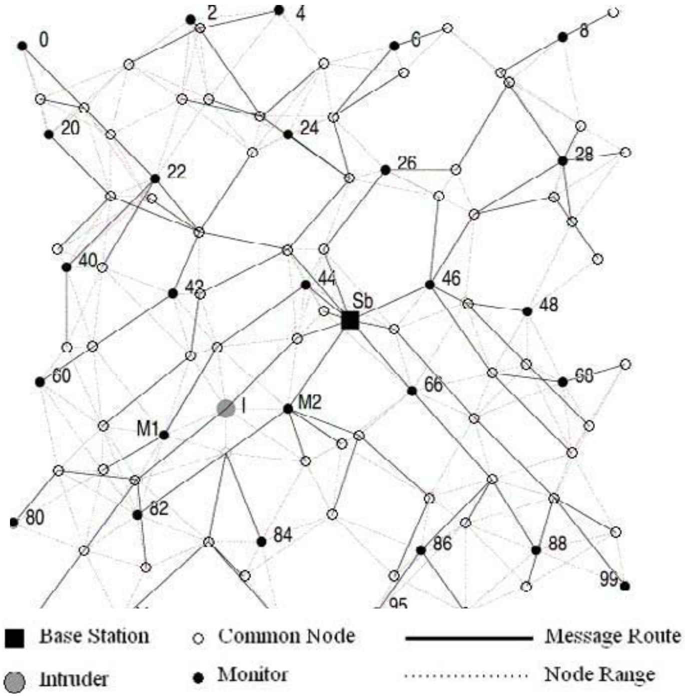


Figure 15.10. *Routing tree of the simulated WSN*

15.5.5. *Experiments*

The objective of experiments is to verify the effectiveness of the system proposed in situations in which the intruder attacks in a sporadic or continuous way. This is done by making the rate of occurrence of the attack vary. It was expected that the possibilities of detecting attacks by the monitor were directly proportional to the frequency of attacks. In addition, it was intended that a better cost-benefit ratio would be obtained thanks to the storage in the monitor of detection data. From the point of view of the monitor, the time is divided into segments and every segment corresponds to the time for filling the array, since the moment when the array is empty up to the array is fulfilled, as messages are being listened to. When the array is completely fulfilled, the segment is completed and the stored message processing can start. This corresponds to the end of phase 1 of the algorithm defined in section 15.5.2. The dimension of the array in fact defines the dimension of a segment of time during which the node will be listening, and therefore the amount of messages that will be collected in order to seek traces of intruders. As has already been said, there is a compromise between the cost of storage and the effectiveness of the detection. The smaller the size of the array and consequently the lower the cost of storage, the shorter the segment size and the greater the losses of message sequences, which implies less efficient detection.

In order to assess this compromise, three different dimensions of array for each of the attacks have been used. To define these three dimensions, we have conducted experiments with real sensor nodes under the Sensornet Project (www.sensornet.dcc.ufmg.br). We have verified that a 100 position array is a reasonable upper limit since more than 80% of the available RAM is already filled. We have defined two additional intermediate dimensions for the array: 30 and 60 positions. We have analyzed the effectiveness of monitors M1 and M2, shown in Figure 15.10, to detect the following attacks, executed by the intruder: data modification, message delay, blackhole, jamming, selective forwarding, repetition and wormhole. For each of these attacks, we have varied the rate of occurrence of the attack from 1%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% to 100% of the time. A 40% rate of occurrence for example means that the intruder made attacks for 40 simulator iterations and acted as a normal node for the 60 other iterations. The relationship between the rules used and the potentially detected attacks is shown in Table 15.3.

Attacks	Associated rules
Selective Forwarding (SF) and Blackhole (B)	Rule 1 – Retransmission
Data Modification (DM)	Rule 2 – Integrity
Repetition (R)	Rule 3 – Repetition
Wormhole (W)	Rule 4 – Coverage
Message Delay (MD)	Rule 5 – Delay
Jamming (J)	Rule 6 – Jamming

Table 15.3. *Relationship between rules and attacks*

The effectiveness of detection is measured based on the time segments defined by the monitor. If an attack occurs in the intervening period of time corresponding to a segment, we check if the attack had been detected correctly by the node monitor in the same interval. In the event of detection, a success is recognized; in the contrary case, a failure is recognized (false negative). If no attack occurs, but an intrusion is detected, or if a suspect is unjustly accused, a false positive is recognized. Natural faults in the experiments have not been considered. All the possible cases have been executed 33 times, for 2,000 iterations.

15.5.6. *Results*

Effectiveness, precision and consumption of energy are the metrics used to assess the proposed IDS. On average, the IDS presents a good efficiency, remaining above 70% of detection for five out of seven attacks, even when these attacks were sporadic (up to 10% of the time) and the monitor used the less efficient array size (30 positions), as shown in Table 15.4. The advantages of using low size arrays are economy of memory space and greater processing speed for each time segment.

Array size	Attack (acronyms defined in Table 15.3)	Effectiveness		
		Attack occurrence: 10% of time	Attack occurrence: 40% of time	Attack occurrence: 80% of time
30	SF, B, DM and R	Between 72% and 81%	Between 83% and 86%	Between 95% and 98%
60	SF, B, DM and R	Between 82% and 88%	Between 93% and 95%	100%
100	SF, B, DM and R	Between 94% and 97%	100%	100%
30, 60 and 100	Wormhole	100%	100%	100%
30	MD	25.8%	72.0%	78.5%
60	MD	30.0%	80.6%	99.6%
100	MD	33.9%	100%	100%
30	J	41.4 %	92.2%	100%
60	J	42.1%	100%	100%
100	J	55.7%	100%	100%

Table 15.4. *Detection effectiveness*

The detection of wormhole attacks has reached 100% efficiency in all cases. Thus, for Rule 4 – Coverage, it is sufficient that the monitor receives a message from a non-neighbor node to indicate the existence of an intruder. The detection of delay message attacks has presented a lower effectiveness mainly for small arrays associated with sporadic attacks. This is due to the fact that the corresponding rule would assume that both messages (sent and delayed) are in the same array; if unfortunately they were in two different arrays, the fault would anyway be detected as a blackhole.

The detection of jamming attacks is shown to be hardly dependent on the array size since the jamming rule does not consider the comparison with future messages. A low efficiency occurs when the attack is carried out outside the interval of promiscuous listening, which is more likely to occur when the frequency of attack is low.

Repetition, jamming and delay attacks present false positives in relation to the attack and the accused intruder, as is shown in Table 15.5.

Attack	False positive
Repetition	The monitor M2 (see Figure 15.10) accuses node P of being an intruder. The false positive occurs because there is no processing to remove repeated messages and node P just forwards repeated messages it receives.
Jamming	Some monitors accuse innocent and intruder neighbor nodes of performing attacks such as blackhole and negligence. This happens because the accused nodes do not succeed in sending their messages or transmitting the messages they receive, because of the jamming attack.
Message delay	Monitors confuse delay with blackhole attacks when the original and delayed messages are not listened to at the same time segment.

Table 15.5. *False positives*

Although not correctly detecting the attacks (imprecision), monitors have detected abnormal behavior of the network caused by an ongoing attack. This information is useful because it identifies collateral effects caused by specific attacks and indicates the affected nodes and their resulting behavior, which may look like other attacks (false positive). Detailed results of this study are available in [SIL 05].

15.5.6.1. *Energy consumption*

We consider the energy consumption caused by listening, reception and transmission of messages made by each network node. Messages of 36 byte size (www.tinyos.net) were used, as well as a transmission rate equal to 62.4 µs/bit [SHN 04]. Energy consumption in each of the situations (transmission, reception and listening) was calculated by considering the value of 7.3 milliamps for the current intensity that passes in the node when it receives messages, and 21.48 milliamps for the current that passes in the node when it transmits messages with a greater power.

The common nodes presented the same energy consumption in experiments with or without monitors. The energy consumption of the monitors varied drastically according to their positioning in the routing tree, with energy consumption running from 28% to 500%. The energy consumption of these nodes is directly related to the number of messages the nodes are exposed to because of promiscuous listening. The higher the network load in the neighboring region of the monitor node, the more it will listen to messages and will consume energy.

If we consider the increase in total energy consumption of the network, there is an increase of 125%. Even with such a percentage of increase in the consumption of energy, the lifetime of the network does not diminish significantly. The distribution of nodes in the form of a tree is responsible for this over-consumption of energy by some nodes, regardless of the deployment of monitors. One of the common nodes (the 34) not implementing IDS functions has consumed more energy than monitor nodes with enabled IDS functions, as illustrated by Figure 15.11. The monitor nodes are part of the IDS and are identified by a dotted line.

This result varies considerably according to the scenario and the protocols used in the target network. For example, when considering a WSN where the protocols better disseminate messages among nodes of the network, the energy consumption of common nodes will be better distributed as well as the energy consumption of monitor nodes.

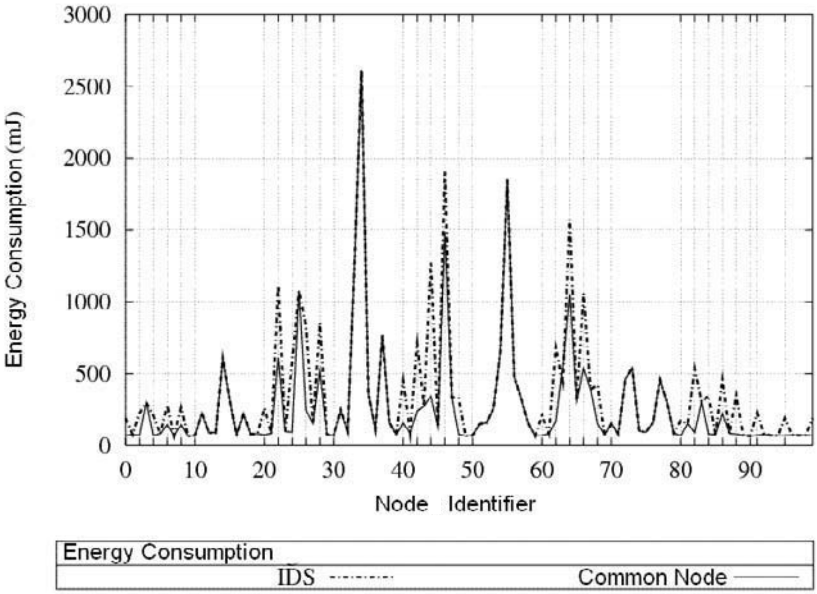


Figure 15.11. *Energy consumption by monitor and common nodes*

15.6. Case study: intrusion tolerance with multiple routes

This section introduces a strategy that provides wireless sensor networks the capabilities of tolerance to intrusion and that, consequently, increases their resilience. The strategy consists of creating alternative routes in routing functions, which also contributes to assisting in intruder detection processes. The TinyOS beaconing routing algorithm [HIL 00] has been modified so that each node uses two paths to send its information to the BS. In the case of an intruder being present in one of the paths and blocking the traffic, the alternative path ensures that part of such information will be transmitted over it. This section describes the modified algorithm, as well as an assessment of its performance in terms of energy and effectiveness (resistance to DoS, intruder detection). The performance was verified by simulation and the results showed good efficiency, even with a large number of intruders.

Multiple paths are redundant paths in routing and are alternatively used without information replication. The alternation of routes (or paths) increases the intruder tolerance of the network, because it offers another option for routing. If there is an intruder positioned in one of the paths, the alternative path still makes the delivery of packets possible. In addition, by analyzing the received packets, it is possible to discover the paths that do not correctly deliver packets and cause problems for the routing. Route switching has been chosen in order to maintain the consumption of energy close to that verified with single route strategies.

The route switching mechanism used in this case study contributes to increase the resilience of the network and still allows effective detection of intruder nodes. In the simulations carried out, it was observed that the intrusion detection algorithm presents high effectiveness in the presence of a few intruders, and also succeeded in identifying a significant number of the intruders when they are in large numbers.

The kinds of network used in this case study are comparable to the networks of the cases covered in the two preceding sections.

15.6.1. *Alternative routes*

Multiple routes may be disjoint, when they do not share any node, or may be interlaced, when they do contain shared nodes [GAN 01]. Disjoint routes are more tolerant to faults and intrusion. Interlaced routes present lower creation and maintenance costs in terms of energy consumption; a fault in a shared node, however, may make all existing paths unusable. Several routing algorithms have already been proposed for WSNs in which the mechanisms for creating and handling routes are justified by the type of network and application. For each protocol, there are many ways to create multiple routes, but in this case, it is restricted to the IP or Information Propagation protocol [BAR 96]. This algorithm, also known as TinyOS beaconing, is used in the TinyOS operating system.

15.6.1.1. *Alternative routes algorithm*

TinyOS beaconing is a routing algorithm based on a packet called a beacon broadcast by the BS. After receiving a beacon, each node determines the neighbor node that will serve as a relay to the BS. To create multiple routes, a node must establish paths other than the one established with the first beacon. Thus, the second path is established through the neighbor node that has delivered the second beacon. This process does not guarantee the creation of disjoint paths but ensures two alternative paths for each node. The route created from the first beacon is defined as the standard route and the route created from the second beacon is defined as the alternative route. Once established as the alternative routes, each packet sent to the BS indicates the route used (standard or alternative).

The routing algorithm performs the sending of messages in different ways according to whether the node is the originator of the message or the forwarding node. An origin node sends messages once through the standard route and once through the alternative route. A forwarding node always sends messages through the standard path. Two situations have motivated this strategy. If a forwarded message can take alternative routes, one at each hop, the path from a node up to the BS would present many possibilities. Nevertheless, as it is necessary to register the path followed by the packet, this strategy would become very costly because every hop should be registered. In addition, loops might occur, which could increase the cost of routing and even prevent the delivery of some packets. While alternative and standard routes are acyclic, its overlapping may generate cyclic paths.

Figure 15.12 illustrates the creation of alternative routes in a small network. Darker arrows indicate standard routes and lighter arrows alternative routes.

A mechanism to decide on the path to use each time a packet has to be forwarded must be defined; it must be unpredictable for the enemy but known by the BS. The choice of the path must take into account the detection and the isolation of the intruder: the number of messages that pass by one of the two routes must be comparable to that of the other route as well; the BS as well as the node must know *a priori* the path used for each message. Thus, when a message does not arrive, the BS knows the path causing the problem.

15.6.1.2. *Intrusion detection algorithm*

The intrusion detection algorithm treats all the packets received and uses loss packets information to identify the possible intruder. To identify the intruder, the BS must know the network topology. To this end, each node must send a message to the BS indicating which of its neighbors are used for each of the routes. These messages indicate the nodes responsible for the standard and alternative routes. From these messages, the BS is able to generate a network connectivity map to be used by the intrusion detection algorithm.

The intrusion detection algorithm is executed recursively. The initialization of the algorithm takes into account the losses in each route for each node. The process starts at the BS. The analysis of a certain node consists of checking the losses that may have happen in all nodes that depend on it to forward packets. If the losses of a route are much higher than the losses of the other route, the node is marked as an intruder. The recursive step consists of analyzing all nodes that use the node under consideration as part of the standard route. Figure 15.13 shows the defined recursive algorithm.

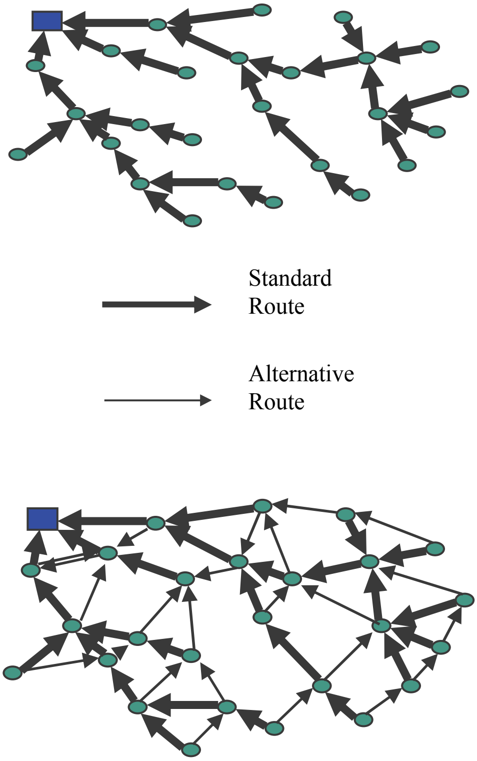


Figure 15.12. *Alternative route formation*

```
Intruder Detection (Node X, Intruder Node, Intruder Score)
  1.   For each node I, neighbor of X that uses this node as
  standard route:

    a.  If Standard Packets (I) << Alternative Packets (I) then

      i.  If Intruder Score = 0 then

            Intruder ← X;
     ii.  Increments Intruder Score;

    b.  If Standard Packets (I) = Alternative Packets (I) = 0 then

      i.  If Intruder Score ≠ 0

            Increments Intruder Score;
     ii.  Otherwise

            Mark I as Failure
  2.   For each node I, neighbor of X that uses this node as
  alternate route

    a.  If Alternative Packets(I) << Standard Packets (I) then

      i.  If Intruder Score = 0 then

            Intruder ← X;
```

Figure 15.13. *Recursive algorithm for intrusion detection*

To improve the precision of the intruder identification process, each node identified as a probable intruder is initialized with a weight that is increased each time the node is suspected of being an intruder. In particular, each time significant losses are noticeable on a route through this intruder node, the weight of the node is increased. The weight of a node suspected of being an intruder represents the extent of its attack. The wider the attack, the bigger its weight.

The algorithm comprises three parts: calculation of losses in neighbor nodes that use the suspect as a standard path; calculation of the losses in the neighbors that use the suspect node as an alternative route; execution of the same recursive function for the neighbor nodes who use the node as a standard route. The result of the execution of the algorithm indicates the possible intruder nodes.

An example of intruder detection is shown in Figure 15.14. The node marked with an X achieves an attack and does not relay the messages sent by the nodes marked 1, 2, 3 and 4. The nodes that depend on the node marked by an X to perform their routing are inside the region delimited by a hatched line. The intrusion detection algorithm starts in the BS. When the X node is analyzed by the detection algorithm, losses are observed for each of the routes of each of the nodes numbered 1 to 4. The losses of nodes 1 and 2 occur in the same proportion in the two routes since both nodes depend on the X node and the presence of the intruder may not be inferred by this analysis. Nodes 3 and 4, by contrast, have greater losses in standard routes, which depend on the X node, than those in alternative routes, which does not depend on the X node. In the algorithm, during the analysis of the losses of neighbors who use the X node as a standard route, excessive losses of nodes 3 and 4 will be observed. Finally, the X node will be identified as a possible intruder, having its weight increased by two units.

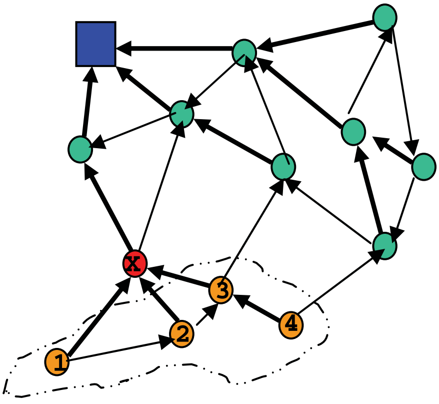


Figure 15.14. *Example of intruder detection by the algorithm*

15.6.2. *Validation of the solution*

Three aspects have been analyzed in the evaluation of this case: performance, functionality and scalability. The performance is based on energy consumption, one of the most important metrics of WSNs. The functionality is evaluated with respect to the reduction of the number of silenced nodes, as well as the result of the detection of intruders. The scalability is assessed by the execution of the algorithm with different data loads varying from some tens to a few thousand nodes and using two types of distributions. The blackhole attack has been used, although other attacks usually involved in routing could have been selected.

To evaluate these aspects, three sets of simulations were carried out using the simulator presented in [MAR 05]:

– energy consumption: the increase in energy consumption caused by alternative routes;

– tolerance to intrusion: the effectiveness of alternative routes mechanism to reduce the number of silenced nodes by some DoS attacks;

– intrusion detection: the effectiveness of the intrusion detection mechanism.

Two types of distributions have been used: random and random band. This last distribution simulates the launch of sensors by an aircraft. The area of the experiment was divided into ten bands (or ranges) and each one receives a tenth of the number of nodes. A range represents the area covered by a straight flight of an aircraft. The simulated networks contain 40 to 1,025 nodes, as done in [OLI 05]. Various scenarios make it possible to verify of the scalability of the solution presented here. In the experiments, nodes produce data at fixed time intervals and send them to the BS.

15.6.2.1. *Energy consumption*

The result obtained in the experiments is the average energy used by the nodes. Table 15.6 shows the average consumption for single routes and the average consumption for alternative routes.

The use of alternative routes has caused an increase of energy consumption of between 3 and 15%. This last value has been registered just once, for a very small network with 40 nodes. This increase in consumption is observed for packets that pass through alternative routes, corresponding to the second best route in terms of distance. When all packets pass through the standard route, they use the better route and thus the consumption is less. We can observe that the variation of the consumption increases when the number of nodes goes lower. In networks with a larger number of nodes, the additional energy consumed by alternative routes is less significant. This is explained by the relative difference in length between alternative and standard routes, which are lower for large networks in relation to small networks.

Distribution	**Number of nodes**	**Average consumption without alternative route (mJ)**	**Average consumption with alternative route (mJ)**	**Percentage of increase**
Random	1,025	501.54	519.45	57%
Random short	400	346.17	360.68	4.19%
Random too short	40	145.03	168.70	16.32%
Bands	1,020	486.28	504.78	3.80%
Bands short	400	330.96	346.46	4.68%

Table 15.6. *Consumption increase due to alternative routes*

15.6.2.2. *Level of intrusion tolerance*

The second group of simulations were intended to verify the effectiveness of switching routes in relation to the increase of tolerance to intrusion. To check the scalability of the presented solution, several experiments were carried out with different quantities of common and intruder nodes.

The intruder nodes were chosen randomly and represent 10 or 30% of total of nodes. Attacks in less than 10% of network nodes have a very low impact and above than 30% have a very significant impact. The latter bring the majority of nodes to silence. The quantity of packets sent by each node to the BS has been registered. At the end, it is therefore enough to count the number of nodes reduced to silence. The results are presented in Figures 15.15 and 15.16. Node distributions are shown in Table 15.6; for each distribution, the bar placed above corresponds to the option with an alternative route and the one below without an alternative route.

The presented results show a smaller number of silenced nodes when alternative routes were used. The total network production remains at the same levels but the data come from a larger number of nodes. For a better monitoring of events, it is extremely important that the largest number of sensor nodes send responses.

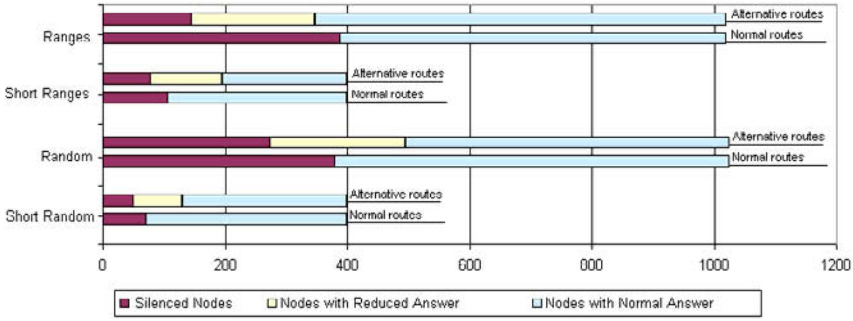


Figure 15.15. *Response of the network in the presence of 10% of intruders in the routing with and without an alternative route*

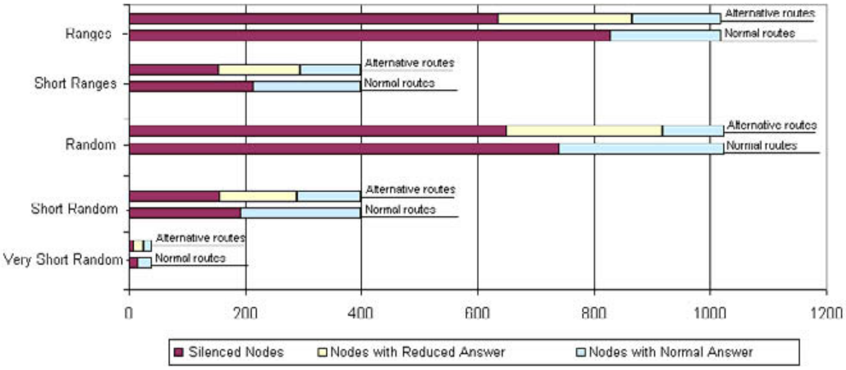


Figure 15.16. *Response of the network in presence of 30% of intruders in the routing with and without an alternative route*

15.6.2.3. *Effectiveness of the intrusion detection solution*

The effectiveness of the solution to detect intruders has also been evaluated. Intrusion detection is more effective when the intruder is positioned at the heart of the network, where routing tasks are more important. Intruders who do not keep this role in routing are not detected since their presence has no effect on the network.

The second group of simulations focused on verifying the effectiveness of the detection algorithm in networks with an intruder only. Tens of simulations were performed by relying on the same distributions previously submitted and on an randomly positioned intruder. For most of the simulations, the intruder did not participate actively in routing and therefore it remained undetected. However, when the intruder participated in the routing, it was detected in all cases.

The result of these comments is that intrusion detection has been particularly effective when the intruders are few in number, due to the fact that they are not present on the same route, that is, that they are not involved in the production of the same set of nodes. However, when the number of intruders is large, this is no longer the case. As the actions of different intruders may interfere with the production of the same set of nodes, their detection becomes more difficult.

The third group of simulations has been performed with the purpose of verifying the effectiveness of the intrusion detection algorithm in the presence of many intruders. The quantities of intruders simulated are 10% and 30% of the total of nodes. The data for simulation are the same as those described in Table 15.6. Table 15.7 and Figures 15.17 and 15.18 show the obtained results.

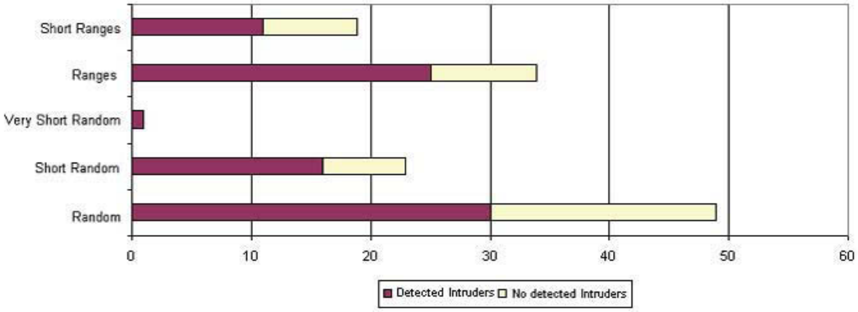


Figure 15.17. *Intrusion detection with 10% of intruder nodes*

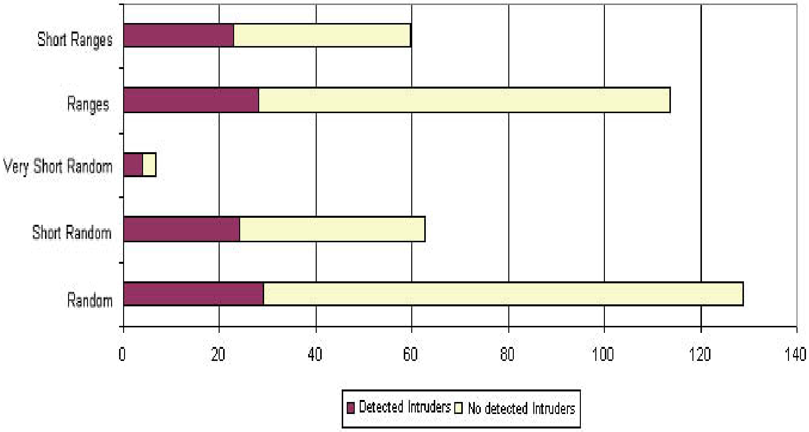


Figure 15.18. *Intrusion detection with 30% of intruder nodes*

Test	Number of nodes	Number of intruders in the routing	Number of detected intruders
Random 10	1,024	49	30
Random 30	1,024	129	29
Short random 10	399	23	16
Short random 30	399	63	24
Very short random 10	40	1	1
Very short random 30	40	7	4
Ranges 10	1,024	34	25
Ranges 30	1,024	114	28
Short ranges 10	399	19	11
Short ranges 30	399	60	23

Table 15.7. *Intrusion detection with a large number of intruder nodes*

No cases of false positives have been observed. The revocation of the discovered intruders, the subsequent restoration of routes and a new execution of the detection algorithm may identify other attackers so that the solution converges toward the total detection of intruders.

15.7. Conclusion

WSNs are a promising emerging technology from which powerful tools for remote monitoring will be possible to emerge. For such technology to be adopted, especially in the context of highly vulnerable applications, the question of security must be the priority. In this chapter, we presented the main types of attacks suffered by WSNs and four security approaches that are being prepared by the authors of this chapter. The first approach addresses a preventive security solution and a state of the art is given on possible solutions that provide data exchange confidentiality, integrity, freshness and non-repudiation and authentication of their origin. The other three proposals focus on which behavior to adopt after an intruder is detected on the network, with mechanisms of tolerance and/or detection of intruders.

This chapter describes four preventive security solutions: SNEP, μTESLA, TinySec and Zhu *et al.* These solutions are interested in the protection of communications in the WSN with solutions offering protection for the sensor sender to the BS and others a link-level sensor-to-sensor protection up to the BS. This chapter highlights the costs of such solutions in terms of energy consumption and storage overhead, and analyzes the residual security flaws and problems induced by such mechanisms over interesting WSN functionalities like the aggregation. Note that all of these solutions are based on the strong assumption that sensors share a key with the BS or a key group in between. These assumptions may affect the deployment of such solutions. Possible automation mechanisms for key distribution are actively studied and are described in Chapter 16.

The proposals to tolerate or to detect an intruder in a WSN reveal that different strategies may be adopted (for example, centralized versus decentralized, intruder tolerance versus intruder detection), each with their own advantages and disadvantages. However, all these strategies have in common the desire to minimize expenditure in energy, which is one of the most valuable resources in WSN. We also emphasize that the proposals presented here are far from exhaustive in intruder detection and tolerance. Many other research works are being carried out on this theme nowadays.

The presented work does not constitute final solutions. It is possible to make several improvements and extensions. For example, in the decentralized detection system work, all monitoring nodes are predefined. In addition to the economy of energy, the rotation of the monitor among nodes would protect the IDS itself. In effect, an intruder may take advantage of some privileged information to better perform an attack. For example, if it knows the location of monitors in the network, it may better hide itself. By contrast, if the roles are dynamically defined by independent rotation cycles, the intruder cannot identify monitors and may not succeed in its attack. This rotation could be done in several ways: randomly; by round-robin protocol; by election, when the nodes shall cooperate to choose the next monitor; and managed by the BS. The work on tolerance, on the other hand, is focused on a certain type of attack and routing algorithm. It would be interesting to verify if the solution also works for other types of attacks and how easy it is adapted to other routing algorithms.

Research in the field of WSN security has intensified in recent years, but the scientific problems are far from solved. Significant scientific progress is expected in the next few years.

15.8. Bibliography

[AMO 04] AMOR N., BENFERHAT S., ELOUEDI Z., "Naive Bayes vs decision trees in intrusion detection systems", *Proceedings of the ACM Symposium on Applied Computing*, p. 420-424, ACM Press, March 2004.

[AXE 99] AXELSSON S., "The base-rate fallacy and its implications for the difficulty of intrusion detection", in *Proceedings of the 6th ACM Conference on Computer and Communications Security*, Singapore, p. 1-7, ACM Press, November 1999.

[BAL 03] BALEPIN I., MALTSEV S., ROWE J., LEVITT K., "Using specification-based intrusion detection for automated response", in *Proceedings 6th International Symposium – Recent Advances in Intrusion Detection (RAID)*, Pittsburgh, PA, p. 136-154, 2003.

[BAR 96] BARBOSA V., *An Introduction to Distributed Algorithms*, Cambridge, Massachusetts, MIT Press, 1996.

[CRO 04] Crossbow Technology, Inc., *Mica 2 Wireless Measurement System*, Product Specification, 6020-0042-04 Rev B, San Jose, USA, February 2004.

[DEN 03] DENG J., HAN R., MISHRA S., "A performance evaluation of intrusion-tolerant routing in wireless sensor networks", in *Proceedings of the 2nd IEEE International Workshop on Information Processing in Sensor Networks (IPSN 2003)*, April 2003.

[GAN 01] GANESAN D., HEIDEMANN J., SILVA F., INTANAGONWIWAT C., GOVINDAN R., ESTRIN D., "Building efficient wireless sensor networks with low-level naming", in *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, Banff, Canada, p. 146-159, ACM Press, October 2001.

[GAY 03] GAY D., LEVIS P., BEHREN R., WELSH M., BREWER E., CULLER , D., "The nesC language: a holistic approach to networked embedded systems", in *Proceedings of the ACM SIGPLAN 2003 Conference on Programming Language Design and Implementation (PLDI)*, p. 1-11, ACM Press, June 2003.

[GHO 98] GHOSH A., WANKEN J., CHARRON F., "Detecting anomalous and unknown intrusions against programs", in *Proceedings of the 14th Annual Computer Security Applications Conference (ACSAC'98)*, Scottsdale, AZ, p. 259-267, December 1998.

[HIL 00] HILL J., SZEWCZYK R., WOO A., HOLLAR S., CULLER D., PISTER , K., "System architecture directions for network sensors", in *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2000)*, Cambridge, p. 93-104, ACM Press, November 2000.

[HU 06] HU Y., PERRIG A., JOHNSON D., "Wormhole attacks in wireless networks", *IEEE Journal on Selected Areas in Communications*, vol. 24(2), p. 370-380, IEEE Press, February 2006.

[ILG 95] ILGUN K., KEMMERER R., PORRAS P., "State transition: a rule-based intrusion detection approach", *IEEE Transactions on Software Engineering*, vol. 21(3), p. 181-199, IEEE Press, 1995.

[KAR 02] KARLOF C., LI, Y., POLASTRE , J., ARRIVE: *Algorithm for Robust Routing in Volatile Environments*, Technical Report UCB/CSD-03-1233, University of California, Berkeley, May 2002.

[KAR 04] KARLOF C., SASTRY N., WAGNER D., "TinySec: a link layer security architecture for wireless sensor networks", in *Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys 2004)*, Baltimore, Maryland, USA, p. 162-175, November 2004.

[KAR 03] KARLOF C., WAGNER D., "Secure routing in wireless sensor networks: attacks and countermeasures", in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, May 2003.

[KO 97] KO C., RUSCHITZKA M., LEVITT K., "Execution monitoring of security-critical programs in distributed systems: a specification-based approach", in *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, p. 175-187, May 1997.

[LAN 99] LANE T., BRODLEY C., "Temporal sequence learning and data reduction for anomaly detection", *ACM Transactions on Information and System Security (TISSEC)*, vol. 2(3), p. 295-331, 1999.

[LIN 99] LINDQVIST U., PORRAS P., "Detecting computer and network misuse with the production-based expert system toolset (P-BEST)", in *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, California, p. 146-161, May 1999.

[LIP 00] LIPPMANN R.P., FRIED D., GRAF I., HAINES J., KENDALL K., MCCLUNG D., WEBBER D., WEBSTER S., WYSCHOGRAD D., CUNNINGHAN R., ZISSMAN M., "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation", in *Proceedings of the on DARPA Information Survivability Conference and Exposition (DISCEX'00)*, Hilton Head, South Carolina, Los Alamitos, CA, IEEE Computer Society Press, p. 12-26, January 2000.

[MAR 05] MARTINS M., SILVA A., LOUREIRO A., RUIZ L., *Simulador para um Sistema de Detecção de Intrusos em Redes de Sensores Sem Fio*, Technical Report, UFMG, Minas Gerais, Brazil, March 2005.

[OLI 05] OLIVEIRA S., WONG H.C., NOGUEIRA , J.M., "NEKAP: Estabelecimento de Chaves Resiliente a Intrusos em RSSF", in *Proceedings of the Brazilian Symposium on Computer Networks – SBRC'05*, Fortaleza, Brazil, 2005.

[OLI 06] OLIVEIRA S., WONG H.C., NOGUEIRA , J.M., PAULA W., "Rotas Alternativas para a Detecção e Aumento da Resiliência à Intrusão Distribuída em RSSF", in *Proceedings of the Brazilian Symposium on Computer Networks – SBRC'06*, Curitiba, Brazil, p. 1247-62, 2006.

[RUI 03] RUIZ L., NOGUEIRA J., LOUREIRO A., "MANNA: a management architecture for wireless sensor networks", *IEEE Communications Magazine*, vol. 4 (2), p. 116-125, 2003.

[RUS 03] RUSSEL S., NORVIG P., *Artificial Intelligence – A Modern Approach*, 2nd edn, Prentice Hall, 2003.

[SHN 04] SHNAYDER V., HEMPSTEAD M., CHEN B., ALLEN G., WELSH M., "Simulating the power consumption of large-scale sensor network applications", in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys'04)*, Baltimore, MD, USA, p. 188-200, 2004.

[PAX 98] PAXSON V., "BRO: a system for detecting network intruders in real-time", in *Proceedings of the 7th USENIX Security Symposium*, San Antonio, TX, January 1998.

[PER 02] PERRIG A., SZEWCZYK R., WEN V., CULLER D., TYGAR D., "SPINS: security protocols for sensor networks", *Wireless Networks Journal*, vol. 8 (5), September 2002.

[PER 00] PERRIG A., CANETTI R., TYGAR J., SONG D., "Efficient authentication and signing of multicast streams over lossy channels", in *Proceedings of the IEEE Symposium on Security and Privacy*, California, p. 56- 73, May 2000.

[SIL 05] SILVA A.P.R., Detecção de Intrusos Descentralizada em Redes de Sensores Sem Fio, Master's dissertation, UFMG, Minas Gerais, Brazil, 2005.

[STA 98] STALLINGS W., *Cryptography and Network Security: Principles and Practice,* 2nd edn, Prentice Hall, 1998.

[STD 02] STADDON J., BALFANZ D., DURFEE G., "Efficient tracing of failed nodes in sensor networks", in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, Geórgia, USA, p. 122-130, 2002.

[TEI 06] TEIXEIRA F. A., Detecção de Intrusos por Observação em Redes de Sensores Sem Fio, Master's dissertation, UFMG, Minas Gerais, Brazil, 2006.

[TSE 03] TSENG C., BALASUBRAMANYAM P., KO C., LIMPRASITTIPORN R., ROWE J., LEVITT K., "A specification-based intrusion detection system for AODV", in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, Fairfax, Virginia, p. 125-134, ACM Press, 2003.

[WOO 02] WOOD A., STANKOVIC J., "Denial of service in sensor networks", *IEEE Computer*, vol. 35 (10), p. 54-62, IEEE Computer Society Press, October 2002.

[ZHU 04] ZHU S., SETIA S., JAJODIA S., NING P., "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks", in *Proceedings of the IEEE Symposium on Security and Privacy (S&P'04)*, Oakland, California, p. 259, May 2004.

This page intentionally left blank

Chapter 16

Key Management in Wireless Sensor Networks

16.1. Introduction

With the miniaturization of electronic systems, sensor networks will gradually invest our everyday life in several areas [HAB 06] like home automation, environment, medicine or food. Applications based on sensor networks include remote health monitoring of the elderly with body sensors (the heart, etc.), the detection of fires in parks, the detection of an individual's presence in a house, or cold chain monitoring for sensitive food products from the producer to the food distributor (temperature sensors positioned on some pallets, for example).

Several research avenues are also open to the concept of ambient intelligence, i.e. the possibility that elements formed by networks that surround us (including sensor networks) have a capacity for automated learning that fits our environment and anticipates our needs. This is also known as "smart home" or "intelligent habitat".

The military sector studies wireless sensor networks (WSNs) and the security issue with great interest, having in mind the killer application of monitoring a territory and an enemy's territory. However, it is very important that the information provided by the sensors is of great reliability and is precise enough to be usable by the deployed troops.

Chapter written by Chakib BEKARA and Maryline LAURENT-MAKNAVICIUS.

This chapter focuses on the management of the cryptographic keys that are useful to protect these applications. In particular, it identifies the security needs specific to WSN architectures, and it then describes different key management approaches suggested by the scientific community, specifying for each protocol under study the costs incurred in terms of storage, calculation and transmission.

16.2. Introduction to key management

The following actors are usually distinguished in WSN applications (see Figure 16.1):

– Sensors: a sensor makes certain measurements of temperature, luminosity, etc. It emits an alarm when an event occurs (e.g. exceeds a temperature threshold) or a message containing results of measurements on demand. Because sensors have a limited coverage and these sensor networks can be extended (e.g. 500×500 m^2), it is necessary that these messages are relayed from sensor to sensor until their destination. The sensors therefore integrate routing functions.

– One (or several) base station(s) (BS): a BS is a bridge connecting the WSN to a fixed IP, GSM, etc., network. It is thus responsible for the communications of the WSN with the outside. In particular, the BS can ask the WSN for new measurements; the BS also relays alarms resulting from the WSN to a platform of administration, or any other message issued by the WSN. Very often, the BS is a fixed node having energy resources that are much more important than the sensors. Therefore, the BS has the capacity to disseminate information on a sufficient coverage to reach a large number of sensors. Let us note that in any case, the sensors located between the BS and remote sensors must route messages to remote sensors, even if the latter receives the messages of the BS directly.

– A cluster head: in order to simplify the management of the WSN and in particular the management of the keys, several solutions recommend forming sub-groups of sensors called clusters. A cluster is made of a set of sensors of the same vicinity that communicate directly with each other (without requiring routing). The sensor elected or indicated as a “cluster head” is responsible for the exchanges between the sensors of its cluster and the rest of the network. The nodes of its cluster are called child nodes. Thus, it is usually the head of a cluster that is responsible for aggregating data from the cluster to the BS. The objective of aggregation is to perform a certain treatment on the statements of measurements provided by a set of sensors (calculation of an average, a minimum or a maximum, etc.) and to transmit only the result; this makes it possible to limit the size of the messages transmitted out of the cluster. On the other hand, this hierarchical organization in clusters has some disadvantages like the automation of the formation of clusters, the election of the cluster head, and the mobility of the sensors from one cluster to another.

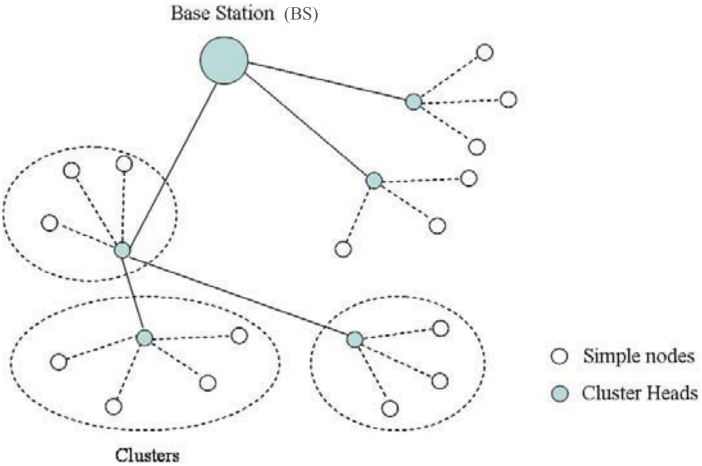


Figure 16.1. *The components of a WSN*

According to the sensitivity level of WSN applications, it is necessary to set up certain security services, traditionally: authentication of sensors and the BS, confidentiality and integrity of data, protection of the aggregation, etc. In Chapter 15, several security solutions (or preventive mechanisms) were presented. The majority of them rest on the use of symmetric (or secret key) cryptography because of low cost calculation, but also the limited size of the symmetric keys, the size of the resulting ciphertext message (equal to the size of the associated cleartext message) and the size of the MAC used for authentication (11 bytes in general). The characteristics of symmetric cryptography actually adapt well to sensors which are known for their low computing and storage capacity, as well as for their low energy autonomy, as the battery sensors are often non-rechargeable and non-replaceable. However, the use of symmetric cryptography raises certain key management problems, such as the generation and secure distribution of secret keys, the renewal of keys, the revocation of secret keys, the broadcast source authentication, etc.

To tackle the problem of symmetric cryptography to establish a secret between two entities, public key cryptography might be at first sight a better candidate. It allows two or more entities to agree on a shared secret with no need for any secret information shared as a preliminary or a protected transmission channel. Certain solutions [MAL 04] make use of it to secure the WSN, but today this cryptography consumes far too many resources [PER 01] given the capabilities of the sensors, so it is not considered as a realistic candidate for securing WSNs.

This chapter thus analyzes the methods for establishing and managing symmetric cryptographic keys. Although applicable to the wired sensor networks, it focuses on WSNs which are more security demanding and impose stronger constraints, like the absence of a network infrastructure. Indeed, wired sensor networks generally assume the existence of an infrastructure and thus a simpler management solution based on a central entity can be designed.

16.3. Security needs of WSNs

We can identify six key security needs, most of which are common to other types of networks:

– Authentication: a sensor needs to authenticate any node with which it communicates. Moreover, as the nodes must collaborate to route the packets of the other nodes, a sensor must first authenticate the node originating the packet before doing relaying in the network. Likely, our network is protected against external attacks (intruders); the aggregation of data is done on authenticated data, the energy of our network is preserved while avoiding unnecessary traffic routing.

– Data confidentiality: due to the broadcast mode used to communicate in WSN, data confidentiality is a vital need to counteract passive eavesdropping. Thus, it may be necessary to set up a secret communication within the network between a sensor and the BS, or between a pair of sensors.

– Data integrity: a WSN is mainly used to collect information of an area of interest; it if of high importance to ensure the integrity of these data, because the whole process of aggregation depends on it. A node of the network needs both to authenticate the origin of data and to make sure that they were not modified maliciously while in transit.

– Non-repudiation: the nodes of WSN must protect themselves against malicious nodes that could deny having performed some actions (e.g. sending a message, etc.).

– Resistance and tolerance to compromising [CAM 04]: following the compromising of some nodes, an attacker can extract all the secret information (secret keys) known to these nodes, but it should not be able to deduce some other secret information. In particular, an attacker should not be able to deduce the secret keys held by non compromised nodes and should not be able to inject into the network some clones of compromised nodes or even fake nodes. It is important that a malicious or compromised node be revoked as soon as possible.

– Availability [CAM 04]: a WSN is an ad hoc network without any infrastructure, and consequently any centralized solution should be avoided. Indeed, the security services described above should not rely on an online central entity like a KDC (Key Distribution Center). Otherwise, this centralized solution could easily be subject to DoS attacks. First the compromising of the KDC would destroy all the network security. Second, the unreliable nature of the communication in the WSN can sometimes make the KDC server inaccessible and thus can affect the security of the WSN itself.

All these security services are guaranteed in the WSN context mainly by using symmetric cryptography. However, these cryptographic keys need a type of management (creation, distribution, establishment, renewal, revocation, etc.) that guarantees the availability of the security services, the tolerance and resistance to compromising, and a low storage and transmission calculation cost, in order to take into account the constraints imposed by the sensors. It should be noted that the proposed security solutions for (mobile and ad hoc) wireless networks, especially those based on the use of public key cryptography, do not adapt to WSN because of the important costs induced during the establishment of shared secrets between sensors, or when sending MACs (of considerable size) in order to authenticate the origin of messages. We therefore focus in this chapter on solutions based on symmetric cryptography.

16.4. Key management problems in WSNs

Unlike other equipment (laptops, PDAs, mobile phones, etc.) with a certain calculation, storage and energy capacity, sensors are miniaturized equipment presenting the following limitations [AKY 02]:

– a low storage capacity (~ 4 to 8 KB of RAM);

– a low calculation capacity (CPU with 8-bits/8 MHz);

– a weak energy autonomy (2 batteries of 1.5 V);

– a low level of physical security, because it is relatively easy to extract all the information that a sensor contains, including cryptographic keys and other secret information [CHA 03, HAR 04].

Moreover, in a WSN, there are four types of communication:

– unidirectional: between a sensor and the BS, or between a couple of sensors;

– global broadcast: between the BS and all sensors;

– local broadcast: between a sensor and its neighbors, or a group of sensors forming a cluster;

– aggregation: in this type of communication, the responses of the sensors come following a request from the BS, or are triggered by events. During the routing of the data towards the BS, certain nodes called "aggregator nodes" aggregate the data received from child nodes to produce only one piece of data (e.g. the mean, the sum, the standard deviation of the data received).

For each of these types of communication, a cryptographic key is essential:

– A pair-wise key for securing communications between a pair of sensors. This secret key serves to guarantee the confidentiality of exchanged data, as well as the authentication of the origin of the data and their integrity.

– A group key shared between the BS and all sensors. This key is used to protect the network against passive eavesdropping.

– A cluster key shared between a group of nodes or between a node and its neighbors. This key is used to protect the communications in the cluster, so that only the nodes of the cluster can decrypt the data transmitted in the cluster.

– A pair key shared between an aggregator node and its child. It serves for an aggregator node to authenticate each child node and ensure the integrity of the received data before the aggregating operation.

In addition to these cryptographic keys, other keys prove to be necessary within the framework of a broadcast source authentication, in order to minimize the energy consumption due to the length of the transmitted data:

– Authentication key of a global broadcast source: this key enables all the nodes of the network to authenticate the origin of the messages issued by the BS, for example, when the BS sends a solicitation message through the WSN.

– Key authentication of a local broadcast source: this key is used to ensure the authenticity of a message broadcast to the neighboring nodes. It can be periodical routing information broadcast.

Because of the constraining technical features of sensors and different types of communication in a WSN, several problems arise when using symmetric cryptography:

– The WSN are generally deployed randomly without preliminary knowledge of the position of each sensor. Therefore, it is not possible to predict before deployment which neighbors will have a sensor. One solution for establishing some secure connections between pairs of sensors would be to pre-configure each sensor with

N-1 secret keys (in case the WSN comprises N sensors). This solution is not satisfactory: the memory size of the sensors does not make it possible to store a great number of keys; it would lead to wasting most of the memory of the sensors because the sensors communicate mainly with their direct neighbors; finally, this solution is not dynamic and does not allow new sensors to join the WSN unless a KDC plays the role of a mediator in the dynamic establishment of secret keys. Another solution would be to use a group key that would ensure both authenticity and confidentiality of messages from sensors, but this solution is not satisfactory either. On the one hand, authentication is of weak level because it only proves a message comes from the group and not from a precise sensor; on the other hand, compromising only one sensor (and thus the group key) renders all the security of the network futile.

– Authentication of a broadcast source like TESLA [PER 00] for ad hoc networks requires the commitment key (see section 16.8.1) to be considered authentic by the nodes of the WSN. If the broadcast source is a BS, it is reasonable to believe that the commitment key can be pre-configured in WSN sensors, but it also means that the sensors are still evolving in the presence of the same BS. On the other hand, if the authentication service is implemented to allow any sensor to authenticate their own broadcast messages, the only way to guarantee the authenticity of this commitment key would be to use public key cryptography, but, as mentioned in section 16.2, this type of cryptography is to be avoided in the WSN [PER 01].

– A revocation mechanism is difficult to implement in a WSN environment. However, one solution is recommended: a sensor which detects a malicious node must disseminate a bad reputation for this node in the network; once the number of valid reputations received from other sensors reaches a certain threshold, the suspected node is removed. This of course assumes that the reputations are verified by all the nodes. Therefore, each node is supposed to have its own information and to obtain information from other nodes, in order to check the reputations. However, this raises storage problems.

From the above, we can conclude that any key management protocol in a WSN must meet the following requirements:

– it must have the lowest possible cost in computing, storage, transmission, and energy consumption;

– it must allow any pair of sensors to establish a shared secret. Similarly, it must allow the establishment of a group key and cluster keys. An unauthorized sensor should not be able to establish shared secrets with other valid sensors in the network, or to be a valid member of a cluster;

– it should not assume knowledge of the positions of nodes prior to their deployment for the preparation of various cryptographic keys. The sensors are often deployed at random (from a helicopter for example). Moreover, deployment errors can occur, which severely disrupt the establishment of the cryptographic keys;

– it should not assign the task of key management to an online KDC which is prone to breakdowns and attacks, making it unavailable;

– it should be tolerant to compromising, by preventing a compromised node from revealing secret information on the security established between non-compromised nodes, and by preventing an attacker from populating the network with clones of compromised nodes or fake nodes having non-existing identifiers;

– it should detect any compromised node or malicious node as soon as possible and prevent an attacker from revoking a legitimate node of the network;

– the nodes must form a securely connected network after its deployment. A network is known as “connected” if there is a path to connect any pair of nodes of the network. A node is known as “securely connected” if there is a path between any pair of sensors that is formed exclusively of secure links, each one of these links being made secure by a shared secret key;

– all new sensors arriving in the WSN after its deployment must be able to establish secure links with its vicinity.

16.5. Metric for evaluating key management protocols in WSNs

For a simplified comparison between the key management protocols proposed in the literature, metrics are defined and used to estimate the costs induced by the implementation of key management protocols. For example, the following metrics are defined in [CAM 04, ESC 02, ZHU 03]:

– storage cost (memory): the memory capacity in bytes necessary for the processing to manage keys, and in particular cryptographic keys;

– cost in calculation: the CPU processing time consumed during the cryptographic operations (establishment of several keys, encryption, authentication, etc.);

– cost in transmission: the number of bytes transmitted during the key management process (establishment of various cryptographic keys, revocation, etc.). Since the consumption of the energy is primarily due to the transmissions of the data, as underlined in section 16.3.2, the cost in transmission is critical in the WSN. According to [PER 01], more than 90% of the energy of a sensor is spent in transmission and according to [WAN 05], the transmission of a bit requires the same energy as the execution of 2,090 instruction cycles.

– network connectivity: this metric measures the probability that the network is securely connected after its deployment;

– tolerance and resistance to compromising: this represents the non-compromised portion of the network, or in other words the number of secure links between non-compromised nodes that an attacker can listen. In addition, this metric measures the ability of an attacker to populate the network of clones of compromised nodes or fake nodes;

– scalability: the non-performance degradation when the size of the network increases. This metric is also important because a WSN may contain tens of thousands of sensors.

16.6. Classification of key management protocols in WSNs

In [CAM 04], Camtepe and Yener present a detailed classification of distribution key protocols in WSNs. The generation and distribution of cryptographic keys are at the heart of any key management protocols in WSNs. Most of the distribution key protocols proposed in the literature deal mainly with the distribution of pair-wise keys, and are based on the pair-wise keys created for the distribution of cluster keys and group keys in the network. Very few protocols [ZHU 03] explicitly treat the establishment of different types of keys that adapt to different types of communication in a WSN. In the same way, protocols [CHA 03] rarely deal in depth with the issue of revocation of malicious or compromised sensors; they simply assign this task to the BS which is considered as the trusted entity of the network. We must also stress the existence of a class of protocols [PER 01, CHE 05, LIU 03] dealing exclusively with the problem of broadcast source authentication in a WSN.

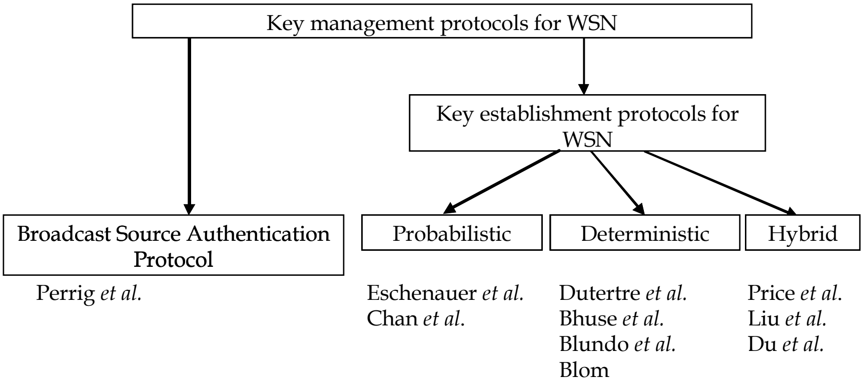


Figure 16.2. *Classification of key management protocols in WSNs*

The classification given in [CAM 04] is mainly based on:

– the type of keys to be established (pair-wise key, cluster key, group key);

– the approach used for key establishment (deterministic, probabilistic, hybrid);

– the mechanism for establishing the key (pre-distribution, dynamic generation). In the pre-distribution mechanism, two neighboring nodes establish a secret key if they share a preliminary key, while in a dynamic generation mechanism, two nodes interact to agree (generate) on a common key;

– the cryptographic material used (random keys, master key, key matrix, polynomials). The probabilistic approaches use random keys initially loaded in the sensors before their deployment. The deterministic approaches require pre-configuring the sensors before their deployment with certain secret elements of matrices, secret polynomials or public values common to all the sensors; these elements enable these sensors to create pair-wise keys with their neighbors, and to create cluster keys.

According to the classification of Camtepe and Yener [CAM 04], the simplified classification illustrated in Figure 16.2 may be obtained.

16.7. Notations and assumptions

The notations in use in the remainder of the chapter are as follows:

– BS: a trusted entity in the network that cannot be compromised and that has no resource constraints contrary to sensors;

– u, v: two sensors representing two nodes of the WSN;

– N_u: four bytes random generated by the node u;

– N: number of sensors in the network. We assume $N \leq 2^{16} - 1$, so each node can have a unique 2-byte identifier;

– m: average degree (the number of one-hop neighbors) of a node in the network;

– d: average number of secure links (secret keys) directly established by a node with its neighbors after its deployment;

– p_{local}: probability that a node directly establishes a secret key with a neighboring node (used for the probabilistic and hybrid approaches);

– $K_{SB,u}$: secret key initially loaded by the BS in node u before its deployment;

– $K_{u,v}$: secret key shared between the nodes u, v;

– $\{M\} < K >$: message M encrypted with the secret key K;

– $MAC\ (K, M)$: message authentication code calculated on the message M using the secret key K and serving to authenticate the origin of this message;

– F : irreversible pseudo-random function (PRF);

– $F_K(M)$: pseudo-random function having as parameters a message M and a secret key K.

In WSN networks, the MAC and cryptographic keys are usually 8 bytes long. Thereafter, this hypothesis is stated. Similarly, we assume that a PRF produces an output of 8 bytes.

16.8. Broadcast source authentication protocols

The BS often uses broadcast mode to communicate with all the sensors. Very often, the messages issued by BS are solicitation messages addressed to all sensors. It is therefore important to authenticate the origin of these messages. A first idea for authentication would be to use the pair-wise key shared individually by the BS with each sensor and thus the BS would have to calculate N MAC and append them to the broadcast message; in the context of WSN, it is not the cost in calculation by the BS which is problematic, but the cost of transmission since $8 \times N$ bytes are then issued. The alternative would be to use a key shared by the BS and all the sensors, but the risk would be high that a compromised sensor impersonates the BS.

16.8.1. *Perrig et al. µTESLA protocol*

The Perrig *et al.* protocol [PER 01], also called µTESLA (micro time efficient streaming loss tolerant authentication), was presented in section 15.3.4, so a short recall of its operation is presented here with particular emphasis on the management and initialization of the nodes.

16.8.1.1. *Short reminder of operation*

Upon initialization, the BS generates a chain of keys of N elements from a secret key $K_g{}^n$, such that $K_g{}^{i-1} = F(K_g{}^i)$, $i = 1 \ldots n$. The key $K_g{}^0$ is called a commitment key and must be known by all the sensors. In general, the key ${K_g}^0$ is initially loaded into the nodes before deployment.

To enable WSN sensors to authenticate the source of the messages issued from the BS, the BS appends to each of its messages a MAC, which is calculated over the transmitted message and a secret key. The specificity of µTESLA is to designate a key ${K_g}^i$ to be used by the BS only in the time interval T_i to T_{i+1} of T duration (see Figure 16.3).

To enable sensors to check the validity of the MAC, the BS broadcasts the used key ${K_g}^i$ δ time intervals after its use. This means that after δ time intervals, the sensors can verify the authenticity of the source of the message it just received. Either they need to verify that ${K_g}^{i-1} = F({K_g}^i)$, or, if they do not possess the key ${K_g}^j$, they need to check that ${K_g}^j = F^{i-j}({K_g}^i)$, the idea being to verify the ownership of the key ${K_g}^i$ to the key chain of BS. Then it is necessary to verify that the MAC is correct.

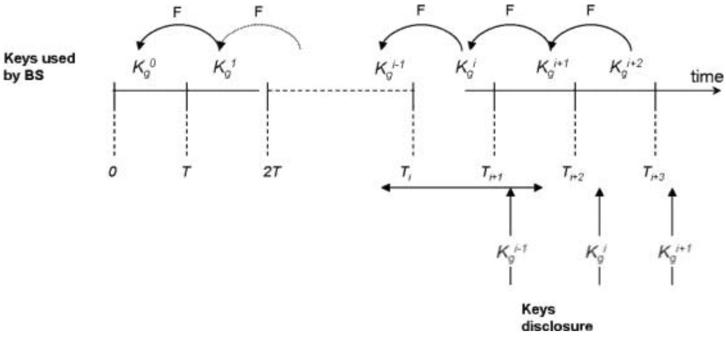


Figure 16.3. *µTESLA protocol with* $\delta = 2$

16.8.1.2. *Initialization of sensors*

The initialization of the sensors can take two forms:

– Form 1: the sensors are pre-configured with µTESLA parameters ${K_g}^0, T_1, T, \delta$ before their deployment.

– Form 2: the sensors are pre-configured with a pair-wise key shared with the BS: $K_{SB,u}$. Thus, the BS can communicate to a sensor u (unicast communication) all the µTESLA parameters in a secure manner. The following exchanges are set up, first from u to the BS, then from u to the BS during time interval $T_{i+\delta}$:

$$u \rightarrow SB : N_u$$

$$SB \rightarrow u : T_{SB} \mid K_g{}^i \mid T_i \mid T \mid \delta, MAC(K_{SB,u}, N_u \mid T_{SB} \mid K_g{}^i \mid T_i \mid T \mid \delta),$$

where T_{SB} is the copy of the BS local clock when broadcasting the message.

Thanks to T_{SB}, the synchronization between the BS and the sensor u is guaranteed. The µTESLA parameters that are received are trusted because they are integrity protected with a MAC.

16.8.1.3. *Advantages and disadvantages*

The µTESLA protocol [PER 01] is advantageous for several reasons:

– it significantly reduces the cost in calculation of the BS, since only one origin authenticated message is sent by the BS instead of N separate authenticated messages. Therefore, the transmission cost is also reduced, thus preserving the total energy of the network;

– it avoids using the public key cryptography for authenticating the initial value K_g^0 of the key chain, thus it takes into account the limitations of sensors in terms of storage, and their inability to implement public key cryptography.

µTESLA suffers from the following disadvantages:

– It relies on tight clock synchronization between the BS and the sensors, in order to guard against the impersonation of the BS. To remedy this, it is possible to embed into the sensors and the BS an internal clock synchronization system, which allows a ε accuracy synchronization, such that $|T_{SB} - T_{sensor}| < \varepsilon$, where ε represents the error of synchronization which is an infinitely small value, and T_{SB} and T_{sensor} are the current time at the BS and sensors respectively. This synchronization could be implemented using a periodic signal broadcast by the BS to all nodes of the network.

– The choice of the time interval value T is crucial. Indeed, the shorter the time interval T, the more calculations are needed to check that key $K_g{}^j$ belongs to the key chain of the BS; this is particularly true if the data are released in distant time intervals (for example T_{10} and T_{150}). Similarly, the longer the time interval T, the more messages are stored for later authentication; there is then a high probability, especially in the case where the BS transmits a large number of messages, to make a sensor's buffers overflow; messages are likely to be lost and then retransmitted, thus leading to an overconsumption of the sensors energy.

– The first initialization method implies a considerable calculation cost for sensors joining the network after their deployment. Indeed, the later it joins the WSN, the longer the verification of key $K_g{}^i$ broadcast by the BS from the commitment key $K_g{}^0$. In addition, a new sensor is not guaranteed to be synchronized with the BS after it joins the network unless it employs onboard clock synchronization techniques.

– The second solution guarantees a better synchronization, and less calculation time for the nodes, but it requires the BS and sensor u to share a pair-wise key $K_{SB,u}$ and it introduces more calculation time by BS (compared to the first solution), and a significant cost in transmission for the network nodes that need to deliver the initialization message to each new incoming node.

– The µTESLA protocol is not easily extensible to another broadcast source like sensors, or in these cases, its extension to sensors would be extremely expensive in storage, as stated in the introduction of section 16.8.

16.8.1.4. *Induced costs*

According to the metrics described in section 16.5, the solution [PER 01] has the following characteristics:

– storage cost: the BS requires $8 \times (n + 1)$ bytes to store its chain of $n+1$ keys, while each sensor of the network needs 8 bytes to store the last disclosed key $K_g{}^i$;

– calculation cost: the BS conducts n PRF operations to generate its key chain, and a sensor needs a maximum of n PRF operations maximum to check all the disclosed keys;

– transmission cost: assuming that the sensors are initialized before the deployment of the network, the BS sends a maximum of $8 \times n$ bytes in case it uses its entire key chain. If the sensors are initialized when joining the network, the BS will send much more data (up to 16 bytes per sensor) and network sensors will exhaust their energy to deliver these data.

16.9. Probabilistic key management protocols

These protocols mainly handle the establishment of pair-wise keys between two neighboring nodes in a WSN. The main idea is to load each node in the network before its deployment with a random set of secret keys. Once the network is deployed, two neighboring nodes directly establish a shared secret if they have at least one common key in their set of keys. This class is called probabilistic since two neighboring nodes have a probability $p < 1$ of directly establishing a secret key.

16.9.1. *Eschenauer et al. protocol [ESC 02]*

16.9.1.1. *Initialization and establishment of pair-wise keys*

This protocol includes two phases:

– Phase 1 – initialization: the BS creates a wide random set P of secret keys, and numbers the keys from 1 to $|P|$. This latter 2-byte number serves as a unique identifier. Before its deployment, the BS configures each node u with a set of k distinct keys of P as well as IDs of these keys. Each key is 8 bytes long.

– Phase 2 – discovery and establishment of keys: after the deployment of the network, each node discovers its neighbors and establishes secure links with them. To do this, it locally broadcasts its ID (over 2 bytes) and the IDs of keys that it owns. Two nodes directly establish a secure link if they know at least one common key.

Given the probabilistic nature of the protocol, some nodes will not be able to directly establish secure links with their neighbors, because they do not know any common key. Two neighboring nodes u and v with no key in common must find a path in between where each node of the path has a secure link with the next one (and therefore it shares a common key). Once the path is found, u generates a secret key $K_{u,v}$ and sends it to v through this path. This process can also be used for renewal of a key between a pair of nodes in the network.

At the end of the two previous phases, the resulting network is almost completely securely connected at $\approx 99.999\%$, i.e., any pair of nodes can thus be interconnected via a path composed of secure links. This percentage is defined as one input to the protocol.

16.9.1.2. *Establishment of cluster keys and revocation of nodes*

The Eschenauer *et al.* protocol does not specifically address the establishment of cluster keys. However, a node can use the secret key put in place with its neighbors;

it can randomly generate a cluster key and securely distribute it to each of its neighboring nodes.

The revocation of a WSN sensor is operated by the BS because it is the only entity trusted by sensors. Thus, to revoke a node u, the BS must send a message containing the identity of this node, as well as the identifiers of the known keys of this node, in order to prevent the nodes of WSN from continuing to use compromised keys. This message is broadcast encrypted with a secret group key GK that the BS has generated randomly for this specific occasion. The BS subsequently broadcasts GK to the nodes by sending individually to each node GK encrypted with the shared pair-wise key $K_{SB,u}$. Each node of the network, whether or not it is a neighbor of u, must remove all the keys it has in common with it. The links secured with these keys are disabled and it is necessary to establish new secure links in accordance with phase 2 of section 16.9.1.1.

16.9.1.3. *WSN connectivity*

One of the basic criteria that should satisfy any key management protocol is the network connectivity, once the nodes are deployed and the pair-wise keys are established between nodes. This connectivity is summarized to guarantee a secure path (made exclusively of secure links) between any pair of nodes in the network.

In the case of the Eschenauer *et al.* protocol, the choice of parameters $|P|$ and k is crucial for obtaining a securely connected network. To do this, Eschenauer based himself on the work of Erdös and Rényi [ERD 60] which relates to random graphs.

The Erdős and Rényi theorem [ERD 60, HWA 04] is as follows: Let $G(N, p)$ be a random graph, where N is the number of vertices and p is the probability that there is a secure link between any two pairs of vertices of the graph.

The probability P_c that the resulting graph is connected (secure connected network) is expressed as: $\mathrm{P_c = lim_{n\text{->}\infty} Pr[G(N,p)\ is\ connected] = e^{e^{-c}}}$ is connected]= $\mathrm{e^{e^{-c}}}$ and the theorem concludes:

$$\mathrm{p = \frac{ln(N)}{\mathscr{N}} + \frac{c}{N}} \qquad [16.1]$$

where c is a constant.

As a result, the Erdös and Rényi theorem gives from the probability $\mathrm{P_c}$ of the desired network connectivity, the probability p that there is a secure link between any two vertices of the graph.

Remember that P_c is one input to the Eschenauer protocol, from which we can find p which represents the probability of global connectivity of a node. From p, we can deduce the average number d of secure links that a node must establish to obtain a connected network $d = p \times (N - 1)$. However, in the WSN, these links are established between a sensor and its (average) m direct neighbors, we find the probability of local connectivity of a sensor defined by

$$\mathrm{p}_{\mathrm{local}} = \frac{\mathrm{d}}{m} \qquad [16.2]$$

The probability that two nodes share at least one key is given by [ESC 02]:

$$\mathrm{p}_{\mathrm{share}} = 1 - \frac{((|\mathrm{P}|\text{-}\mathrm{k})!)^2}{(|\mathcal{P}| - 2k)! \times |\mathcal{P}|!},$$

from which we can deduce $|P|$ by considering $\mathrm{p}_{\mathrm{share}} \geq p_{local}$, and assuming that the number k of keys loaded in each sensor is one input parameter which depends on the storage capacity of sensors.

16.9.1.4. *Disadvantages*

The Eschenauer *et al.* protocol suffers from the following drawbacks:

– A compromised node may disclose keys held by non-compromised nodes. Thus, it does not satisfy the need for tolerance and resistance to compromise.

– In phase 1, an attacker can choose its victims carefully by analyzing the key identifiers held by each node. In phase 1, each node u broadcasts the list of key identifiers it owns, and each neighboring node v that shares at least one key with u can establish a secure link with it. The lists of key identifiers are transmitted unencrypted, so an attacker can, during this phase, easily retrieve them. Then an attacker has only to identify the smallest set of nodes whose union of the key lists is the largest component of the generated key space, then it has to compromise these nodes, thus compromising most of the key space, with the minimum of effort.

– An attacker can populate any part of the network with clones or fake nodes by configuring them with the keys extracted from compromised nodes.

– Once deployed, the WSN is no longer guaranteed to be securely connected.

– The revocation process has a high transmission cost because the BS has to circulate two messages in the WSN, one of them containing the list of compromised keys. On the other hand, the way a node is detected as compromised is not described in [ESC 02].

– The storage capacity of sensors is wasted since on average only d keys among the k keys being loaded into the sensor are useful in building secure links.

16.9.1.5. *Costs incurred*

Let analyze the costs of the approach [ESC 02]:

– Storage cost: assuming that $|P| \leq 2^{16} - 1$; each key identifier is 2 bytes long, and each node needs $(8+2)\times \mathrm{k} + 8\times(1-\mathrm{p}_{\mathrm{local}})\times \mathrm{m}$ bytes to store its keys, such that among the m keys established with its m neighbors, an average of $(1-p_{local})\times m$ keys are established through a secure path. Optionally, an extra $8\times m$ bytes are needed to store the cluster key of the neighboring nodes.

– Calculation cost: if a node has at least one common key with a neighbor, then no calculation cost is necessary to establish a shared secret. If a node has no common key with a neighbor, then the calculation cost is equivalent to the encryption of the generated secret key sent through the secure path in between. In the case of distributing a cluster key to its m neighbors, the node will perform m encryption operations.

– Transmission cost: each node sends $2+2\times k$ bytes to discover the shared keys with its neighbors. In addition, 8 extra bytes are needed for each key it establishes via the secure path. In the latter case, all the nodes contribute to the delivery of these 8 bytes, which is very energy consuming. Optionally, to distribute its cluster key, a node sends $8\times m$ bytes.

16.9.2. *Other approaches*

Unlike [ESC 02], Chan *et al.* [CHA 03] propose to associate each node before its deployment with randomly selected k nodes, by generating a unique secret key to be shared between them. From the identifier of its neighbor, a node knows when it can directly establish a key with it, because this key is unique. This approach has the advantage of providing greater tolerance and resistance to compromisation because the secret key established between non-compromised nodes cannot be disclosed. On the other hand, for the same level of connectivity and network size N, [CHA 03] requires more memory capacity of sensors; indeed, contrary to [ESC 02] where the same key can serve to secure multiple links, in [CHA 03], one key makes it possible to secure only one link.

16.10. Deterministic key management protocols

This class of protocols guarantees a total connectivity of the network because the sensors have the cryptographic material necessary to establish a secure link between any pair of nodes, contrary to the probabilistic approach where a secure link is only possible with a certain probability. The challenge for such deterministic protocols is to reduce the storage cost and the risks related to compromising.

16.10.1. *Dutertre et al. protocol [OTC 04]*

This protocol [DUT 04] rests on the strong assumption that sensors can be trusted right after their deployment because they can be compromised during the phase of pair-wise key establishment. Compromising can only take place once the configuration is completed. Moreover, the sensors are deployed in successive generations, each generation of deployed sensors is identified by $i \in [1,t]$.

16.10.1.1. *Initialization and establishment of pair–wise keys*

The protocol [OTC 04] works in two phases:

– Phase 1 – initialization: before deployment, each node is loaded with two main secret keys bk_1, bk_2, such that bk_1 is used to authenticate to other nodes of the same generation, and bk_2 is used to generate keys to be shared with the nodes of the same generation. In addition, each node u belonging to generation i receives a secret key GK_i known by nodes of generation i only, and a unique value R_u and a set of unique secret keys $Su_j = F_{GK_j}(R_u), j \in [i+1,t]$, such that each key Su_j is used to authenticate the nodes of the generation j.

– Phase 2 – establishment of pair-wise keys: after the deployment of nodes of the generation i, each node $u \in i$ broadcasts the message: $Hello, u, N_u, i, MAC(bk_1, Hello \mid u \mid N_u)$:

- a neighboring node v of the same generation i, after authenticating the message, replies with: $v \rightarrow u : Ack, u, v, N_v, MAC(bk_1, Ack \mid u \mid v \mid N_v \mid N_u)$;

- u and v then calculate their pair-wise key: $K_{u,v} = F_{bk_2}(N_u \mid N_v)$;

- a node w of an earlier generation $j < i$ answers node u: $w \rightarrow u : Ack1, u, w, R_w, MAC(Sw_i, Ack1 \mid u \mid w \mid R_w \mid N_u)$;

- u calculates $Sw_i = F_{GK_i}(R_w)$ and verifies the authenticity of the message, then it answers: $u \rightarrow w : Ack2, w, u, MAC(Sw_i, Ack2 \mid w \mid u)$;

- u and v then calculate their pair-wise key: $K_{u,w} = F_{Sw_i}(N_u \mid R_w)$.

Once the node u completes the establishment of the keys with its neighbors, it definitively removes from its memory the keys bk_1, bk_2 and GK_i. In order to do this, just after its deployment, the node u activates/initializes a timer to a value $T' < T_{min}$, where T_{min} is the minimum time necessary for an attacker to compromise a node. Once the timer expires, the node u removes bk_1, bk_2 and GK_i.

16.10.1.2. *Establishment of cluster keys, revocation and renewal of group keys*

Even if the protocol [DUT 04] does not deal explicitly with the establishment of cluster keys, a node u can generate a key randomly and communicate it securely to all its neighbors thanks to the previously agreed pair-wise key.

Before the deployment of the nodes, all the nodes from any generation are configured with a group key GK used by the BS to encrypt the broadcast messages. After a set of nodes S are compromised, it is important to change the group key. The BS then emits to each neighboring node u a message for revoking nodes encrypted with the pair-wise key $K_{SB,u}$. This message is relayed by the neighboring nodes to each of their own neighbors encrypted again with an adapted pair-wise key. The message sent by BS to a nearby node v is as follows:

$$SB \rightarrow v : S, SEQ, \{GK'\} < K_{SB,v} >, MAC(K_{SB,v}, S \mid SEQ \mid \{GK'\} < K_{SB,v} >),$$

where SEQ is a 2-byte sequence number and GK' is the new group key replacing GK.

16.10.1.3. *Advantages and disadvantages*

The protocol [DUT 04] has the following advantages:

– guaranteed network connectivity: each pair of nodes is able to establish a pair-wise key and thus a secure link;

– uniqueness of pair-wise keys: the pair-wise keys established between the nodes are unique, unlike the Eschenauer *et al.* protocol. This property makes the protocol more resistant to compromising;

– good tolerance and resistance to compromises: on the one hand, if a node u of a generation i is compromised after the phase of pair-wise key establishment, an attacker will not be able to establish keys with nodes of the same generation, or with nodes of former generations, because the useful secret parameters have been destroyed. The attacker will only be able to establish keys with the not yet deployed future generations. In addition, after the compromising of a node, because pair-wise keys are unique, it is not possible to make deductions about the keys used in the non-compromised part of the WSN.

The disadvantages of [DUT 04] are as follows:

– great vulnerability of the network during the phase of establishment of keys. The assumption is strong that a node cannot be compromised during the key establishment phase. If this assumption is not checked, a compromising then leads to the disclosure of the secret keys bk_1, bk_2 and GK_i, and it is then all of the security of the network which is compromised;

– very significant cost in calculation and transmission for the process of revocation;

– important storage cost for first generation nodes which must memorize the Su_j, especially if the number of generations t is large.

16.10.1.4. *Costs incurred*

The analysis of the costs for the protocol [DUT 04] is as follows:

– Storage cost: each node u of the generation i initially stores bk_1, bk_2, GK_i and GK, a single value R_u (of 4 bytes) and the secret keys $Su_j, j \in [i+1, t]$. Furthermore, each node establishes m secret keys with its neighbors. In total $8 \times (m + t - i + 4) + 4$ bytes are needed. In addition, $8 \times m$ bytes are necessary to store the cluster keys of the neighbors.

– Calculation cost: to establish a pair-wise key with a neighboring node, a node must compute at least one MAC generation and one PRF operation (in the event that nodes are of the same generation). To distribute its cluster key, a node must compute m encryption operations. To renew the group key GK of the BS after a node is compromised, each node must perform at maximum $m - 1$ encryption operations and $m - 1$ signature operations.

– Transmission cost: in order to establish a pair-wise key, each node must send 20 bytes on average. For the distribution of its cluster key, each node sends $8 \times m$ bytes on average. For the renewal of the group key after the compromising of a node, each valid node sends on average $20 \times (m - 1)$ bytes.

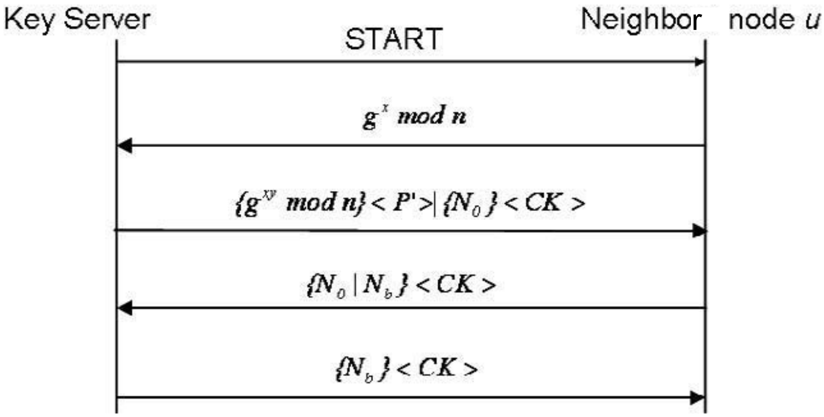


Figure 16.4. *Distribution of a cluster key based on the HDH-EKE protocol [DH 05]*

16.10.2. *Bhuse et al. protocol [BHU 03]*

Bhuse *et al.* consider the strong assumption that compromised nodes cannot reveal secret information. More precisely, they consider either that nodes cannot be compromised or that in the event of compromising, a node destroys itself. Contrary to the previous protocols, this protocol focuses mainly on establishing cluster keys and it relies, partly, on the Diffie-Hellman protocol.

16.10.2.1. *Initialization and establishment of cluster key*

The protocol works in two phases:

– Phase 1 – initialization: before the deployment of the nodes, the BS loads all the nodes with identical parameters: a 7-byte secret password P, as well as public DH parameters n and g [DIF 76];

– Phase 2 – establishment of cluster keys: after deployment, the nodes organize themselves into clusters and each cluster elects a node to be the key server (KS) of the cluster. Each KS creates a cluster key CK of 7 bytes and distributes it to the nodes of its cluster. The CK key is used to encrypt/decrypt the data inside the cluster and is only known by nodes of the same cluster. It is periodically renewed by then KS, and then indirectly distributed to the nodes of the cluster. The nodes belonging to several clusters obtain the keys of all the clusters to which they belong. The inter-cluster communications are done through these nodes. It should be noted that a cluster contains nodes which do not have a sufficient coverage to join the KS directly, and must rely on other nodes of the cluster to route their traffic. The KS initializes only its direct neighbors with the cluster key; the latter initialize in their turn their direct neighbors and so on until all the cluster is initialized with the cluster key.

Figure 16.4 shows how a node u is initialized by the key server through the HDH-EKE protocol:

– The KS broadcasts in the cluster a random value of a counter C, and each node of the cluster calculates a one-time password $P' = F(C,P)$. In this way, the password P is never directly used.

– The KS broadcasts locally a START message indicating to its neighbors the distribution of a cluster key. The KS calculates the cluster key $CK = g^{y} mod\ n$, where y is a random secret, also known as the private DH parameter of the KS.

– A neighboring node u answers the START message by sending its public key $g^{x} mod\ n$, where x is a random secret value corresponding to the private DH parameter of node u.

– The KS calculates the secret key $K_{SC,u} = g^{xy} mod\ n$, and then sends $\{g^{xy} mod\ n\} < P'>| \{N_0\} < CK >$ to u, where N_0 is a unique value generated by the KS.

– The node u finds $K_{SC,u}$ using P', and then deduces CK by computing $CK = (g^{xy} mod\ n)^{y^{-1}} \bmod n$, and recovers N_0. The last two exchanges allow mutual authentication and complete the process of establishment.

This process is repeated between the KS and each of its neighboring nodes, and between each node initialized and each of its neighbors not yet initialized with the cluster key. It should be noted that during this process, each node may possibly establish a pair-wise key with each of its neighbors, following the example of the key $K_{SC,u} = g^{xy} mod\ n$.

The KS periodically renews the CK by broadcasting in the group a random counter value C', greater than the previously broadcast value. The new cluster value CK' is obtained as follows: $P' = F(C',P)$ and $CK' = F(CK, P')$.

16.10.2.2. *Revocation and renewal of the cluster key*

This protocol does not define the revocation process, either for the created cluster key or for the pair-wise keys which can possibly be created. With the assumption considered that a sensor cannot be compromised, an attacker has limited means of action. If it manages to compromise the current CK, it will only be able to take advantage of it for a short period of time since the cluster key is periodically renewed by the BS. The attacker will not be able to calculate the new cluster key since it does not know the password P.

16.10.2.3. *Induced costs*

The security analysis of the solution [BHU 03] leads to the following results:

– Storage cost: each node requires an average of 72 bytes to store DH parameters and secret P. In addition, each node stores at least one cluster key of 7 bytes. Optionally, for each created pair-wise key, each node needs 8 additional bytes.

– Calculation cost: to retrieve the key of its cluster, each node must carry out the HDH-EKE protocol. This protocol is CPU expensive because it is based on the same mathematical principles as the public key cryptography. That is, each node must make an average of two modular exponentiations, one modular inversion and two decryption operations.

– Transmission cost: during the course of the HDH-EKE protocol for the establishment of a cluster key, the total exchange is up to 144 bytes.

16.10.2.4. *Advantages and disadvantages*

The approach has the following advantages:

– Use of cluster keys: the communications within the WSN are mainly secure thanks to cluster keys. This brings the advantage of reducing the number of keys to be stored.

– Malicious action limited in time after a cluster key has been disclosed: the cluster key will be exploitable only during the current time interval. Once the key is renewed by the KS, the attacker no longer has access to the data transmitted in the network.

– Malicious action limited in space after a cluster key has been disclosed: an attacker holding a valid cluster key will only be able to populate the cluster with clones nodes or fake nodes; it will not be able to populate another cluster.

– Uniqueness of the pair-wise key: if the establishment of the cluster keys also serves to establish pair-wise keys, then the pair-wise keys are guaranteed to be unique. Thus, the compromising of a pair-wise key will not have any consequences for the other communications within the WSN.

The solution [BHU 03] suffers from the following drawbacks:

– Criticality of the password P: if the secret P is disclosed, all the security of the WSN is compromised.

– Calculation and transmission and high time latency costs: these are due to the usage of HDH-EKE protocol. The initialization of the entire cluster can take a long time, because the protocol is done node by node.

– Vulnerability to DoS attacks: this vulnerability is caused by the HDH-EKE protocol. Indeed, the first two exchanges are not authenticated, thus allowing fake nodes to issue solicitations to a node already initialized. Moreover, as it is the requested node which carries out the first heavy calculations (2 encryption operations), it will be relatively easy especially in a WSN to exhaust the energy of a sensor thanks to multiple requests.

16.10.3. *Other protocols*

The deterministic approach of Blundo *et al.* [BLU 98] proposes using a symmetric polynomial with a bi-variable secret of the form: $f(x,y) = \sum_{i=0\cdots\lambda,\, j=0\cdots\lambda} a_{ij} x^i y^j \bmod(p)$ where λ is a security parameter strictly lower than the size of the network.

Each node u is initialized with a monovariable secret polynomial: $f(ID_u, y)$. In order to establish a pair-wise key, u and v need to exchange their identifiers u *ID* and v *ID* and each of them calculates: $K_{u,v} = f(ID_u, ID_v) = f(ID_v, ID_u)$. This approach to establish a pair-wise key is greedy in terms of computing time. If fewer than λ nodes are compromised in the network, the communications between non-compromised nodes are guaranteed as certain. Nevertheless, if $\lambda + 1$ nodes are compromised, an attacker can gain control on the entire network.

16.11. Hybrid key management protocols

This class of protocol is a hybridization between the probabilistic class and the deterministic class. It aims to reduce the too important storage cost required by the probabilistic class, and to improve the security level against compromisation of the deterministic class.

16.11.1. *Price et al. protocol [PRI 05]*

This protocol is based on a variant of the Blom protocol [BLO 84] that belongs to the deterministic class, and the Eschenauer *et al.* protocol [ESC 02] that belongs to the probabilistic class. The idea of Price *et al.* is to use the Blom protocol to build the second half of a pair-wise key to be shared between two nodes. The first half of this key is pre-established by using the Eschenauer *et al.* protocol.

16.11.1.1. *Initialization, discovery and development of the pair-wise key*

The Price *et al.* protocol is based on two phases:

– Phase 1 – initialization: the BS initializes each node with a set of k separate random keys (and their identifiers) randomly selected from a pool P, as described in the Eschenauer *et al.* protocol. The only difference is that the keys are only 4 bytes long instead of 8. The BS chooses a key among k to serve as a primary key unique to its holder, and the identifier of this key as a unique identifier of the node in the network. Each node has a unique identifier, a single primary key and several secondary keys. Then, the BS system generates the Blom system as follows:

$$G=\begin{bmatrix} 1 & 1 & \cdots & 1 \\ S & S^2 & \cdots & S^N \\ S^2 & (S^2)^2 & \cdots & (S^N)^2 \\ & & \vdots & \\ S^\lambda & (S^2)^\lambda & \ldots & (S^N)^\lambda \end{bmatrix}$$

The Blom system is a public matrix $G(N,\lambda+1)$, where λ is a security parameter, S is a public value and each element of the matrix has a length of 4 bytes. It should be noted that G is defined modulo p, a large prime number (for example, 512 bits long). The BS generates a symmetric secret matrix $D(\lambda+1,\lambda+1)$, then it calculates the symmetric secret matrix $A(N,\lambda+1)=(D\times G)^T$. The BS loads each node u with the line $A(i)$ where i is the identifier of the node u. Each node is also loaded with the public parameters p,S,λ.

In the original version of the Blom protocol, a pair of nodes of the network can create a pair-wise key, each one using its secret line and the public column of G corresponding to the other node. The set of created keys is given by the secret key matrix $K=A\times D=(D\times G)^T\times D$. The Blom system is guaranteed secure if the number of compromised nodes in the network does not exceed λ. If more than λ nodes are compromised, an attacker can calculate any pair-wise key being established in the network.

– Phase 2 – discovery and establishment of 8-byte pair-wise keys: after the deployment of nodes, each node u discovers its neighbors with which it shares 4 bytes secret keys, and then it tries to supplement these keys with other missing 4 bytes. In order to do this, u locally broadcasts the following message:

$ID_u \| N_u$, where ID_u is the identifier of u.

Let v be a neighboring node receiving the message. v checks that it shares a 4-byte key $K1$ with u by using ID_u. If so, v calculates the secret key $S_{v,u} = A(ID_v) \times G(ID_u)$ where $G(ID_u)$ is the column of G corresponding to u, such that: $G(ID_u) = 1, S^{ID_u}, ..., (S^{ID_u})^\lambda$. v sends to u the following message $ID_v, N_u, \{ID_{link}, N_v, SK_{u,v}\} < k_{v,u} >$, where ID_{link} is a unique identifier of the link established between (u, v), and $k_{u,v} = K1 \| S_{v,u}$. $SK_{u,v}$ is a 4-byte secret session key that is randomly generated, which represents the second part of the 8-byte secret key: $K_{u,v} = K1 \| SK_{u,v}$.

By receiving the message from v, the node u calculates $S_{u,v} = A(ID_u) \times G(ID_v) = S_{v,u}$, then gets $SK_{u,v}$ and finally obtains $K_{u,v} = K1 \| SK_{u,v}$. u sends to v the following message to conclude the establishment process: $N_v \mid \{ID_{link}, N_u\} < K_{u,v} >$. If v finds the same value N_v as it previously sent in the preceding message, then it deduces from it that the entity which answered knows the correct key $K_{u,v}$, and therefore the correct secret $K1$ and $A(ID_u)$. The node is deduced to be node u.

If u and v have no common key, then u finds a secure path to node v, then generates a secret key $K_{u,v}$ and sends it via this path, as for the Eschenauer *et al.* protocol.

16.11.1.2. *Revocation*

The protocol supports two revocation modes: the revocation of links and the revocation of nodes. In the first mode, only the 8-byte secret keys $K_{u,v}$ established with compromised nodes are revoked, while in the second mode all the 4-byte secret keys $K1$ held by the compromised nodes are revoked, hence there is a probability that secret keys between non-compromised nodes are revoked. The BS uses µTESLA to authenticate the broadcast revocation message.

16.11.1.3. *Advantages and disadvantages*

The Price *et al.* protocol offers the following advantages:

– Optimized memory: an 8-byte secret key is only generated on demand, unlike the Eschenauer protocol.

– Good resistance to compromisation: even if $\lambda + 1$ nodes are compromised, an attacker cannot gain the full control of the network. This is not the same for the Blom protocol. Therefore, the λ value is not forced to be as large as that of the Blom protocol.

– Uniqueness of pair-wise keys: the uniqueness of the generated pair-wise keys ensures good resistance and tolerance to compromisation because a node being compromised does not reveal any secret key between non-compromised nodes.

Several disadvantages can be noted:

– High calculation cost: this is mainly due to the modular multiplications and exponentiations made in the Blom system.

– No dynamicity of WSN networks: the Blom system is generated only once and cannot be modified after the unexpected introduction of new nodes in the network.

16.11.1.4. *Induced costs*

The establishment of pair-wise keys by the Price *et al.* protocol has the following characteristics:

– Storage cost: each node u requires $6 \times k + 4 \times (\lambda + 1) + 68$ bytes to store its k keys, its secret line $A[ID_u]$, as well as the public parameters p, S, λ. For each established secret key, a node requires 4 additional bytes if the key is directly established, and 8 bytes if it is established via a secure path.

– Calculation cost: in order to establish a secret key, a node u must carry out on average one modular exponentiation, $\lambda + 1$ modular multiplications, one modular vectorial multiplication, and one encryption and decryption operation.

– Transmission cost: on average, each node transmits 18 bytes during the establishment of a key with a neighboring node.

16.11.2. *Other protocols*

Du *et al.* [DU 03] propose generating w distinct secret matrices $A_1, ..., A_w$ as described in [BLO 84], and to load each node u with a set of $2 \leq t < w$ distinct secret lines $A_x[ID_u]$, where x is randomly selected between 1 and w. Two nodes u and v establish a secret key if they have at least one line from the same space (same matrix) A_y.

Liu and Al [LIU 03] propose generating w distinct bi-variable secret polynomials $f_1(x,y),...,f_w(x,y)$ as for [BLU 98], and to load each node u with $2 \leq t < w$ distinct monovariable polynomials $f_x(ID_u, y)$, where x is randomly selected between 1 and w. Two nodes u and v establish a secret key if they know at least one polynomial of the same space (the same bi-variable polynomial).

16.12. Comparison of key management protocols in WSNs

This section proposes to compare the various approaches described in this chapter, according to the metrics defined in section 16.5.

16.12.1. *Type of key managed*

Most of the key management protocols manage pair-wise keys, cluster keys and group keys. Some of the protocols explicitly describe the establishment of one or more types of keys, while others cover the establishment of only one type of key, often the pair-wise keys. In general, they establish cluster keys and group keys on the basis of the established pair-wise keys. However, few protocols are interested in the problem of broadcast source authentication. Most of the protocols are based on µTESLA to authenticate a broadcast source, which is often the BS. However, µTESLA is not adapted to WSN for several reasons:

– µTESLA is well suited to a permanent broadcast source, like Broadcast TV. In the case of WSNs, the BS sends data occasionally and with little volume. Moreover, the capacity of storage and calculation of sensors and the need for immediate message authentication from the BS raise constraints on the duration of time intervals and the delay of key disclosure.

– The adaptation of µTESLA to a broadcast source different from the BS in WSNs is practically impossible because of the storage cost that would be required to load each node with the initial key $K_{g_i}^0$ of each node of the network.

– µTESLA can be subject to DoS attacks, since the messages broadcast (by the BS) are stored, then checked later on. An attacker can exploit this flaw by flooding network with packets, thus overloading the buffers of the sensors.

16.12.2. *Resulting network connectivity*

The probabilistic and hybrid approaches guarantee the maximum possible connectivity. However, it may happen that the key establishment phase does not lead to a fully connected network, because some nodes have not been able to establish keys with some of their neighbors, either directly or through a secure path. In this case, the network is divided into disjoint sub-networks, where secure communications within sub-networks are possible, but a secure communication between sub-networks is not possible. On the other hand, the deterministic approaches guarantee a full network connectivity after its deployment, because any two nodes of the network are able to establish a pair-wise key.

16.12.3. *Calculation cost*

Probabilistic protocols have the lowest calculation cost for the establishment of pair-wise keys. In [ESC 02] and [CHA 03] protocols, a secret key is established between two nodes without any calculation, if they have at least one common key. Then come the deterministic and hybrid protocols, where each node contributes to the creation of a pair-wise key with another node. In this case, there are three types of key establishment based on the cryptographic material in use:

– Usage of a shared primary key, like the [DUT 04] and [BHU 03]. The [BHU 03] protocol has a higher cost than [DUT 04] protocol, because the first uses the Diffie-Hellman protocol, whereas the latter uses only symmetric cryptography.

– Use of a secret matrix as in [BLO 84, DU 03, PRI 05]. This type of establishment is greedy in calculations.

– Use of symmetric bi-variable secret polynomials [BLU 98, LIU 03]. These protocols have a high cost because of the modular exponentiations.

In order to establish cluster keys, the majority of protocols are based on the generated pair-wise key, and involve the same costs, that is, the cost for cluster key encryption with each of the pair-wise key that is shared with each of its neighbors. However, the Bhuse *et al.* protocol has a higher cost due to the use of the Diffie-Hellman protocol.

The distribution of the group key is done with initialization before the deployment of the nodes. The renewal of the key occurs mainly after the revocation of a node, and can be done in three ways:

– Solution 1: the BS sends each valid node u the group key encrypted with $K_{SB,u}$.

– Solution 2: the use of cluster keys established to distribute the group key.

– Solution 3: the BS initially creates a hash tree where the leaves of the tree are the hash secret keys shared between the nodes and the BS, the root of the tree is the group key, and the value of each node of the tree is equal to the hash of its child values. Each node u receives the branch of the tree corresponding to its key $K_{SB,u}$. The renewal of the key after the revocation of a node is to remove the tree branch corresponding to u, to update the tree, and then to distribute each new value of the tree, including the root of the tree, encrypted with its child values. Assuming that we have a full binary tree of N nodes, the BS broadcasts $\log_2 N$ values, each value being encrypted with each one of its two child values.

The first solution has a high calculation cost for the BS and a high cost in transmission. The second solution has a low cost in calculation for the BS which only has to broadcast one encrypted message, but the time for propagating the key is longer, as the new group key is decrypted/re-encrypted at the entrance of each cluster. The third solution has an acceptable cost in calculation and transmission at the BS (logarithmic) unlike the first solution, but it implies a higher transmission cost than the second solution, because the BS broadcasts $2 \times \log_2 N$ encrypted values in the network instead of a single value, as in the case in the second solution.

16.12.4. *Storage cost*

The initial storage cost varies from one protocol class to another. In the probabilistic protocols, the number of keys to be stored depends on the desired connectivity [ESC 02] or on the size of the network [CHA 03]. However, both protocols are not optimized in terms of the memory space management, since many of the loaded keys will never be used.

In the deterministic class, this cost can correspond to the simple storage of a shared primary secret as in [DUT 04] and [BHU 03], or to the storage of $\lambda + 1$ secret elements, as in [BLO 84] and [BLU 98]. The λ value depends on the desired security level. The greater λ, the more resistant the WSN is to compromisation, but on the other hand the more the storage cost is important. The deterministic class manages the memory capacity effectively, since the stored data are all useful.

In the hybrid class, the storage cost depends on the protocol. For [PRI 05] a node stores a set of k random keys and a secret line of $\lambda + 1$ elements, and in [DU 03] a node stores a set of t secret lines of $\lambda + 1$ elements, while in [LIU 03] a node stores a set of t polynomials of degree λ. The value of the security parameter λ can be lower than that used in the deterministic class, while guaranteeing the same resistance to compromisings.

16.12.5. *Transmission cost*

In general, the deterministic class – except [BHU 03] – presents the lowest transmission costs , because two nodes do not need a discovery phase, as is the case for the hybrid class [PRI 05, DU 03, LIU 03] or the probabilistic protocol [ESC 02]. By sending only their identifier, as in [BLO 84, BLU 92] or the probabilistic protocol [CHA 03], two nodes can easily establish a secret key, with an additional transmission cost for [CHA 03] if the key is established through a path.

Thus, the [BHU 03] protocol has a high cost, because each node must exchange its public key and other data with each of its neighbors. Then come the protocols [ESC 02, PRI 05] where each node sends part or all the identifiers of the preloaded secret keys. Finally, there are protocols [DU 03, LIU 03] where each node sends space identifiers (matrices or bi-variable polynomials) to which its secret data refer.

16.12.6. *Security analysis*

Any proposed protocol is a compromise between security and performance. Better performance like moderate consumption of energy, computing and storage sacrifices the security level, and the reverse is also true.

All the key management protocols vary between two basic solutions in terms of performance and security:

– Use of only one group key, a solution that sacrifices security in favor of performance.

– Each node shares a unique secret key with each node of the network, a solution that sacrifices performance for the benefit of security.

All the proposed solutions aim to improve the security of the first solution and to reduce the storage cost of the second solution.

As described in section 16.5, the security metric measures the consequences of the compromising of a node, i.e. the ability of an attacker to deduce other secret keys it does not know initially, and its ability to populate the network with clones and fake nodes.

The security of deterministic approaches can vary greatly if the security threshold is exceeded. In [BLO 84] and [BLU 92], if the number of compromised nodes is less than or equal to the security threshold λ , an attacker cannot deduce the established keys between two non-compromised nodes, but once the threshold is exceeded, an attacker can calculate all the keys established in the network. Similarly, in [BHU 03] and [DUT 04], if the password P or the main keys bk_1, bk_2 are protected, an attacker cannot deduce the keys between two non-compromised nodes, but once these secrets are revealed by one node at least, an attacker can deduce all the keys established in the network. If the security threshold is not exceeded, in the [BLO 84] and [BLU 92] protocols, an attacker has the opportunity to populate any part of the network with clones and fake nodes and in [BHU 03], it can populate only the clusters for which it has the cluster keys. The ability of an attacker to populate the network is reduced in [DUT 04], because an attacker can only establish secret keys with nodes of a new generation.

The security of the probabilistic class depends on the protocol itself. In [ESC 02], a compromised node may reveal the secret keys established between non-compromised nodes, while in [CHA 03], a compromised node does not reveal any other established key in the network, because the keys are unique, contrary to [ESC 02] where a key can serve to secure multiple links. An attacker can populate any part of the network in [ESC 02, CHA 03] with clone nodes and fake nodes, but its capacity is reduced in [CHA 03] as each compromised key is unique in the network.

The hybrid class presents a better tolerance to compromisation compared to the other classes. In [PRI 05], even if $\lambda+1$ nodes are compromised, an attacker cannot systematically deduce all the keys established between the non-compromised nodes. First of all, the attacker must have recorded as a preliminary all the exchanges between the nodes of the network during the "discovery and key establishment" phase, a task which is very difficult to realize. Then, a node must be in possession of the preloaded 4-byte key shared between two non-compromised nodes, which constitutes the first half of the shared 8-byte secret key. Even if $\lambda+1$ nodes are compromised, the probability of recovering all the preloaded 4-byte secret keys is very low. In the same way, in [DU 03, LIU 03], by compromising $\lambda+1$ nodes, the probability that all these nodes know the secret information (lines or monovariable polynomials) resulting from the same space is very tiny, therefore an attacker will not be able to compromise the secret keys being established between valid nodes having some secret information resulting from the same space.

To limit the ability of an attacker to populate the network, Zhang *et al.* [ZHA 05] propose using the prediction of the unique physical location of a node in the calculation of secret information loaded in the node. Before establishing a secret key with a neighbor v, a node u checks that the position of v is within its coverage. Even if v lies about its position, by indicating a false position Pos' being in the coverage of u, v, v will not be able to calculate the secret key calculated by u, because it does not have secret information associated with Pos'. In this way, an attacker can only populate the vicinity of the nodes being in the vicinity of the compromised node.

16.12.7. *Scalability*

Scalability is a very important factor that determines whether a protocol is applicable to large networks, such as the WSN which can easily reach tens of thousands to hundreds of thousands of sensors. We deal here particularly to scalability in terms of memory needed.

In the probabilistic class, the number of keys preloaded in the nodes is related to the size of the network. The larger the network, the higher the number of preloaded keys increases in order to guarantee the maximum connectivity. However, it should be noted that [ESC 02] offers a better scalability than [CHA 03].

In the deterministic class, scalability depends on the protocol. In [BHU 03], and on the condition that the secret P is never revealed, whatever the size of the network, the nodes will only need to store P. In [DUT 04], if the size of the network increases but the number of sensor generations to be deployed also increases, the necessary storage space increases significantly for the nodes of the network. In [BLO 84, BLU 92], increasing the size of the network means an increase of the security parameter λ, thus implying more storage space in the nodes.

In the hybrid class [DU 03, LIU 03], increasing the size of the network has less impact on the storage space of the nodes than in the other two classes, because the value of the parameter λ does not increase in the same way as in [BLO 84, BLU 92]. The same remarks apply to [PRI 05] where the preloaded keys are 4 bytes long, contrary to [ESC 02, CHA 03] where the keys are 8 bytes long, and where an 8-byte key is only completed on demand.

16.13. Conclusion

WSN security received special attention from researchers over the past decade. Although the problem of key management in WSN has been investigated, the solutions proposed in the literature do not necessarily fit the characteristics and constraints of current sensors on the market. Because of their low storage and computation capacity, and their low energy autonomy, it is difficult to find a solution satisfying all these constraints. Using the key management protocols described in this chapter, we found that each protocol contributes to the problem of key management. Some protocols focus more on security at the expense of resources, others reduce the storage cost at the expense of energy consumption, the level of security or network connectivity, while others try to give a good security level at a reasonable storage cost, but suddenly introduce a high calculation cost.

The choice of a solution must take into account the capacity of sensors, the size of the network and its scalability, the level of risk that needs to be considered in the network as well as the reliability of transmissions in the network.

The use of public key cryptography based on elliptic curves could solve problems in the near future related to key management in WSNs. The first results [MAL 04, WAN 05] are quite encouraging and technological advances will produce better performing sensors with greater energy autonomy.

16.14. Bibliography

[AKY 02] AKYILDIZ I.F., SU W., SANKARASUBRAMANIAM Y., "Wireless sensor networks: a survey", *Computer Networks* 38, pp. 393–422, March 2002.

[BHU 03] BHUSE V., GUPTA A., PIDVA R., "A distributed approach to security in sensor-nets", *Vehicular Technology Conference*, October 6-9 2003, Orlando, Florida, USA, VTC 2003-Fall, IEEE 58th, Vol. 5, pp. 3010- 3014.

[BLO 84] BLOM R., "An optimal class of symmetric key generation", *Advances in Cryptography: Proc. of EUROCRYPT 84*, Lecture Notes in Computer Science, 209, Springer-Verlag, Berlin, pp. 335-338, 1984.

[BLU 92] BLUNDO R., SUNTIS A.D., HERZBEG A., KUTTEN S., VACCARO U., YUNG M., "Perfectly secure key distribution for dynamic conferences", in *Proc. of the 12th Annual International Cryptology Conference on Advances in Cryptology*, Santa Barbara, California, USA, Lecture Notes in Computer Science, Vol. N° 740, pp. 471-486, August 16-20, 1992.

[BLU 98] BLUNDO R., SUNTIS A.D., HERZBEG A., KUTTEN S., VACCARO U., YUNG M., "Perfectly secure key distribution for dynamic conferences", *Journal of Information and Computation*, Vol. 146, No. 1, 1998.

[CAM 04] CAMTEPE S.A., YENER B., *Key Distribution Mechanisms for Wireless Sensor Networks: a Survey*, Technical report TR 05-07, Rensselaer Polytechnic Institute, Computer Science Department, http://www.cs.rpi.edu/research/pdf/05-07.pdf, March 23, 2005.

[CHA 03] CHAN H., PERRIG A., SONG D., "Random key pre-distribution schemes for sensor networks", in *IEEE Symposium on Security and Privacy*, Berkeley, California, USA, May 2003, pp. 197-213.

[CHE 05] CHEN W. H., CHEN Y.J., *A Bootstrapping Scheme for Inter-Sensor Authentication within Sensor Networks,* IEEE COMMUNICATIONS LETTERS, Vol. 9, No. 10, October 2005.

[DH 05] DH Key-Exchange Protocols, https://www.cs.tcd.ie/courses/baict/bass/4ict11/Coursework/4ICT11MT6.2.pdf.

[DIF 76] DIFFIE W., HELLMAN M.E., "New directions in cryptography", *IEEE Transactions on Information Theory*, Vol. 22, No. 6, pp. 644-654, 1976.

[DU 03] DU W., DENG J., HAN Y.S., VARSHNEY P., "A pairwise key pre-distribution scheme for wireless sensor networks", *CCS*, October 27-30, 2003, Washington DC, USA.

[DUT 04] DUTERTRE B., CHEUNG S., LEVY J., *Lightweight Key Management in Wireless Sensor Networks by Leveraging Initial Trust*, SDL Technical Report SRI-SDL-04-02, April 6, 2004, www.csl.sri.com/users/bruno/publis/sri-sdl-04-02.pdf.

[ERD 60] ERDOS P., RÉNYI A., "On the evolution of random graphs", *Publ. Math. Inst. Hungar. Acad. Sci.*, pp. 17-61, 1960.

[ESC 02] ESCHENAUER L., GLIGOR V.D., "A key management scheme for distributed sensor networks", in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 41-47, Washington DC, USA.

[HAB 06] HABBANI A., ROMAIN O., EL ABBADI J., GARDA P., *Réseaux de capteurs : Système d'acquisition et de routage d'information*, e-TI, e-revue en Technologies de l'Information, http://www.revue-eti.net, April 2006.

[HAR 04] HARTUNG C., BALASALLE J., HARN R., *Node Compromise in Sensor Networks: The Need for Secure Systems*, Technical Report CU-CS-990-05, Dept of Comp Sci, Univ of Colorado at Boulder, January 2005.

[HWA 04] HWANG J., KIM. Y., "Revisiting random key pre-distribution schemes for wireless sensor networks", *SASN'04*, October 25, 2004, Washington DC, USA.

[LIU 03] LIU D., NING P., "Establishing pairwise keys in distributed sensor networks", *CCS*, October 27-31, 2003, Washington, DC, USA.

[MAL 04] MALAN D.J., WELSH M., SMITH M.D., "A public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography", *The First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp. 71-80, October 2004, Santa Clara, California, USA.

[PER 00] PERRIG A., CANETTI R., TYGAR J.D., SONG D., "Efficient authentication and signing of multicast streams over lossy channels", in *IEEE Symposium on Security and Privacy*, pp. 56-73, May 2000, Oakland, California, USA.

[PER 01] PERRIG A., SZEWCZYK R., WEN V., CULLAR D., TYGAR J.D., "Spins: security protocols for sensor networks", in *Proc. of the 7th Annual ACM/IEEE ICMCN*, pp. 189-199, July 2001, Rome, Italy, 2001.

[PRI 05] PRICE A., KOSAKA K., CHATTERJEE S., "A secure key management scheme for sensor networks", in *Proceedings of the 10th Americas Conference on Information Systems*, New York, August 2004.

[WAN 05] WANDER A. S., GURA N., EBERLE H., GUPTA V., SHANTZ S. C., "Energy analysis of public-key cryptography for wireless sensor networks", in *Proc. of the 3rd IEEE ICPCC*, pp. 324-328, March, 8-12 2005, Kauai Island, Hawaii.

[ZHA 05] ZHANG Y., LIU W., LOU W., FANG Y., "Securing sensor networks with location-based keys", *Proceedings of the IEEE WCNC'05*, pp. 1909-1914, 13-17 March 2005, New Orleans, Louisiana, USA.

[ZHU 03] ZHU S., SETIA S., JAJODIA S., "LEAP: efficient security mechanisms for large scale distributed sensor networks", *Proceedings of the 10th ACM Conference on CCS*, pp. 62–72, October 27-30, 2003, Washington DC, USA.

Conclusion

This book, divided into three parts, has tried to gather various works around the security of mobile and wireless networks. Part 1, “Basic Concepts”, provided a survey of mobile and wireless networks, and the foundations of security. It reviewed various technologies with a focus on vulnerabilities and security solutions. Part 2, “Off-the-Shelf Technologies”, provided the related security of the current mobile and wireless technologies. Finally, Part 3, “Emerging Technologies”, provided more research work on security in emerging wireless technologies. As such, this book showed that each technology poses its own challenges in the design of security solutions. Compared to wired technologies, the radio resource is easy to spy on, so that security and wireless communication might be seen as an oxymoron. Also, mobile terminals need to have robust and lightweight security solutions.

Wireless networks, which by their nature, facilitate access to the radio, are more vulnerable than wired networks and need to encrypt communications to deal with sniffing and continuously checking the identity of the mobile nodes. The mobility factor adds more challenges to security, namely monitoring and maintenance of secure traffic transport of mobile nodes. This concerns both homogenous and heterogenous mobility (inter-technology), the latter requires homogenization of the security level of all networks visited by the mobile.

According to the network architecture, either it is infrastructure-based (fixed access points) or infrastructure-less (ad hoc and sensor networks). Ensuring a reliable and secure routing and also maintaining a level of trust between the nodes of the network are essential for the continuation of service deployment over these networks.

Written by Hakima CHAOUCHI.

From the terminal's side, it is important to protect its resources (battery, disk, CPU) against misuse and ensure the confidentiality of its data. In an ad hoc or sensor network, it becomes essential to ensure terminal's integrity as it plays a dual role of router and terminal.

The difficulty of designing security solutions that could address these challenges is not only to ensure robustness faced with potential attacks or to ensure that it does not slow down communications, but also to optimize the use of resources in terms of bandwidth, memory, battery, etc. More importantly, in this open context the wireless network is to ensure anonymity and privacy, while allowing traceability for legal reasons. Indeed, the growing need for traceability is now necessary for the fight against criminal organizations and terrorists, but also to minimize the plundering of copyright. It is therefore facing a dilemma of providing a network support of free exchange of information while controlling the content of the communication to avoid harmful content. Actually, this concerns both wired and wireless networks. All these factors influence the selection and implementation of security tools that are guided by a prior risk assessment and security policy.

Finally, we are increasingly thinking about trust models in the design of secured systems, that should offer higher level of trust than classical security mechanisms, and it seems that future networks should implement both models: security and trust models.

In fact, if communication nodes will be capable of building and maintaining a predefined trust level in the network, then the communication system will be trustable all the time, thus allowing a trusted and secure service deployment. However, such trust models are very difficult to design and the trust level is generally a biased concept presently. It is very similar to the human based trust model. Note that succeeding in building such trust models will allow infrastructure based networks but especially infrastructure-less or self-organized networks such as ad hoc sensors to be trusted enough to deploy several applications. This will also have an impact on current business models where the economic model would have to change in order to include new players in the telecommunication value chain such as users offering their machines to build an infrastructure-less network. For example, in the context of ad hoc networks, we could imagine that ad hoc users become distributors of content or provide any other networked services[1], being a sort of service providers. In this case, an appropriate charging and billing system needs to be designed.

[1] Patent 2007, INPI n° 0756559, France: H. Chaouchi, M. Maknavicius "AAA architecture in ad hoc networks".

Other consequences of having robust trust models might concern network operators that wish to rent their nodes to support new services from service providers, for instance, dynamically allowing the installation of new elements of code on their routers to offer a service to users. Another example is to offer the possibility of a network operator subcontracting certain network control and management features of its network. This will definitely open new possibilities for new players and new technologies to join the telecommunication value chain in this ever-growing market.

This page intentionally left blank

List of Authors

Tara ALI YAHIYA
LIP6
Paris
France

Cuihtlauac ALVARADO
FTR&D MAPS/AMS
France Télécom R&D
Lannion
France

Chakib BEKARA
LOR
GET
Institut national
des télécommunications
Evry
France

Christian BONNET
Communications mobiles
EURECOM
Sophia Antipolis
France

Julien BOURNELLE
LOR
GET
Institut national
des télécommunications
Evry
France

Laurent BUTTI
France Télécom R&D
Network and Service Security
Issy-les-Moulineaux
France

Ana CAVALLI
LOR
GET
Institut national
des télécommunications
Evry
France

Hakima CHAOUCHI
LOR
GET
Institut national
des télécommunications
Evry
France

Isabelle CHRISMENT
LORIA-ESIAL
Nancy University
Vandoeuvre-lès-Nancy
France

Jean-Michel COMBES
Network and Service Security
France Télécom R&D
Issy-les-Moulineaux
France

Pierre CRÉGUT
FTR&D MAPS/AMS
France Télécom R&D
Lannion
France

Sérgio DE OLIVEIRA
UFMG/UNIPAC
Belo Horizonte
Brazil

Olivier FESTOR
LORIA
INRIA Lorraine
Villers-lès-Nancy
France

Franck GILLET
LOR
GET
Institut national
des télécommunications
Evry
France

Jérôme HÄRRI
Communications mobiles
EURECOM
Sophia Antipolis
France

Artur HECKER
INFRES
GET
Ecole nationale supérieure
des télécommunications
Paris
France

Maryline LAURENT-MAKNAVICIUS
LOR
GET
Institut national
des télécommunications
Evry
France

Antonio A.F. LOUREIRO
UFMG
Belo Horizonte
Brazil

Daniel MIGAULT
Network and Service Security
France Télécom R&D
Issy-les-Moulineaux
France

Mihai MITREA
ARTEMIS
GET
Institut national
des télécommunications
Evry
France

José-Marcos NOGUEIRA
UFMG
Belo Horizonte
Brazil

Jean-Marie ORSET
LOR
GET
Institut national
des télécommunications
Evry
France

Olivier PAUL
LOR
GET
Institut national
des télécommunications
Evry
France

Guy PUJOLLE
LIP6
Paris
France

Françoise PRÊTEUX
ARTEMIS
GET
Institut national
des télécommunications
Evry
France

Isabelle RAVOT
Fraude et Assurance des Revenus
France Télécom
Lausanne
Switzerland

Jérôme RAZNIEWSKI
France Télécom R&D
Network and Service Security
Issy-les-Moulineaux
France

Ana Paula RIBEIRO DA SILVA
UFMG
Belo Horizonte
Brazil

Mohamed SALAH BOUASSIDA
LORIA
Henry Poincaré University
Nancy
France

Fernando A. TEIXEIRA
UFMG
Belo Horizonte
Brazil

Pascal URIEN
INFRES
GET
Ecole nationale supérieure
des télécommunications
Paris
France

Franck VEYSSET
France Télécom R&D
Network and Service Security
Issy-les-Moulineaux
France

Hao-Chi WONG
Palo Alto Research Center
USA

This page intentionally left blank

Index