DOTTORATO DI RICERCA IN INFORMATICA
IX CICLO
UNIVERSITA' DEGLI STUDI DI SALERNO

Secure End-to-End Communications in Mobile Networks

Fabio Petagna

November, 2010

PhD Program Chair
Prof.ssa
Margherita Napoli

Supervisor
Prof.
Alfredo De Santis

1. PhD program chair:
   Prof.ssa Margherita Napoli

2. PhD Committee:
   Prof. Alfredo De Santis,
   Dott. Marco Faella,
   Prof. Domenico Talia

3. Supervisor:
   Prof. Alfredo De Santis

Day of the defense: April 29th, 2011

Abstract

Cellular communication has become an important part of our daily life. Besides using cell phones for voice communication, we are now able to access the Internet, conduct monetary transactions, send voice, video and text messages and new services continue to be added. The frequencies over which voice is transmitted are public, so voice encryption is necessary to avoid interception of the signal over the air. But once the signal reaches the operators Base Station (BS), it will be transmitted to the receiver over a wired or wireless mean. In either case, no protection is defined. This does not seem a problem, but this is not true. Along the path across operator network, voice is at risk. It will only be encrypted again, with a different key, from the BS to the receiver if the receiver is herself a mobile user. Moreover, voice encryption is not mandatory. The choice whether or not to accept an unprotected communication is up to the network. When adopted, the same encryption algorithm is used for sending SMS messages between mobile telephones and base stations and for encrypting of calls. Unfortunately, vulnerabilities in this encryption systems were already revealed more than 10 years ago and more continue to be discovered.

Currently the most popular communication technologies are the GSM and the UMTS. The UMTS is in use as a successor to GSM. Along with mobile phone services, It provides rapid data communication. The security algorithms in UMTS differs from GSM in two important ways: encryption and mutual authentication. Although security standards have been improved, the end- to-end security is not provided.

At the time of this writing, the user who makes a call with another, remote, user by means of a mobile network, is subject to several threats: phishing, session hijacking and eavesdropping are such examples. This is mainly due

to the lack of either mechanisms for mutual authentication of the end point of the conversation and the end-to-end digital encryption of the content of the conversation.

In this Thesis we first give an overview about several generations of cellular networks such as GSM and 3G and discuss their security issues and different types of attacks. Then, we introduce novel end-to-end security systems SPEECH and SEESMS for voice and text communications, respectively. Finally, we move our attention to video-communication over UMTS. In particular, we propose an end-to-end security systems for videotelephony named SECR3T.

SPEECH is a software for making secure calls by using a Windows Mobile powered handheld device and the GSM data communication channel. The notion of security implemented by SPEECH is stronger than that available in other secure conversation software, it includes the mutual authentication of the end point of a conversation, the end-to-end digital encryption of the content and the possibility to digitally sign the content for non-repudiation purpose.

SEESMS is a software framework written in Java which allows two peers to exchange encrypted and digitally signed SMS messages. The communication between peers is secured by using public-key cryptography. The key-exchange process is implemented by using a novel and simple security protocol which minimizes the number of SMS messages to use.

SECR3T is a full fledged secure communication system for mobile devices based on the Circuit Switched Domain (CSD, in brief) of 3G networks. The use of CSD is the most innovative contribute of the project since, to the best of our knowledge, no solution is available for such a channel. Similarly to the SPEECH project, the notion of Security implemented by SECR3T includes the mutual endpoint authentication and the end-to-end channel encryption. A SECR3T packet is recognized by the network as a frame of the native videotelephony protocol which is 3G-324M. This allows a SECR3T client to communicate also with a different video-client in a non-secured mode.

Contents

List of Figures

List of Tables

1

Introduction

For the majority of people cellular communication is a fundamental part of daily life. Besides using cell phones for voice communication, people are now able to access the Internet, conduct monetary transactions, send text messages and many other new services continue to be added day by day. However the wireless medium has, as opposed to the wired one, many limitations such as open access, limited bandwidth and systems complexity. These limitations make it challenging, although still possible, to provide security features such as authentication, integrity and confidentiality. Moreover, the fact that the packet switched core of the current generation of 3G networks is connected to external networks such as the Internet, make wireless communication vulnerable to several types of attacks such as denial of service, viruses, worms etc. which have already been used against hosts over the Internet. Therefore, it is important to provide users with a secure channel for communication.

In this Thesis, we design and develop three communication systems to securely carry out audio, text and video conversation.

This chapter gives an overview about the various generations of cellular networks. For those not familiar with the cellular network architecture, a brief description of the new 3G cellular network architecture is provided. Limitations of cellular networks, their security issues and the different types of attacks are also discussed. The last part of the chapter provides a brief introduction to our secure communication systems: SPEECH, SEESMS and SECR3T.

1.1 Generations of cellular networks

Cellular Networks have been around since the 1980s and each year their subscribers increase at a very fast rate. First generation (1G) networks were the first cellular networks introduced in the 1980s. They were only capable of transmitting voice at speeds of about 9.6 kbps max. In the US the system was known Advanced Mobile Phone System (AMPS) and in Europe the Nordic Mobile Telephony (NMT). Both these technologies used analog modulation to transmit data as a continuously varying waveform.

1G systems had some limitations such as no support for encryption, poor sound quality and inefficient use of the spectrum due to their analog nature. Second generation (2G) cellular networks also known as personal communication services (PCS) introduced the concept of digital modulation meaning that voice was converted into digital code, and then into analog (radio) signals. Being digital, they overcame certain limitations of 1G systems. Various 2G technologies have been deployed around the world. Code Division Multiple Access (CDMA), North American Time Division Multiple Access (NA-TDMA) and digital AMPS (D-AMPS) have been deployed in the US whereas Global System for mobile communication (GSM) has been deployed in Europe and USA and Personal Digital Cellular (PDC) has been deployed in Japan.

Although 2G systems were a great improvement from 1G, they were only used for voice communication. 2.5G is a transition step between 2G and 3G and it is also known as data services over 2G. There have been several deployments of 2.5G across the world. In the USA, they are known as 1xEV-DO and 1xEV-DV. In Europe or places where GSM has been used, 2.5G technologies such as High Speed circuit switched data (HSCSD), General packet Radio Service (GPRS), Enhanced Data Rate for GSM Evolution (EDGE) have been deployed.

The Third generation (3G) standard is currently being deplyed as the next global standard for cellular communications. It provides services such as fast Internet surfing video telephony. There are three main technologies that are being applied. In the US CDMA2000, in Europe Wideband CDMA (W-CDMA) and in China Time Division-Synchronous Code Division Multiple Access (TD-SCDMA). Although 3G has not been fully deployed, people have already started talking about the fourth generation (4G) technology. This generation will be designed to have data rates of up to 20Mbps. It will

have support for next generation Internet such as IPv6, QoS and Mo-IP, lower system cost and high capacity and capable of supporting communication in moving vehicles with speed up to 250 km/hr.

1.1.1 2G and 2.5G

GSM is the most widely adopted 2G technology in the world. Although it was initially employed in Europe, it has become a global technology with subscribers in about 197 countries. Its specifications were completed in 1990 and service began in 1992. This chapter will not explain the implementations details of 2G/2.5G as long as it is out of the scope of the Thesis, interested readers are encouraged to look at (1) for more details. However, some of the data services which are part of the 2.5G extension are:

- Short Messaging Service (SMS): Transfer of messages between cell phones. Large messages are truncated and sent as multiple messages.
- High-Speed Circuit-Switched Data (HSCSD): This was the first attempt at providing data at high speeds data over GSM, with speeds of up to 115 kbps. This technique cannot support large bursts of data. HSCSD was not widely implemented and GPRS became a more popular technique.
- General Packet Radio Service (GPRS): This technique can support large bursty data transfers.
- Enhanced Data Rates for GSM Evolution (EDGE): The standard GSM uses GMSK modulation. Edge uses 8-PSK modulation. GPRS and EDGE combined provide data rates of up to 384 kbps.
- Cellular Digital Packet Data (CDPD): CDPD is a packet based data service. CDPD is able to detect idle voice channels and uses them to transfer data traffic without affecting voice communications.

CDMA is the primary 2G technology in the USA. CDMAOne, also known as IS-95a was the initial technique. This technique allows users to use the entire spectrum and can support more users than TDMA and GSM. Speed between 4.8 and 14.4 kbps can be supported. The CDMATwo extension can provide data rates of up to 115.2 kbps. The 2.5G extension to this technology can be divided into two techniques. 1xEV-DV

uses one radio frequency channel for data and voice, whereas 1xEV-DO uses separate channels for data and voice. These are fully compatible with both CDMAOne and its 3G replacement CDMA2000, to make the transition as easy as possible.

1.1.2 3G

3G is the next generation wireless cellular network whose aim is to provide a world wide standard and a common frequency band for mobile networking. The International Telecommunication Union (ITC) started the process in 1992, the result of this effort was a new network infrastructure called International mobile telecommunications 2000 (IMT- 2000), with the 2000 signifying that this new technology would be available in 2000. Application services include wide-area wireless voice telephone, mobile Internet access, video calls and mobile TV, all in a mobile environment. To meet the IMT-2000 standards, a system is required to provide peak data rates of at least 200 kbit/s. The following is the list of objectives that IMT-2000 aims to receive (2)

1. To make a wide range of services, both voice and data available to users, irrespective of location.
2. To provide services over a wide coverage area.
3. To provide the best quality of service (QoS) possible.
4. To extend the number of services provided subject to constraints like radio transmission, spectrum efficiency and system economics.
5. To accommodate a great variety of mobile stations.
6. To admit the provision of service by more than one network in any area of coverage.
7. To provide an open architecture which will permit the easy introduction of technology advancements as well as different applications.
8. To provide a modular structure which will allow the system to start from small and simple configuration and grow as needed, both in size and complexity within practical limits.

The 3rd generation partnership project(3) was formed in 1998 to produce specifications for UMTS, a 3G technology based on Universal Terrestrial Radio Access (UTRA) radio interface and the extended GSM/GPRS network. A second radio interface also exists called IMT Multicarrier (IMT-MC) which is being promoted by the 3GPP2 organization. This interface is backward compatible with IS-95 to make a seamless transition to 3G. This proposal is known as CDMA2000.

1.1.2.1 3G - UMTS Architecture

To understand the threats to a network, one must understand the network infrastructure. UMTS is considered the most important 3G proposal. It is being developed as an evolution of GSM and therefore based on the GPRS network which is a 2.5G technology and the UTRA radio interface.

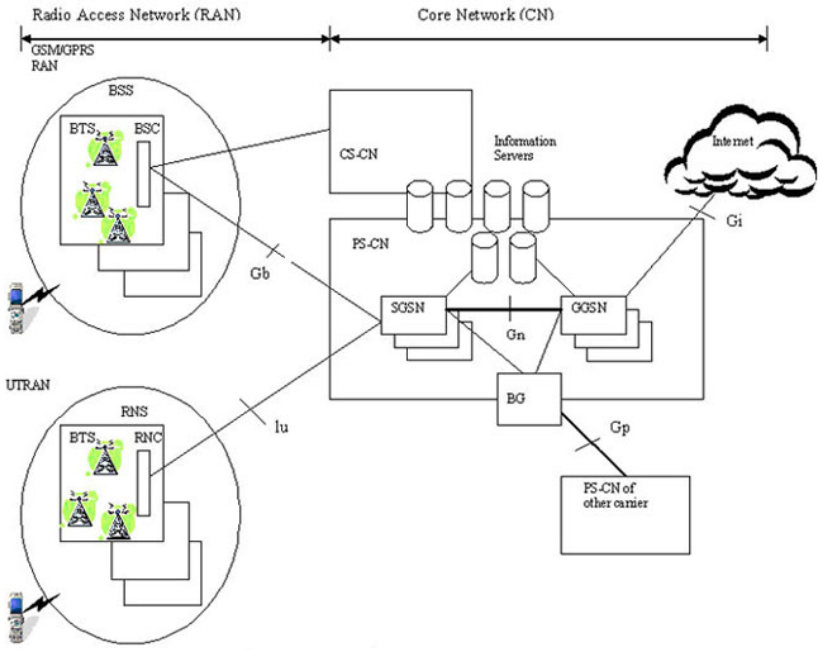

Figure 1.1: 3G network architecture

As can be seen in Figure 1.1 (4), the 3G network has two main parts

1. The Radio Access Network (RAN)

2. The Core Network (CN)

The RAN consists of the existing GPRS/GSM RAN system which is connected to the Packet Switched Network (PS-CN) and also to the circuit switched network (CS-CN). The PS-CN will eventually connect to the UTRAN system as part of the full transition to 3G. The UTRAN consists of subsystems, with each subsystem consisting of one Radio Network Controller (RNC) which is connected to several Base Transceiver Stations (BTN). The GRPS RAN has a similar architecture.

The Core Network consists of the PS-CN and the CS-CN. The PS-CN consists of several information servers, the SGSN and the GGSN. Each SGSN connects one or more RSC and BSC with the PS-CN. Its functionality includes access control, mobility management, paging and route management (4). The GGSN is the logical gateway to the Internet. The BG interface can be used to connect to another PS-CN or to another carrier. The information servers provide several functions. The Home Location Register (HLR) maintains subscriber information and the Authentication Center (AuC) maintains authentication information. There are also IP based servers such as DNS, DHCP and RADIUS servers which interact with the SGSN/GGSN and provide control and management functions.

1.2 Security issues in cellular networks

The infrastructure for Cellular Networks is massive, complex with multiple entities coordinating together, such as the IP Internet coordinating with the core network. For these reasons, providing security on every possible network communication path is a real challenge.

1.2.1 Limitations of wireless networks

Compared to Wired Networks, Wireless Cellular Networks have a lot of limitations.

1. Open Wireless Access Medium: Since the communication is on the wireless channel, there is no physical barrier that can separate an attacker from the network.

2. Limited Bandwidth: Although wireless bandwidth is increasing continuously, because of channel contention everyone has to share the medium.

3. System Complexity: Wireless systems are more complex due to the need to support mobility and making use of the channel effectively. By adding more complexity to systems, potentially new security vulnerabilities can be introduced.

4. Limited Power: Wireless Systems consume a lot of power and therefore have a limited time battery life.

5. Limited Processing Power: The processors installed on the wireless devices are increasing in power, but still they are not powerful enough to carry out intensive processing.

6. Relatively Unreliable Network Connection: The wireless medium is an unreliable medium with a high rate of errors compared to a wired network.

1.2.2 Security threats in cellular networks

There are several security issues that have to be taken into consideration when using the cellular infrastructure for communications.

1. Authentication: Cellular networks have a large number of subscribers, and each has to be authenticated to ensure the right people are using the network. Since the purpose of 3G is to enable people to communicate from anywhere in the world, the issue of cross region and cross provider authentication becomes an issue.

2. Integrity: With services such as SMS, chat and file transfer it is important that the data arrives without any modifications.

3. Confidentiality: With the increased use of cellular phones in sensitive communication, there is a need for a secure channel in order to transmit information.

4. Access Control: The Cellular device may have files that need to have restricted access to them. The device might access a database where some sort of role based access control is necessary (5).

5. Operating Systems In Mobile Devices: Cellular Phones have evolved from low processing power, ad-hoc supervisors to high power processors and full fledged operating systems. Some phones may use a Java Based system, others use Microsoft Windows CE and have the same capabilities as a desktop computer. Issues may arise in the OS which might open security holes that can be exploited.

6. Web Services: A Web Service is a component that provides functionality accessible through the web using the standard HTTP Protocol. This opens the cellular device to variety of security issues such as viruses, buffer overflows, denial of service attacks etc. (6)

7. Location Detection: The actual location of a cellular device needs to be kept hidden for reasons of privacy of the user. With the move to IP based networks, the issue arises that a user may be associated with an access point and therefore their location might be compromised.

8. Viruses And Malware: With increased functionality provided in cellular systems, problems prevalent in larger systems such as viruses and malware arise. The first virus that appeared on cellular devices was Liberty. An affected device can also be used to attack the cellular network infrastructure by becoming part of a large scale denial of service attack.

9. Downloaded Contents: Spyware or Adware might be downloaded causing security issues. Another problem is that of digital rights management. Users might download unauthorized copies of music, videos, wallpapers and games.

10. Device Security: If a device is lost or stolen, it needs to be protected from unauthorized use so that potential sensitive information such as emails, documents, phone numbers etc. cannot be accessed.

1.2.3 Types of attacks

Due to the massive architecture of a cellular network, there are a variety of attacks that the infrastructure is open to.

1. Denial Of Service (DOS): This is probably the most potent attack that can bring down the entire network infrastructure. This is caused by sending excessive data to the network, more than the network can handle, resulting in users being unable to access network resources.

2. Distributed Denial Of Service (DDOS): It might be difficult to launch a large scale DOS attack from a single host. A number of hosts can be used to launch an attack.

3. Channel Jamming: Channel jamming is a technique used by attackers to jam the wireless channel and therefore deny access to any legitimate users in the network.

4. Unauthorized Access: If a proper method of authentication is not deployed then an attacker can gain free access to a network and then can use it for services that he might not be authorized for.

5. Eavesdropping: If the traffic on the wireless link is not encrypted then an attacker can eavesdrop and intercept sensitive communication such as confidential calls, sensitive documents etc.

6. Message Forgery: If the communication channel is not secure, then an attacker can intercept messages in both directions and change the content without the users ever knowing.

7. Message Replay: Even if communication channel is secure, an attacker can intercept an encrypted message and then replay it back at a later time and the user might not know that the packet received is not the right one.

8. Man In The Middle Attack: An attacker can sit in between a cell phone and an access station and intercept messages in between them and change them.

9. Session Hijacking: A malicious user can highjack an already established session, and can act as a legitimate base station.

1.3 Secure communications systems

In this Thesis, three software systems for end-to-end Security are described. They have been designed at the extent of addressing the security issues in cellular networks described in Section 1.2.2 with particular attention to user authentication, eavesdropping and non-repudiation of the communication content. The three project are called SPEECH, SEESMS and SECR3T. All of them are end-to-end security oriented and run at application level both on mobiles devices and PC. For this reason they are not able to resist to all the attack described in Section 1.2.3, infact, DOS and channel Jamming attacks which are strictly related to the physical network, cannot be faced by those software systems. Instead, attacks such as message forgery, replay, MITM and eavesdropping can be efficiently withstood by those tools.

1.3.1 SPEECH

SPEECH (Secure Personal End-to-End Communication with Handheld) is a software system for making secure calls by using a Windows Mobile powered handheld device and a wireless data communication channel. The security mechanisms included in SPEECH cover the mutual authentication of the end point of a conversation, the end-to-end digital encryption of the content of a conversation and the possibility to digitally sign the conversation content for non-repudiation purpose.

SPEECH is able to operate on different types of network and adapt its behavior to the bandwidth of the underlying network while guaranteeing a minimal-acceptable quality of the service (currently GSM and TCP/IP networks are supported). This has been achieved by adopting a very light communication protocol and by using a software codec explicitly optimized for the compression of voice data streams while retaining a good sampling quality. As a result, SPEECH is able to work in full-duplex mode, with just a slight delay in the conversation, even when using a 9600 bps CSD communication channels, such as the one provided by 2G GSM networks.

There are several application areas for SPEECH. For example, it can be used in an economic transaction conducted over a public phone line to verify the real identities of the parties who are participating to the transaction. Moreover, it guarantees against the possibility for an eavesdropper to access the content of the conversation. Finally, the non-repudiation feature ensures that either party of the call could not deny the content of the conversation in a later moment.

1.3.2 SEESMS

SEESMS (Secure Extensible and Efficient SMS) is a software framework written in Java which allows two peers to exchange encrypted and digitally signed SMS messages. The communication between peers is secured by using public-key cryptography. The key-exchange process is implemented by using a novel and simple security protocol which minimizes the number of SMS messages to use. SEESMS supports the encryption of a communication channel through the ECIES and the RSA algorithms. The identity validation of the contacts involved in the communication is implemented through the RSA, DSA and ECDSA signature schemes. Additional cryptosystems can be coded and added to SEESMS as plug-ins.

Special attention has been devoted to the implementation of an efficient framework in terms of energy consumption and execution time. To this end, an experimental analysis was conducted to determine which combination of cryptosystems and security parameters were able to provide a better trade-off in terms of speed/security and energy consumption. This experimental analysis has also been useful to expose some serious performance issues affecting one of the cryptographic libraries that is commonly used to implement security features on mobile devices. These issues have been tackled by profiling the code of these libraries and determining the reasons of these bad performance. Then, the performance of this library has been improved by implementing some algorithmic and programming optimization techniques. The resulting code exhibits a significant performance boost with respect to the original implementation, and requires less memory in order to be run.

1.3.3 SECR3T

Voice and video communication tools are considered unreliable when used in mobile context or in poor signal strength conditions. This is particularly true for IP connections when routed on the Packet-Switched Domain (PSD) over 3G mobile networks.

The SECR3T (Secure End-to-End Communication over 3G Telecommunication Networks) project aimed to give a solution to handle the lack of communication tools explicitly designed to increase the security level over the Circuit Switched Domain(CSD) of 3G networks. SECR3T is a full fledged secure communication system for mobile devices based on the native CS Domain of 3G networks. The use of CSD is the most innovative contribute of the project since, to the best of our knowledge, no solution is available for such a channel. Similarly to the SPEECH project, the notion of Security implemented by SECR3T includes the mutual endpoint authentication and the end-to-end channel encryption.

Using SECR3T users are able to authenticate the peer either by means of X.509 digital certificates or by a pre shared passphrase. The adopted end-to-end security mechanisms have been embedded within the native 3G-324M protocol and they have been proved to be transparent to the network operators. A SECR3T packet is therefore recognized by the network as a frame of the native videotelephony protocol 3G-324M. This allows a SECR3T client to communicate also with a different video-client in a non-secured mode.

Relying on the CSD, SECR3T provides a better QoS with respect to the PSD based solution for 3G networks and it requires less power consumption as the user is registered once on the base station instead of leaving the handset to implement heavy keep alive protocols.

2

Mobile networks security

2.1 GSM Security

GSM is the second generation technology for mobile phone communications. It was accepted as the international standard for digital cellular telephony in the late 1980s, and it started to be deployed in the 1990s. GSM replaced first generation cellular phone systems, which were analog systems that could support only a limited number of users. Two of the major systems that were in existence were the advanced mobile phone system (AMPS), the standard chosen in the United States, and total access communications system (TACS), mainly deployed in Europe. The downfall of first generation systems was the need for greater capacity as well as a technology that could support international communications. Static and cross-channel interference are major annoyances with analog phones while nonexistent with digital. Last but not least, security and privacy can be easily implemented on digital networks through encryption methods.

Today GSM is one of the most widely deployed digital cellular telephone systems in the world. Competing technologies are the United Statesdeveloped CDMA and time division multiple access (TDMA). Although these technologies are intrinsically incompatible, many phones today support multiple technologies, and mobile telephone companies have made agreements to allow users to call and be reached independently of the service offered in their coverage area.

When GSM was conceived and standardized by the European Telecommunications Standards Institute (ETSI), two security services were targeted: authentication and

encryption. Since the goal for authentication was to allow the telephone company to identify the user for billing purposes, only one-way authentication was requested, whereas trust in the network was considered implicit. Encryption was designed to protect the air link between the mobile user and the telephone operator ground antenna, while communication confidentiality within the operator's network was not taken into account.

GSM security relies on symmetric key cryptography, a secret key Ki is shared between the user and the network operator. This key is the secret used to perform the authentication protocol and to calculate session encryption keys. If the value of this key is revealed, an attacker may impersonate a victim as well as eavesdrop on his conversations; for this reason, the standardization committee decided to store this key in a smart card, which is a tamper-resistant device. Smart cards used in GSM are called SIM cards. Besides storing the secret key Ki, the SIM card also provides a protected environment within which sensitive cryptographic operations are performed. Every SIM card is personalized (i.e., it contains the unique user identification IMSI code and secret key Ki). A SIM card can be moved from one handset to another without the user having to change his subscription contract or his telephone number.

Another service offered by GSM is user anonymity for privacy protection. An IMSI is a nonconfidential value linked to a particular user. IMSI knowledge would allow identifying the user's physical location worldwide, while use of a temporary identity, known as temporary mobile subscriber identity (TMSI), can provide anonymity. The TMSI is frequently updated (every time the user moves to a new location area or after a certain time period) to avoid linking user information with TMSI. There are situations where IMSI use is mandatory (e.g., on the first use of the mobile after purchase, on the first use of the mobile under the coverage of another operator, or whenever the provided TMSI cannot allow establishment of user identity).

ETSI defined three algorithms to achieve authentication, encryption, and encryption key derivation from Ki. Input and output lengths as well as key lengths were specified. Algorithm choice for authentication was left open, whereas encryption algorithms must be standard to allow for roaming between operators.

Reference algorithms were designed by ETSI and kept secret from the public for operator use only. A3 allows calculating the response $SRES$ to a challenge $RAND$ sent by the network operator to the user. A8 uses the same $RAND$ as input to calculate

the encryption key *Kc*. A5 is the voice encryption algorithm. Algorithms A3 and A8 are often combined in a single algorithm referred to as A3/A8, the use of which is to calculate the challenge response *SRES* and encryption key *Kc* given the secret key *Ki* and the random input *RAND*. Besides reference A3/A8 algorithms, operators have the option to implement proprietary algorithms, or published algorithms that fit the requested characteristics. The mobile phone and the visited network must support the same A5 algorithm to allow encrypted voice communication.

GSM security is based on triplets, including

- The challenge *RAND*
- The challenge response *SRES*
- The voice encryption key *Kc*

The algorithms to calculate *SRES* and *Kc* from *RAND* and the secret key *Ki* are described in the following sections.

2.1.1 User authentication

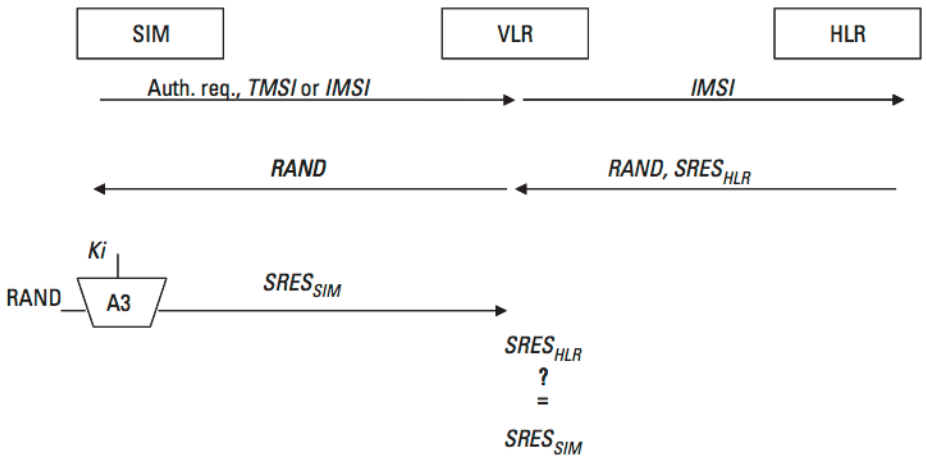

Figure 2.1: GSM authentication

User authentication is achieved by performing a challenge response between the network and the user. (See Figure 2.1) To be more specific, user authentication occurs in the SIM card whereas network authentication occurs in the authentication center

(AuC) or the home local register (HLR). If the mobile user is in a visited local register (VLR) coverage area (i.e., an area under the coverage of an operator that has roaming agreements with the operator the user subscribed to), the AuC or HLR transfer the authentication results, success or failure, to the VLR.

Network authentication is not required in GSM, as the cost to build a rogue network station was considered sufficiently prohibitive to put off potential attackers.

The algorithm A3 is implemented in the SIM card and the AuC or HLR. To authenticate, the user receives a 128 bit random challenge *RAND* from the network. Using the 128-bit secret key *Ki* algorithm, A3 computes a 32-bit challenge response signed response (SRES) and transmits it to the network for verification.

The authentication procedure is outlined in the following steps:

1. Authentication is initiated by the user whenever he wants to make a call from his mobile (MS) or go on standby to receive calls. The user transmits his identity through TMSI and the authentication request.

2. The network establishes the identity of the SIM through the 5-digit TMSI. If the TMSI is recognized, the VLR sends a request for authentication to the HLR; if not, it will request the user's IMSI.

3. The HLR generates a 128-bit random *RAND* challenge. Using the user *Ki* and *RAND*, it applies A3 and calculates the expected $SRES_{HLR}$. RAND and $SRES_{HLR}$ are both sent to the VLR.

4. The VLR sends RAND to the SIM.

5. The SIM calculates the $SRES_{SIM}$, using its secret key *Ki* and the challenge *RAND*. $SRES_{SIM}$ is sent to the VLR for verification.

6. If $SRES_{HLR} = SRES_{SIM}$, then the SIM is authenticated and allowed access to the network. If $SRES_{HLR} \neq SRES_{SIM}$, then an authentication rejected signal is sent to the SIM and access to the network is denied.

2.1.2 Voice encryption

The frequencies over which voice is transmitted are public, so voice encryption is necessary to avoid interception of the signal over the air. Once the signal reaches the operator's BS, it will be transmitted to the receiver over a wired or wireless mean. In either case, ETSI didn't define any protection: voice transmission in clear over a wired means is publicly accepted, as this is what happens for fixed base telephone conversations, and voice transmission in clear over wireless portions of the network is supposedly not at risk, as it is assumed that the attacker is not aware of the communication path within the operator's network. Voice will only be encrypted from the BS to the receiver if the receiver is herself a mobile user. In the latter case, it should be noted that a different encryption key will be used between the caller and his base station and between the receiver and her base station.

Voice encryption is not mandatory; the choice whether or not to accept an unprotected communication is up to the network. A session encryption key must be computed before a secure communication can take place. The encryption key Kc will change after each user authentication, since the same $RAND$ value is used for encryption key derivation.

The algorithm A8 is implemented in the SIM card and the AuC or HLR. To generate the 64-bit encryption key Kc, the SIM uses the 128-bit random challenge $RAND$ from the network and the secret key Ki. Kc is transmitted to the MS for voice encryption.

The voice encryption algorithm implemented in a MS is A5. It's a stream cipher that takes Kc as input and produces a key stream as output. Ciphertext is obtained by XORing the plaintext and the key stream.

Multiple versions of the A5 algorithm have been defined; the network and MS must support at least one common version to communicate securely. The most widely used A5 algorithm today is A5/3 (A5 version 3); it is based on Kasumi.

When a MS wishes to establish a connection with the network, it indicates which version of the A5 algorithm it supports. If the MS and the network have no versions of the A5 algorithm in common, the network decides whether to accept an unciphered connection or to release the connection. If the MS and the network have at least one version of the A5 algorithm in common, then the network selects the one of its choice.

The voice encryption procedure is outlined in the following steps:

1. The SIM card applies algorithm A8 to the 128-bit input $RAND$ using key Ki to calculate the encryption key Kc.
2. The SIM card transfers the encryption key Kc to the MS.
3. When the MS wants to establish a connection, it informs the network of the A5 algorithms it supports.
4. If the MS and the network have at least one version of the A5 algorithm in common, then the network selects the one of its choice.
5. If the MS and the network have no A5 algorithms in common, the network accepts an unciphered connection or releases the connection.

2.1.3 Other security features

SIM authentication and voice encryption are considered GSM main security features, but a number of minor protections are also available. We have already mentioned anonymity, which allows the concealment of the SIM's permanent identity IMSI, linked to a particular user's identity. The same TMSI should not be used for a long time period to avoid user traceability.

IMEI aims to reduce mobile phone theft. The network can request the IMEI of the mobile station it's communicating with. If the value provided corresponds to the IMEI of a stolen phone, the network may interrupt the communication. Unfortunately, no security feature protects IMEI integrity, so the barring of stolen phones depends on the terminal providing the genuine IMEI to the network.

User-to-SIM authentication may be requested before a user is allowed to employ SIM services. This proof, whose goal is to limit the use of stolen SIM cards, is generally accomplished by PIN verification. PINs are generally 416 digits long, but users can disable this feature. Also, to limit the use of a stolen mobile platform with a different SIM card, mobile phone owners can pair their device with their SIM card. This feature, known as SIM lock, allows a SIM card and a mobile platform to share a secret. The SIM will be denied access to the terminal unless it can prove knowledge of the secret.

2.1.4 Security limitations and attacks on GSM

A number of security limitations have been reproached to GSM. The most obvious is its lack of support for mutual authentication, which enables an adversary to set up a false BS and communicate with any user, since only user authentication is requested and since the network can opt for non-encrypted communications.

As we've mentioned in previous sections, data within the network is not protected. This concerns voice, which is transmitted in clear, as well as signaling information, including cipher keys and authentication tokens. Any adversary that can access an operator's network from the inside will be able to impersonate victims or network elements as well as eavesdrop on communications.

Yet another GSM design limitation is its lack of integrity protection. This is not a major issue on voice communications, where throughput is more important than error detection or protection and where personal voice characteristics allow us to recognize who we are speaking to. Lack of integrity becomes an issue in GPRS, a technology based on GSM, where data transmission is supported.

On top of design limitations, many GSM algorithms have been broken over the years. First, the variable and key lengths are too short to be considered secure, given the increase in computational power since the definition of GSM. The secret key Ki is 128 bits long, which is still acceptable today in symmetric key cryptography, but SRES is only 32 bits long, giving a 2^{16} chance of collision using the birthday paradox, and the encryption key Kc is only 64 bits long. Even worse, in early versions of A5 algorithm, only 54 of the available 64 bits were used for encryption.

Particular implementations of the A3/A8 authentication and cipher keygeneration algorithms, as well as of the A5 algorithm have been breached. The first reference version of the A3/A8 algorithm designed for mobile telecommunication operators, called COMP128 v1, was broken in the early 1990s. COMP128 v1 was kept secret and was not publicly revised. Once it leaked, it was attacked and broken. The first attacks by Berkeley students in 1992 showed that by analyzing COMP128 v1 output on chosen $RAND$ values, the secret key Ki could be retrieved. Once Ki is known, an attacker can clone a SIM card and impersonate a user or make calls at the expense of the victim whose SIM card was cloned. Later COMP128 v1 was also broken by side channel attacks based on power consumption (7).

Early versions of the A5 voice encryption algorithm have also been reverse engineered and broken. The A5 version 1 algorithm was broken in 1994 by an attack that allows finding the voice encryption key by eavesdropping on a two-minute conversation (8). In 1999, an attack on the weaker A5 version 2 was announced (9). Another attack on A5 version 2 based on a ciphertext-only analysis of encrypted off-the-air traffic was published in (10).

Elad Barkhan, Eli Biham, and Nathan Keller (11) have shown a ciphertext-only attack against A5/2 that requires only a few dozen milliseconds of encrypted off-the-air traffic. They also extended their attack against A5/1 and A5/3 on mobile phones that support A5/2 by retrieving the key first used in an A5/2 algorithm and then switching to another A5 version.

GSM networks lack the flexibility to quickly upgrade once security breaches are identified. We described the encryption algorithm A5/3 and the authentication and key generation algorithm MILENAGE, but these have not been widely adopted in GSM.

2.2 3GPP Security

The 3GPP Agreement was signed in 1998 to complete a set of globally applicable technical specifications for a 3G mobile system based on the evolved GSM core networks and the radio access technologies based on UMTS terrestrial radio Access. A separate standardization body, 3GPP2, is developing another third generation mobile cellular system based on CDMA2000 and an evolution of the North American standard ANSI-41.

3GPP security specifications describe both access security and network security. Access security is improved by adding services not provided by GSM and correcting GSM vulnerabilities by employing different algorithms. Network security is an entirely new feature compared to GSM.

3GPP provides over-the-air mutual authentication between the user universal subscriber identity module (USIM) and the network, encryption, and integrity of user and signaling data.

Since 1998, the 3GPP technology has been evolving, and multiple releases of the specification have been published. The first release of 3GPP specifications, release

99 (12), was essentially a consolidation of the underlying GSM specifications and the development of the new UMTS Terrestrial Radio Access Network (UTRAN). Innovative services defined include multimedia messaging service to send text, audio, images and video clips, location services to send user's emergency and commercial data according to their location, mobile station execution environment to allow a mobile station to negotiate its execution environment, and access to the Internet or an ISP. In Release 4 (13), major security enhancements concern the definition of encryption algorithms based on Kasumi and the establishment of mobile application part (MAP) application layer security.

The main improvement in release 5 (14) is the ability to support IP-based communication between network elements. Confidentiality, integrity, authentication, and antireplay protection are obtained thanks to IPSec. Release 6 is now finalized.

2.2.1 3GPP authentication and key agreement

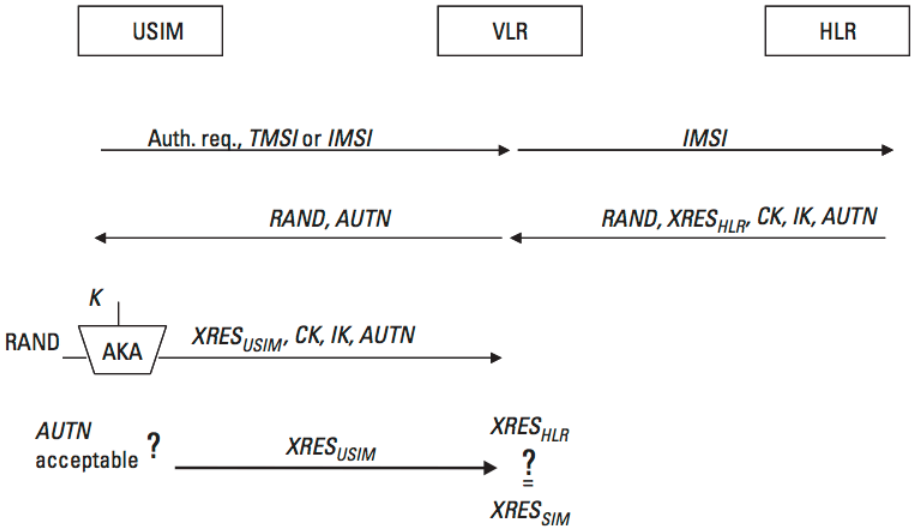

Figure 2.2: 3GPP authentication

3GPP provides mutual authentication and key agreement between the user USIM and the network through the AKA protocol. (See Figure 2.2) AKA is a secret key algorithm; a secret key K must be shared between the USIM and the HLR. It is the HLR that generates authentication values and transfers them to the VLR of the network under which coverage the user is located the moment authentication is performed.

Authentication is requested by the USIM. Once the HLR has transferred authentication values to the VLR, exchanges occur between the USIM and the VLR. Authentication values consist of a quintet (in analogy to GSM triplets) including:

- The challenge $RAND$
- The challenge response $XRES$
- The cipher key CK
- The integrity key IK
- The authentication token $AUTN$

The AKA procedure is outlined in the following steps:

1. Authentication is initiated by the user whenever he wants to make a call from his mobile station or go on standby to receive calls. The user transmits his identity through TMSI and the authentication request
2. The network establishes the identity of the USIM through the 5-digit TMSI. If the TMSI is recognized, the VLR sends a request for authentication to the HLR; if not, it will request the user's IMSI
3. The HLR generates the AKA quintet $Q = (RAND, XRES, CK, IK, AUTN)$ and sends it to the VLR
4. The VLR sends $RAND$ and $AUTN$ to the USIM
5. The USIM verifies if $AUTN$ is acceptable, where $AUTN$ is the network authentication token. If $AUTN$ is valid, the USIM calculates the expected response $XSRES_{USIM}$ using its secret key K and the challenge RAND. $XSRES_{USIM}$ is sent to the VLR for verification. The USIM also calculates the encryption key CK and the integrity key IK
6. If $XRES_{HLR} = XSRES_{USIM}$, then the USIM is authenticated and allowed access to the network. If $XRES_{HLR} \neq XSRES_{USIM}$, then an authentication rejected signal is sent to the USIM and access to the network is denied

The authentication algorithm was not standardized by the 3GPP organization because the architecture demands that every operator manages her users' authentication and sends the authentication quintet Q to the VLR. Nevertheless, the reference algorithm MILENAGE, was designed and is used by most operators.

AKA was designed in such a way as to facilitate roaming and handover between 3GPP and GSM networks because it is expected that for a long transition period, both networks will coexist. To ease roaming between 3GPP and 3GPP2 networks, 3GPP2 has decided to adopt AKA as its authentication and key agreement scheme as well.

2.2.2 3GPP encryption and integrity functions

Signaling system number 7 (SS7), defined by the International Telecommunications Union (ITU) for the PSTN in the 1980s, was the first de facto standard to be used for communication within and between operators' networks. No standard security means are defined in SS7, since wired telephone operators rely on the private nature of the network to infer that attacks will be limited. To adapt SS7 to wireless communications, the MAP protocol was developed and included in release 99. Wireless access to operators' networks nevertheless implies new breaches for attackers, so 3GPP developed security mechanisms specific to MAP (MAPsec) in release 4. MAPsec has been improved throughout 3GPP releases.

Unfortunately, MAPsec provides some degree of protection only on the mobile part of the signaling protocol, not on the entire SS7 protocol. Instead of defining a security protocol for SS7, more and more operators are now switching to IP and IPsec for security. Moreover, MAP can run on top of IP, leaving the choice between IPsec and MAPsec for security.

2.2.2.1 MAPsec

MAPsec is an application layer security protocol, fully useful if applied by all interconnected operators.

Before protection can be applied, SA must be established between the respective MAP network elements. SAs define, among other things, which keys, algorithms, and protection profiles to use to protect MAP signaling. Network operators negotiate among each other and distribute to all network elements the necessary MAPsec-SAs to use between networks.

Each SA contains the sending and receiving public land mobile network (PLMN) identifier, a SPI to identify the SA, an integrity and encryption key and the respective algorithms to use, a protection profile identifier (to identify the security features provided), and an expiration date for the SA.

An interdomain SA and key management agreement should

- Define how to carry out the initial exchange of MAPsec SAs
- Define how to renew the MAPsec SAs
- Define how to withdraw MAPsec SAs
- Decide if fallback to unprotected mode is to be allowed
- Decide on key lengths, algorithms, protection profiles, SA expiration times, and so forth

The security services provided by MAPsec are

- Data integrity
- Data origin authentication
- Antireplay protection
- Confidentiality (optional).

MAPsec provides three different protection modes:

- *Protection mode 0*: no protection
- *Protection mode 1*: integrity, authenticity
- *Protection mode 2*: confidentiality, integrity, and authenticity

MAP messages protected by means of MAPsec consist of a security header and the protected payload. In all three protection modes, the security header is transmitted in cleartext. The protected payload format is described in (15). At present, the only mandatory algorithms standardized in MAPsec are AES in counter mode with 128-bit key length for encryption and AES in a CBC MAC mode with a 128-bit key for integrity.

2.2.2.2 IPsec

The security protocols to be used at the network layer to protect IP signaling traffic are the IETF-defined IPsec protocols, a description of which is provided in (16). In (17), 3GPP defined a minimum set of features required for interworking purposes. IPsec is restricted to ESP and tunnel mode only. Also, key management and distribution between security gateways, defined next, is handled by the protocol IKE. Within their own network, operators are free to use any IPsec feature, including the ones not incorporated in (17).

3GPP defined security domain for network protection (i.e., a network in which the same level of security and usage of security services is provided). Typically a network operated by a single operator will constitute one security domain. Security gateways are entities on the borders of the IP security domains and will be used for securing native IP-based protocols. All IP traffic shall pass through a SEG before entering or leaving the security domain.

The security services provided by IPsec are

- Data integrity
- Data origin authentication
- Antireplay protection
- Confidentiality (optional)
- Limited protection against traffic flow analysis when confidentiality is applied

At present, the only IPsec algorithms mandatory in (17) are 3DES and AES-CBC with 128-bit keys for encryption, and HMAC_MD5 and HMAC_SHA1 for integrity and data origin authentication.

3

A System for Secure GSM Communication

3.1 Introduction

In these years we have been witnessing a significant growth of the telecommunication market. This growth can be temporally divided in two phases. In a first phase the telecommunication networks, such as the GSM mobile phone networks(18) or the Internet, were growing autonomously and independently. A true interoperability between these networks was substantially absent, both because of technical and economical reasons. In a second phase, which is still in progress, several public communication networks are converging in a unified network which is accessible using the most disparate devices. Such a shift has been driven by the recent technological advancements and by the impressive push of the market. We cite, as example, the recent introduction of mobile terminals which can indifferently operate on a GSM network or on a Wi-Fi based network, or the availability of cheap flat rates for surfing the Internet through a mobile phone connection.

Even in this renewed scenario, one of the most popular applications remains the voice-based communication. A voice-based communication service is essentially a service where someone uses some sort of address (e.g., a mobile phone number, a nick name, an IP address) to contact and communicate, via voice, with a remote user by means of a public communication network. There are essentially two types of security issues related to this service. The first type concerns with the problem of *trusting the*

other endpoint of the conversation (e.g., is he really the one he claims to be?). The second type concerns with the problem of *trusting the communication network* (e.g., is someone eavesdropping my conversation?).

Mobile communication networks have historically suffered of several serious security weaknesses that were swept under the carpet by the telecom operators according to the principle of *security by obscurity* (i.e., a system is secure because only authorized people know how it works). Let us consider, as an example, the case of GSM based mobile phone networks. GSM networks are essentially organized in a hierarchical way (see (18)). The GSM enabled mobile equipments (i.e., the mobile phones) are connected to the network through a wireless link with the ground base stations. The base stations are the endpoint of the physical GSM network and are, on their turn, connected to the local public switched telephony network (PSTN or ISDN).

The GSM standard introduces some mechanisms to secure conversations. Three cryptographic algorithms are introduced to this end: A5, A3 and A8(7). A5 is used to encrypt conversations, A3 is used for the authentication between the SIM card and the BS, and A8 is used for performing the key agreement between the SIM card and the BS. Instead, no standard security mechanism is defined for securing the ground based part of the communication.

Unfortunately, the A5 algorithm, and some of its variants which are currently in use by most of the GSM networks, suffers of some serious security weaknesses, as described in the work of Golic(19) and of Barkan et al.(11). These weaknesses allow a determined user to eavesdrop, at a relatively small cost, voice conversations on a public GSM communication channel. It should also be pointed out that the security mechanisms introduced by the GSM standard only apply to the communications occurring between a mobile equipment and the base station it is connected to, while no standard security mechanism for the wired part of the communication.

The problem of avoiding the eavesdropping of a GSM conversation becomes even more dramatic with the progressive switch to the unified network because, in this case, a mobile conversation ending in a network other than the GSM will be subject to the security issues of all the communication networks it will traverse.

Another security problem that emerges from the interconnection of communication networks is the user authentication problem. This is both a problem of trusting the network and trusting the other endpoint of a conversation. In the traditional mobile

phone networks, each user needs to be authenticated, through his SIM card, before accessing the network. By the same token, each mobile phone number can be related to a particular user. This implies that, given a mobile phone number, it is possible to determine the real identity of the user who owns that number. Such a scenario does not apply, e.g., to users connected by the Internet. In this case, the only information we are supposed to know about a user is his IP address, which can be even masqueraded. It is relatively easy for a user connected to the Internet to call, by means of a Voice-over-IP software and a fake geographic number, a user of a mobile phone network while impersonating another user. The arising problem is, in this case, to be able to verify the real identity of the user we are going to talk to.

Finally, the last security issue we point out is the problem of non-repudiation of a conversation. The current communication technologies do not provide users with the ability to prove that the content of a past conversation has not been altered. Consider the case of a commercial transaction conducted during a phone conversation. After the conversation has ended, one of the two endpoints of the conversation could deny its content and refuse to go on with the transaction. Notice that, in these cases, even listening a recording of the conversation could not be enough to establish the truth because one of the two endpoints could claim it has been altered by the other endpoint. This is essentially a problem of trusting the other endpoint of a conversation. The problem here is to not allow a user to deny the content of a conversation he has had in the past.

Starting from these considerations, we believe there is need of a strong formulation of the concept of secure conversation that should guarantee the following security properties:

- **Confidentiality**. The content of the conversation is encrypted. It should be unfeasible for a malicious user eavesdropping on the communication channel to determine the real content of the conversation.
- **Authentication**. Each of the endpoints of the conversation has a proof about the real identity of the other endpoint of the conversation.
- **Non-Repudiation**. It is not possible for the endpoints of a past conversation to deny its contents.

3.2 Existing solutions

There are an increasing number of software and/or hardware solutions that a user can adopt in order to perform "secure" conversation using a mobile phone. We cite, as examples: Criptofonino(20), Cryptophone(21), Sectera(22), VectroTEL X8(23), SnapSoft-ZX2(24) and SecureGSM(25). All these products work by establishing an encrypted communication channel between the two endpoints of a conversation on the top of a standard GSM data call. Encryption is performed by using standard cryptographic algorithms such the AES(26) algorithm. Moreover, special care is taken in using ad-hoc compression technologies which guarantee the possibility to transport the voice stream at an acceptable quality while using a low-bandwidth communication channel such as the one provided by the GSM standard.

In addition to these products, a new generation of voice communication software based on Wi- Fi networks is starting to appear. These software run on handheld devices connected to the Internet through a Wi-Fi connection and use the TCP/IP protocols to perform secure calls with remote peers connected to the Internet using an equivalent software. The most famous example of these applications is Skype(27).

It should be pointed out that all these solutions do only provide support for the confidentiality of mobile conversations. They do all assume that the other endpoint of a secure conversation is trusted. So, there is no explicit support for other security features such as user authentication or non-repudiation of conversations. In addition, the most part of these solutions is available as closed-source commercial products and there is a general lack of information about the way the security feature they claim are implemented.

3.3 SPEECH

SPEECH is a software system for making secure calls by using a Windows Mobile 2003 powered handheld device and a data communication channel. It works by digitizing the input voice, processing it using some cryptographic algorithm and, then, sending the out-coming data stream to the handheld device of the other endpoint of the conversation where it will be played. The confidentiality of conversation is obtained by encrypting the input voice stream using the AES256 symmetric cipher. The encryption of the communication is done according to the end- to-end paradigm: this means

that the voice stream remains encrypted along all the communication channel joining the two endpoints of an ongoing conversation. Moreover, SPEECH sup- ports the authentication of the peers of a conversation either by using X.509 digital certificates or passphrases. Finally, non-repudiation of conversations is implemented through a digital sig- nature scheme.

One of the noteworthy features of SPEECH is the ability to operate on different types of communication networks (currently, GSM networks and generic IP-based networks). For this reason, SPEECH adapts the quality of the digitalized voice stream to the bandwidth of the under- lying communication channel. The management of the voice stream is done using Speex(28), an open source audio codec which, together with a light communication protocol, allows to talk in full-duplex mode with good audio quality and short delay over a standard GSM data connection.

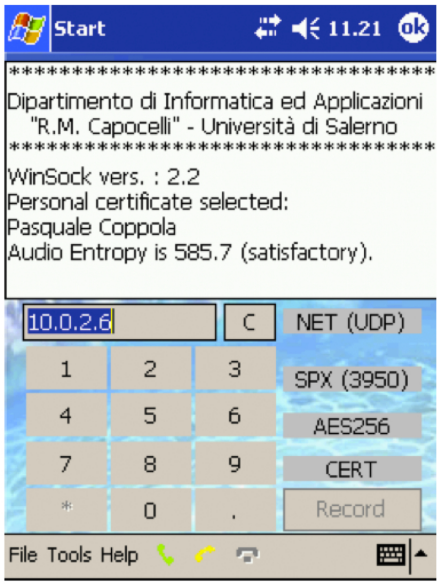

Figure 3.1: SPEECH main window

Figure 3.1 shows the main SPEECH window. Figure 3.2 shows the options window where the user can choose among security and communication channel parameters.

The SPEECH architecture, shown in Figure 3.3, is organized as a stack of five independent modules. During a conversation, the input voice is sampled by the Audio module in a voice data stream and is sent down through the other modules of the architecture. These modules encode the voice data stream, compress it, encrypt it

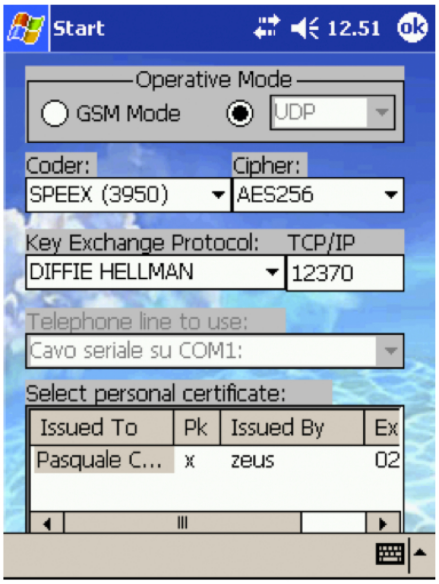

Figure 3.2: SPEECH options window

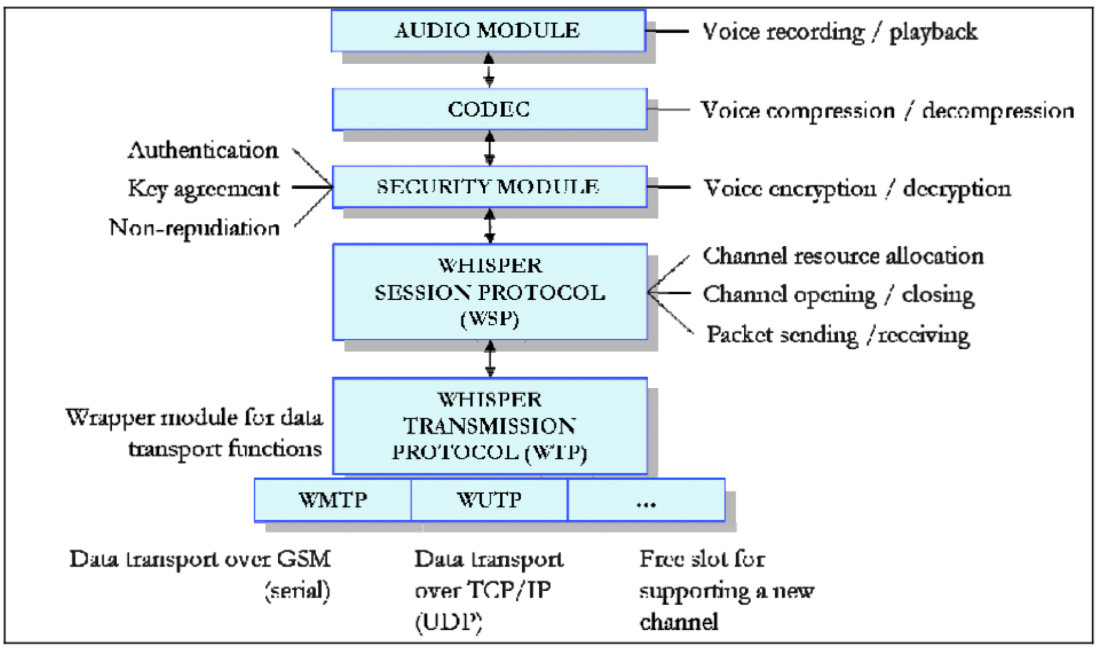

Figure 3.3: SPEECH architecture

and, then, send it to the other endpoint of the conversation. The communication between modules is defined through a standard interface. The main advantage of this approach is that it keeps minimal the amount of changes needed to introduce new features in the system, such as the support for a new type of communication channel or the introduction of a new security feature.

All the modules of the SPEECH architecture work in a multi-threaded architecture so to be able to process, at the same time, the voice stream outgoing to the other endpoint of the conversation and the voice stream incoming from him.

In the following sections we examine in details the inner working of each module.

3.3.1 Audio module

The audio module is in charge of performing two tasks. First, it uses the microphone device of the handheld it is running on for encoding the input audio stream. Audio is sampled at a frequency of 8000Hz where each sample is represented using 16 bits. Second, it plays the output audio stream received from the other endpoint of the conversation using the system audio device.

3.3.2 Voice codec

This module compresses the input audio stream coming from the audio module and uncompresses the output bitstream coming from the other endpoint of the conversation through the lower levels of the stack. The compression and uncompression is done using the Speex codec, an open-source software which uses an efficient audio compression format optimized for processing voice conversations. It is based on the CELP(29) codec and is designed to operate using bit rates ranging from 2 kbps to 44 kbps. The compression factor is adapted to the bandwidth of the underlying communication channel. An important advantage of the Speex co- dec is that it is, at the best of our knowledge, the only open source codec that has been explicitly optimized to run on devices not having a floating point processing unit (this is typically the case of handheld devices).

3.3.3 Security module

The security module implements all the security features of SPEECH, such as peers authentication, key agreement, stream encryption or the digital signature of an ongoing conversation. It is automatically recalled whenever a new conversation is initiated as well as to process incoming and outgoing voice data stream. A thorough discussion of the security features available in the current implementation of this module is available in Section 3.4.

3.3.4 WSP module

As previously said, one of the requirements of SPEECH was the ability to operate on different types of communication channels such as GSM or Wi-Fi networks. This is an engaging task because each type of communication channel is characterized by its own performance (e.g., the round trip time of data packets, the bandwidth) and quality of service (the error rate, the sup- port for retransmission of lost data packets). This implies the need for a communication protocol which is light, efficient and, wherever needed, reliable and which abstracts the details of the communication channel in use.

The WSP module implements the *Whisper Session Protocol*, a light session-based communication protocol which can operate both on reliable and unreliable mode. It is an evolution of the Nautilus Session Protocol(30) which is part of Nautilus, a generic software for perform- ing secure half duplex voice conversations. The main difference with respect to the original protocol is that the WSPl supports full duplex data transmission. This required the introduction of several synchronization primitives and a multi-threaded architecture needed for handling, at the same time, the outgoing and the incoming voice streams.

Data packets exchanged by WSP are formed by a fixed length header and a variable length block of user data. The header is made of 2 bits used to specify the type of packet and 14 bits used to maintain a progressive sequence number used to detect gaps in the communication. The current version of WSP supports the following types of packets:

- **Reliable**. The receiver of this packet has to acknowledge to the sender its reception. Whenever a new reliable packet has to be sent, a new unique sequence number is generated and used to mark it. After the transmission, the sender

waits from the receiver a packet of type Acknowledge carrying the same sequence number. If the sent packet is not acknowledged within a fixed interval of time, the packet is assumed to be lost and it is retransmitted.

- **Acknowledge**. This kind of packet is sent as an acknowledgement for a reliable packet previously received. The sequence number field is filled with the number of the packet to be acknowledged.
- **Unreliable**. The packet is sent without any acknowledgement.

3.3.5 WTP module

The WTP module implements the *Whisper Transmission Protocol*, a low-level data transmission protocol which has been designed to abstract the primitives (e.g., establish a new session, send or receive data) needed to perform voice calls over different types of data communication channels. Currently, SPEECH offers two implementation of the WTP protocol which use, in turn, a GSM based and UDP/IP based communication channel.

In the GSM based WTP implementation, the user wishing to initiate a new conversation has to supply the mobile phone number of the other user he is interested to talk to. The WTP module will try to establish a serial communication channel with the destination of the call through a GSM data calls.

In the UDP/IP WTP implementation, the user wishing to initiate a new conversation has to supply the IP address of the other user he is interested to talk to. The WTP module will try to establish a connection with the remote user by means of UDP datagrams.

In both cases, once the connection has been established, the initialization of the conversation will be managed by the upper level of the SPEECH architecture.

3.4 The SPEECH security

In this section we introduce the security features available in SPEECH and the way they are implemented by the security module (see Section 3.3.3).

3.4.1 User authentication and key agreement

Whenever two users try to initiate a new secure conversation, the two SPEECH installations running on their handheld engage in a key agreement protocol. The purpose of this protocol is to agree on a common session key to be used for performing cryptographic operations on the voice data stream and, optionally, to verify the identity of the parties of the conversation. SPEECH supports three different forms of user authentication and key agreement schemes, each with a different level of security. Those schemas produce a shared random secret which is hashed in two encryption keys (one for encryption and one for decryption) of the correct size for the symmetric cipher currently in use.

- **Basic key agreement**. Whenever two users initiate a new conversation, the SPEECH installation running on their handhelds use a vanilla implementation of the 4096bit Diffie-Hellman key-exchange protocol(31) to agree on a common secret key. This form of agreement does not guarantee to the user the identity of the other end- point of the conversation but it is enough when we are just interested in guaranteeing the confidentiality of a conversation.

- **Passphrase based key agreement**. Two users interested in having a secure conversation choose a common passphrase. Whenever a new secure conversation has to be initiated by these users, they will generate each a new session key using the shared passphrase. The reuse of the same passphrase is always possible, because the generated session key will be never the same as the key-exchange algorithm which generates the common key is based on the exchange of encrypted random values. This approach provides with a basic form of authentication since it is expected that the passphrases are known only by their legitimate owners.

- **Certificate based key agreement**. Two users initiating a new secure conversation own a legitimate X.509 digital certificate which has been previously loaded in their de- vice. Moreover, the two devices are supposed to be loaded with the certificate of the root certification authority of the user to be called. If these conditions are met, the two parties use the standard TLS 1.0 protocol(32) to perform the mutual authentication and keys agreement. The call originator plays the client role of the TLS protocol while the receiver of the call play the server

role. According to the TLS specification, each client submits its X. 509 certificate and provides its verification.

Notice that this is the only user authentication scheme that makes it possible to activate the non-repudiation feature.

Those three protocols run on the top of WSP and use only packets marked as reliable.

3.4.2 Key escrowing

Providing end users with the ability of performing strongly encrypted phone calls arises once again the problem about balancing the preservation of individuals privacy and national security interests. For this reason, the system to design should support an *ethical key-escrow* allowing the decryption of a conversation if and only if a selected set of disjoint agencies have authorized it.

All the three key agreement protocols supported by SPEECH produce session keys in deterministic way starting from a larger non-deterministic common secret shared by the endpoints of a new conversation. The key escrowing algorithm used by SPEECH works as follows. Suppose there are n security agencies we want to involve in the key escrowing process and that each of these agencies has revealed its public key. At the beginning of a new conversation, the secret shared by the two users is split in n disjoint segments, where each segment is encrypted using the public key of the corresponding agency. Then, each segment is sent to the corresponding agency. If, in a later moment, there is need to escrow the key used to secure a past conversation, the agencies will decrypt their fragments and assemble the original secret.

The problem that arises in this protocol concerns with the transmission of the fragments of the secret to all the agencies that participate to this process. As a matter of fact, it is not possible to always assume the possibility for a handheld device running SPEECH to establish several connections toward different destinations at the same time (e.g., a GSM based smartphone would only be able to make a GSM data call at time). For this reason, we decided to not de- liver the encrypted secret fragments directly to each agency, but simply to put them on the communication channel before starting the conversation (i.e. sending them to the peers). In this way, all the agencies

eavesdropping on the communication channel will get a copy of the encrypted secret fragments together with the whole conversation.

The key escrow process we developed is considered ethical because no agency can access private data without the agreement of all the others. Moreover, it is possible to extend this mechanism by using a different partitioning scheme, such as the one documented in (33), so to allow a subset of all the agencies to rebuild the session key.

3.5 Confidentiality

All the conversations made with SPEECH are encrypted using the AES256 symmetric cipher with 128 bit block size and 256 bit key generated according to the common secret agreed using one of the approaches described Section 3.4.1. The cipher mode we use is *OutputFeedBack* (OFB). This mode works by repeatedly encrypting an initial vector and processing with the xor operator the resulting 128 bit together with the original 128 bit data block. This is a suitable operative mode for our communication channel. The choice of the OFB mode has been done by considering that when using a wireless communication channel like the one provided by GSM networks or by Wi-Fi networks the number of data losses or communication errors is relatively high. By using the OFB mode, we are guaranteed that an error occurring in a data block will not propagate over the entire blocks chain.

3.6 Non-repudiation

Non-repudiation of a conversation is implemented in SPEECH by having each party of the conversation sign, with his private key, the hash of a recording of the whole conversation. The problem that arises is to make it possible for both parties to have, at the end of the communication, the same identical copy of the conversation. Such a task is not easy as it seems because both the underlying communication channel and the transport protocol may be unreliable and, so, voice packets can be damaged or be lost. For this reason, SPEECH adds to each packet of the encrypted voice stream an integrity checksum and a sequence number. The first is used to verify if a received package has been damaged by the transmission, the second is used to detect lost packets.

At the end of conversation, the total number of damaged and lost packets is evaluated. If a certain threshold is not exceeded, then the party will request to the other party a copy of all the packets to be recovered using a reliable protocol. Otherwise, if the threshold is exceeded, the non-repudiation service becomes unavailable. The definition of the threshold is crucial because accepting a large number of contiguous errors may alter the real meaning of the conversation; in other words peers could not be aware of what they are signing[1]. Another reason for bound- ing the number of packets to recovery is that the downloading of a large number of recovered packets may take a long time to finish. Finally, if the rebuilding process was successful, the parties use a reliable protocol to exchange the checksum of the whole conversation. Each user can later listen the conversation and decide to send or not the signature. This feature has been implemented by means of standard public key cryptography, such as RSA with SHA-1(34).

[1]Adopting the same criteria introduced by current laws about digital signature, assuming that signer MUST visualize the document before signing it.

4

A System for Secure and Efficient SMS Communication

4.1 Introduction

SMS messages are currently one of the most widespread forms of communication. As reported by (35), in 2009 worldwide SMS traffic topped five trillion messages, and that figure is set to exceed 10 trillion in 2013. Sending an SMS is cheap, fast and simple. The success of SMS messages has motivated many researchers to explore fields of application as an extension of their original purpose. We have seen many unusual or strange applications, such as devices which allow the switching on and off of house heating systems using an SMS (36). Alternatively, through SMS, whenever the temperature of a refrigerator exceeds a certain threshold, it is possible to automatically communicate the problem (37). Indeed, through SMS, fridges can even signal when they are running out of beer (38).

Along this path, we also have been experiencing a multitude of services which allow users to order financial transactions (often, microtransactions) by sending SMS messages. This is the case, for example, of the service provided by the Province of Rimini Mobility Agency, in Italy, which allows registered users to buy electronic tickets using a simple SMS which contain a standard fixed string of text (39). Users are able to buy multiple tickets at the same time, where each ticket is encoded as a standard SMS.

Many of these services seem to ignore one important drawback of SMS based communication: the substantial lack of security. For example, by using bulk SMS service providers it is relatively simple to forge an SMS and send it to a recipient, as if it was transmitted by any sender. So, services like the ones we mentioned before are prone to be attacked by malicious users (40). In this case, for example, it would be easy to damage a user of the service by just sending to the Mobility Agency servers several forged SMS that have apparently been originated by that user, thus forcing the user to buy a multitude of tickets.

Two are the major security vulnerabilities affecting SMS based communication: the lack of confidentiality during the transmission of a message and the absence of a standard way to certify the identity of the user who sent the message. These vulnerabilities originate from the protocol used to exchange SMS messages and from the infrastructures used to implement it. There are currently several proposals, mostly coming from the scientific research, about how to secure SMS messages. Some of these proposals require security to be injected at the protocol level. Instead, most of them consist of software frameworks which can be installed on mobile phones and/or on the SIM cards in order to implement security features.

This chapter presents a novel contribution to this field, consisting of a software framework which allows two peers (end users and/or software applications) to exchange SMS messages in a secure way. This proposal differs from the ones already existing in literature, because it allows users to choose which cryptosystems and security parameters to use when sending a secure message, in order to achieve the optimal trade-off between the requirements of the user, the cost and the efficiency of the operation. For this reason, this chapter presents also an experimental analysis aimed at evaluating the efficiency of the cryptosystems implemented in the framework under several usage conditions, so to give users a better insight regarding the cryptosystems to use according to their requirements. This analysis has also been useful to expose some serious performance issues which seem to exist in one of the cryptographic libraries commonly used to implement security features on mobile devices and which contradicts the theoretical expectations about the performance of the ECDSA algorithm. In the last part of the chapter, these issues have been investigated and an optimized version of this library has been developed. The cryptosystems using this new library exhibit a significant

performance boost, a lower memory footprint as well as a behavior that is consistent with theoretical expectations.

4.2 Related works

There have been several proposals up to now to secure SMS based communications on a GSM network. A first category of contributions tries to address these problems by changing the original GSM specifications in order to introduce security features. This is the case, for example, of the proposal presented by (41) which argues for a modification of the GSM protocol at the transport level to achieve confidentiality between mobile equipment (ME) and the GSM base station (BS) connected to it. The advantage of this approach, if followed, is that it would be able to inject security features at infrastructural level, thus allowing to strengthen the entire communication network. However, it is unlikely that these proposals will be implemented and widely adopted in the near future, mostly because of the technical difficulties arising from the implementation of structural changes in a well established network architecture like the GSM one.

A second category of contributions to secure SMS communication — which is becoming viable because of the increasing diffusion of ME with advanced computational capabilities — introduces security features through the implementation of security schemes at the application level. The resulting software frameworks can be categorized according to the place where the application implementing the security scheme, and their cryptographic keys, are stored. The first possibility is to locate the application and its keys in a programmable SIM card used by the ME. This solution is adopted by systems like the one developed by (42) or by (43).

The use of a programmable SIM card has several advantages, such as the tamper resistance of the card and the possibility to move it from one ME to another without any data loss. However, it also has a relevant drawback: the limited computational capabilities of a programmable card do not allow the execution of complex security schemes within a reasonable amount of time.

An alternative approach, adopted in systems like the one presented by (44), is to use a SIM card only to store the cryptographic keys used in a scheme, while using the computational capabilities of the ME to run the scheme. In addition, it is also possible to use a SIM card to perform certain cryptographic operations, while executing the

remaining part of the application through the ME, like in the mobile payment scheme presented by (45).

Even just storing the cryptographic keys on a SIM card has an important disadvantage: the user is tied to the SIM provided by a particular operator and the interoperations with SIM cards relative to other operators may be difficult or impossible to achieve.

These disadvantages can be largely overcome by working at the application level, without using the SIM card at all. This approach is used by systems like SafeSMS (46), MIABO (47) and the systems described by (48) and by (49), which adopt a simple key management scheme based on pre-shared passphrase and/or public-key cryptography. As those systems belong to the Person-to-Person (P2P) model only, they are suitable for a restricted set of users and the security level ensured is strictly related to the key-distribution scheme. In literature, there are other systems targeted for Client-Server (CS) scenarios (e.g., m-commerce, homeland security) in which the involved entities are not just humans, but also authorities which supply services (e.g., banks, CA, security agencies) through applications running on servers. Some available solutions to such problems are: Trusted-SMS (50) and SMSSec (51). The Trusted-SMS framework (TSMS) allows users to exchange non-repudiable SMS by means of the ECDSA algorithm. Three different entities take part into TSMS: a service supplier which is responsible for providing services, a CA which manages keys, and the ME which uses the supplied services by means of a J2ME midlet. TSMS presents two provisional scenarios and two transactions scenarios which differ in the role that CA has in key management. SMSSec is an end-to-end protocol with the object of providing SMS security. It does not require any private-key to be stored in the mobile device, but provides user authentication and encryption by means of a PIN code running an ad hoc protocol with an Authentication Source (AS) authority. SMSSec uses symmetric (AES) and asymmetric (RSA) cryptography for the encryption and key-agreement respectively. On one hand, this approach allows a fast encryption process, while not altering the size of the SMS. Whereas, on the other, prior to any transactions, there is need of a new key-agreement with the AS, with the consequent exchange of additional initialization messages.

4.3 Our proposal

Sending or receiving SMS messages is a common habit for most of the people using a cellular phone. The general expectations are that mobile equipment should be able to promptly send and receive a message with almost no delay. The way of sending and receiving secure SMS messages could change this habit since it involves processing incoming/outcoming secure messages which may be of several seconds. Thus, the efficiency of these systems is almost as important as the security they guarantee. Moreover, the efficiency of a system for guaranteeing secure SMS messages is heavily influenced by the same ingredients which govern its own security: in other words, the cryptosystems and the security parameters it uses. The user should be given the possibility to choose to trade part of the security of a system with shorter response times, and vice-versa. Moreover, such a customization should be allowed on a per-message basis, because the same user might need to send messages, even to the same recipient, with different levels of security. As a matter of fact, all the systems for sending secure SMS messages presented so far in literature are bound to a particular cryptosystem. While this choice simplifies their development, it may have a negative effect on their ability to meet the requirements of the users, as we said above.

By keeping this in mind, we designed SEESMS, a Java based framework for exchanging secure SMS that aims to be efficient by supporting several cryptosystems through a modular architecture. This choice offers the advantage to easily experiment and assess the performance of several cryptosystems using several security parameters. The same advantage holds also for the final users, as they may choose which combination of cryptosystem/security parameters better suit their needs. SEESMS works at the application level (see Section 4.2) and can be used for exchanging secure SMS in the P2P and in the CS scenarios. It can be seen as a tool that uses an SMS based communication channel as bearer service to exchange encrypted, non-repudiable and tamper-proof messages. The current version of SEESMS supports some of the most used digital signature schemes (i.e. RSA, DSA, ECDSA (52)) and public-key based cryptosystems (i.e. RSA, ECIES (53)). Comparing to other key-management mechanisms (such as PGP), SEESMS uses a centralized and lightweighted implementation in which only a central authority can distribute signed public-keys.

The Java language has been chosen for the implementation of the framework because it is widely adopted in mostly all the mobile phones, ranging from the old-fashioned models to the latest ones. The cryptographic library used for implementing SEESMS is (54) (BC). To our knowledge, this is the only existing Java library available for mobile devices that supports RSA, DSA, ECIES and ECDSA algorithms together.

4.4 The architecture

The SEESMS framework adopts a hybrid architecture. If a user is interested in sending/receiving a secure message through SEESMS and has never used it before, then he has to contact a trusted third-party server, called Secure SMS Management Center (SSMC), to request a customized copy of the SEESMS client application. Similarly, if the user has already installed the SEESMS client, but does not own the public-key of the recipient of the message (or the public-key of the user who sent him a secure SMS message), he has to contact the SSMC server to ask for a copy of his key (this behavior is similar to the PGP key-servers). Instead, if the user already owns the public-key of his recipient, he will establish a direct communication in a peer-to-peer fashion, without further interaction with the SSMC server. Due to the use of a standard interface definition, all the cryptosystem engines have the same interface resulting in the ability to load them in the framework seamlessly.

The following sections describe in details the SEESMS software components and their internal architecture, as described in Figure 4.1.

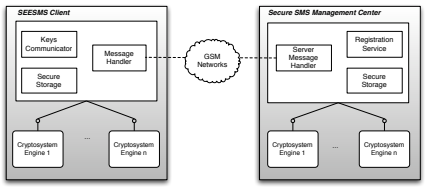

Figure 4.1: The architecture of SEESMS

4.4.1 Secure SMS management center

The SSMC is in charge of handling the provisioning process, used to deliver to new users a customized copy of the SEESMS client application, and the key-distribution process,

used to send the public-keys of registered users following a client request. The entire communication with clients is done by using signed SMS messages. The application includes the following modules:

- **Registration Service (RS).** The RS is used to register new users, to provide them a copy of the SEESMS client application and to run key-exchange protocols with them. More details about these functions are provided in Sections 4.5.1 and 4.5.2.
- **Server Message Handler (SMH).** The SMH is a module that can be used to exchange messages with another peer by means of SMS messages. It also includes the code needed to serialize/deserialize SMS messages and send/receive them through a GSM modem.
- **Secure Storage (SS).** The SS implements a secure local storage area used to encrypt and to maintain sensitive data about the users that are registered to the service, such as their public-keys or their registration information. Data is encrypted using the AES symmetric cipher and stored in a relational database.
- **Cryptosystem Engines (CE).** The CE are the modules that take care of securing the messages exchanged with a remote user. Each CE carries the implementation of a cryptosystem and offers up to three standard set of functions: *Key Generation*, *Message Encryption/Decryption* and *Message Signature/Verification*. These engines are used by the SSMC to implement the user registration phase and the key-exchange protocol. The current version of SEESMS includes the engines implementing ECC (ECDSA and ECIES), RSA and DSA.

4.4.2 SEESMS client

The SEESMS client application can be used by two parties to exchange encrypted and digitally signed SMS messages. It includes the following modules:

- **Message Handler (MH).** The MH is responsible for sending and receiving secure SMS messages. It is a trimmed version of the SMH, not including the code needed to handle communication over a GSM modem.

- **Secure Storage (SS).** The SS implements a secure local storage area used to hold sensitive data such as the cryptographic keys of a user.
- **Cryptosystem Engines (CE).** Similarly to the SSMC case, the CE are used to implement the registration phase and the key-exchange protocol and, moreover, all the functions related to secure communications with another user.
- **Keys Communicator (KC).** The KC module implements the client-side key-exchange protocol described in Section 4.5.2 which is used to communicate to the SSMC the cryptographic keys generated by the client.

4.5 SEESMS in action

This section describes all the phases concerning the transmission of a secure message with SEESMS, assuming that the sender of the message has been using the framework for the first time.

4.5.1 Provisioning of the client application

A user interested in exchanging secure SMS messages using SEESMS firstly has to download and install a copy of the application using the RS offered by the SSMC.

The provisioning process takes several steps:

1. The user registers on a web site hosted by the SSMC and asks for a copy of the client application.
2. The RS generates a random string, called *nonce*, used in a subsequent phase of the registration process (step 4) to ensure that the user running the client application is the same that has requested for it. The nonce is split in two parts: the first part is communicated to the user through the registration web page and email, the second part is hard-coded in the copy of the client application that will be delivered to that user.
3. The RS creates a software package containing the second part of the nonce generated in the previous step and the public-key of the server, and publishes it on a web site hosted by the SSMC. A WAP-Push message indicating the URL where

the package (just created) has been published, is sent to the phone number specified by the user during the registration phase. The URL is randomly generated in order to avoid the possibility for an attacker to download a copy of the application instead of the legitimate user.

4. The user downloads and installs the client on his mobile equipment and starts the application. During the first execution, the user inputs the part of the nonce received in step 2 of the registration process. The nonce is thus reassembled by combining this part with the other part embedded in the downloaded package in order to be used later (see the Section 4.5.2).

At the end of the initialization, the user inputs a passphrase that is used to generate a pair of cryptographic keys which are subsequently stored in the SS of the client application.

4.5.2 Key-exchange protocol

The public-keys generated by the user at the end of the provisioning phase must be sent to the SSMC server. The transmission must be performed in such a way to guarantee that the communication originates from the same user who downloaded the application in the previous phase.

For the sake of simplicity, let us consider the case when the user has only one pair of cryptographic keys: the private-key ($\mathcal{SK}_u$) and the public-key ($\mathcal{PK}_u$). Moreover, let $\mathcal{N}_u$ be the part of the nonce generated and sent to the user during step 2 of the provisioning phase (see the Section 4.5.1). The key-exchange protocol between a new user and the SSMC server requires the following steps.

The client:

1. composes a message $\mathcal{T}$ containing his phone number and a timestamp;
2. computes $\mathcal{H}$ as the keyed-hash (HMAC) of $\mathcal{T}||\mathcal{PK}_u$ using the key $\mathcal{N}_u$;
3. composes a message $\mathcal{T}'$ containing $\mathcal{T}$ and $\mathcal{H}$;
4. computes $\mathcal{D}$ as the result of the decryption of the message $\mathcal{T}'$ using the private-key $\mathcal{SK}_u$;

5. sends to the server an SMS message $\mathcal{M}$ containing $\mathcal{D}$ and the public-key $\mathcal{PK}_u$.

After receiving the message $\mathcal{M}$, the server:

1. reads from $\mathcal{M}$ the public-key $\mathcal{PK}_u$ and the value $\mathcal{D}$;
2. computes $\mathcal{T}'$ as the result of the encryption of $\mathcal{D}$ with the public-key $\mathcal{PK}_u$ contained in $\mathcal{M}$;
3. extract from $\mathcal{T}'$ the fields $\mathcal{T}$ and $\mathcal{H}$;
4. computes $\mathcal{H}'$ as the keyed-hash (HMAC) of $\mathcal{T}||\mathcal{PK}_u$ using, as a key, the nonce $\mathcal{N}_u$ generated for that user;
5. extracts from $\mathcal{T}$ the phone number of the user, the timestamp and keyed-hash $\mathcal{H}$;
6. checks the timestamp to avoid reply attack. If the check is successful, compares the extracted tag $\mathcal{H}$ with $\mathcal{H}'$ computed at step 4. If the tags are identical, the public-key of the user is saved in the SS, a signed confirmation SMS is sent to the client and the registration process ends correctly. Otherwise, the registration process ends with a failure.

From this moment on, all the SMS messages involved in the communications between the client and the SSMC will be signed. Moreover, due to the HMAC verification, Man-In-The-Middle attacks will be easily detected and reported.

4.5.3 Exchange of a secure message

SEESMS implements secure SMS messages exchange by using binary SMS messages rather than traditional textual messages. Each binary SMS message can hold 140 bytes (equivalent to the 160 7-bit characters used for textual messages). The 140 bytes are partitioned as shown in Figure 4.2.

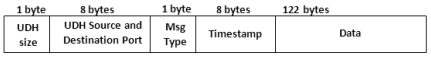

Figure 4.2: SMS Payload organization

The first two fields (i.e., *UDH size* and *UDH Source and Destination Port*) represent the SMS User Data Header (UDH), a standard extension to the GSM specifications

which allows to deliver the message to an application listening on a specific port at the receiver ME. The subsequent 9 bytes are used to specify the message type (1 byte) and the timestamp (8 bytes). The *Msg Type* field indicates the cipher used to process the current message and the key length used by the cipher (e.g., 1024 bits RSA public-key or 160 bits ECIES encrypted text). The *Timestamp* field marks the time when the message has been sent. Finally, the *Data* field is used by the chosen cryptosystem to carry the content of the message together with cryptographic tokens such as signatures, public-keys and so on.

Suppose a registered user is interested in sending a secure SMS message to another user through SEESMS. If he already owns the public-key of the recipient, then he has just to input which cryptosystem to use, the corresponding security parameters and the text of the message to be sent. The input message is processed by the chosen cryptosystem and one or more SMS messages are generated and sent to the recipient. If the public-key of the recipient is not known, then the user application sends to the SSMC server the phone number of the recipient, asking for his public-key. The SSMC responds with an SMS signed with his own private-key, containing a copy of the public-key of the recipient (if it exists).

4.6 Experimental setup

Several tests have been conducted in order to evaluate the efficiency of the cryptographic algorithms available with SEESMS and to determine which security configuration would better suit the needs of a user. The framework is designed to handle any kind of cryptographic operations. Nevertheless, in the tests have been evaluated only the signature operations because otherwise it would have implied a longer exposition. Moreover, this choice is justified by the observation that signing operations have a computational complexity similar to the encryption ones.

This section briefly discusses the cryptosystems involved in our experiments and describes the security equivalence with respect to their key sizes.

4.6.1 Input cryptosystems

The cryptosystems included in our experimentation are RSA, DSA and ECDSA. The RSA cryptosystem is the most widely used public-key based cryptosystem. It may

be used to provide both secrecy and digital signatures and its security is based on the intractability of the Integer Factorization Problem (IFP). The Digital Signature Algorithm (DSA) is the first digital signature scheme to be recognized by a government. Its security relies on the Discrete Logarithm Problem (DLP) that is shown to be as hard as the IFP. The Elliptic Curve Digital Signature Algorithm (ECDSA) has been proposed as an ANSI X9.62 standard. Unlike the Discrete Logarithm Problem (DLP) and the Integer Factorization Problem (IFP), the Elliptic Curve Discrete Logarithm Problem (ECDLP) has no subexponential-time algorithm. For this reason, the "strength-per-key-bit" is substantially greater in an algorithm that uses elliptic curves.

Since the ECDLP appears to be harder than the DLP (or the problem of factoring a composite integer n), cryptography using elliptic curves offers the same security level of RSA and DSA with smaller keys. A detailed presentation of the security-equivalent configurations is described in (55) and summarized in Table 4.1.

Table 4.1: Rough Comparison of RSA and ECDSA Key Size Security Levels (in bits)

ECDSA	RSA
112	512
128	704
160	1.024
192	1.536
224	2.048
256	3.072

4.7 Experimental results

The tests compare the performance of RSA, DSA and ECDSA when signing random messages of fixed length using an increasing level of security. A message is signed by hashing it using the SHA1 algorithm and, then, signing the resulting text using one of the supported algorithm. All the cryptographic algorithms have been evaluated using different key sizes ranging from 512 to 3.072 bits, for RSA and DSA algorithms, and from 112 to 256 bits, for ECDSA. This work analyzes the performance of the three algorithms by comparing them using the security-equivalent configurations described in Table 4.1. For the sake of brevity, in the results have been reported only RSA and

DSA security configurations, thus referring the interested reader to the above mentioned table for the equivalent ECDSA security configuration.

All the collected data and results have been averaged on 200 different runs. The measurements have been conducted on two widely available mobile devices: the Nokia N95-8GB (Symbian OS 9.2 - CPU 332 MHz) and the HTC S620 (Windows Mobile 5.0 - CPU 201 MHz).

The expectations about these tests were that ECDSA would perform much better than RSA and DSA when producing a digital signature. Furthermore, we expected that RSA and DSA to process a digital signature verification much faster than ECDSA.

4.7.1 Time efficiency

We evaluated the time efficiency by measuring separately the time elapsed to sign and to verify a single generic message. These two measurements report the time that the user has to wait every time he sends and receives a secure SMS message on the ME, provided that it has already been configured. The execution times have been evaluated by using the `System.currentTimeMillis()` primitive available within the J2ME framework.

Figure 4.3 and 4.4 show the time needed to digitally sign an SMS using, respectively, an HTC S620 and a Nokia N95-8GB. Despite the expectations, the RSA algorithm performs generally better than ECDSA. The DSA algorithm is slightly faster than RSA for small key sizes, however it is only available with keys no longer than 1.024 bits. The only case where ECDSA outperforms RSA is when using very long keys (near 3.072 bits). This behavior is worth of further investigation because it is widely known from literature that ECDSA should perform generally faster than RSA and DSA.

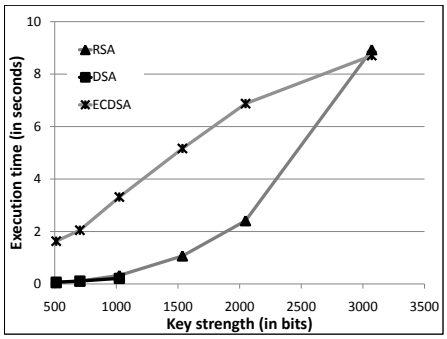

Figure 4.3: HTC S620 average signature generation time

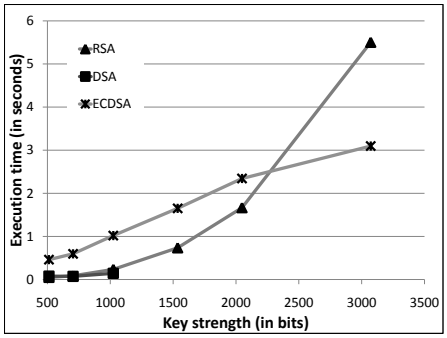

Figure 4.4: Nokia N95-8GB average signature generation time

Concerning the signature verification process, the RSA algorithm performs much better than ECDSA and slightly better than DSA (see Figures 4.5 and 4.6). This is partially what it is expected because, for public-key operations, RSA can benefit of the public exponent size which, according to the algorithm, is often a prime close to a power of 2 (e.g., 3, 5, 7, 17, 257, 65.537). However, we were surprised to notice such a big difference between the performance of RSA and ECDSA. For example, when using key sizes of 1.024 bits, RSA ($\sim$ 15 ms) was approximately 300 times faster than ECDSA ($\sim$ 4500 ms) on a HTC S620 phone.

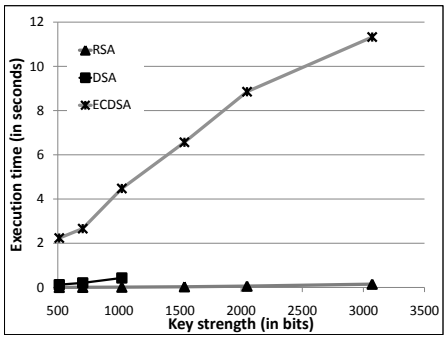

Figure 4.5: HTC S620 average signature verification time

These results seem to indicate that ECDSA performs, in practice, worst than expected. In order to further investigate this behavior, the memory usage was profiled when signing a message by mean of the Sun Java Wireless Toolkit Memory Profiler (WTK). The results, presented in Figures 4.7, 4.8 and 4.9, show that not only ECDSA has much higher memory requirements than RSA and DSA, but also that during the lifespan of a signature generation, a significant amount of memory is apparently and

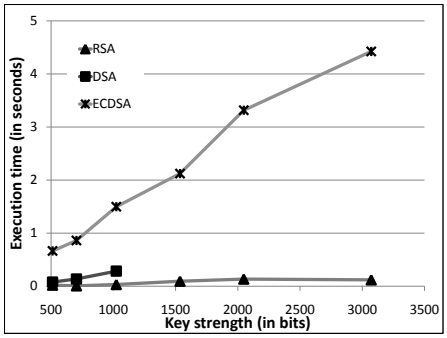

Figure 4.6: N95-GB average signature verification time

repeatedly allocated and deallocated. This behavior is likely to be due to the activity of the Garbage Collector module used by the Java virtual machine which runs the application. This module is automatically activated by the system whenever the memory usage of an application reaches a certain upper threshold, and its reaction is to reclaim (and to free) all the memory that is not in use anymore. The overhead due to memory allocations and deallocations is likely to be responsible for the bad performance of ECDSA. As shown below, even the bigger power consumption with the respect to the other two cryptosystems is likely to depend from this reason. The other two cryptosystems, instead, show simpler memory profiles. In their case, since the maximum amount of memory threshold is never reached, the use of the Java Garbage Collector module is reduced. In Figures 4.7, 4.8 and 4.9 the horizontal scale axis is not relevant because it depends on the average user-input time.

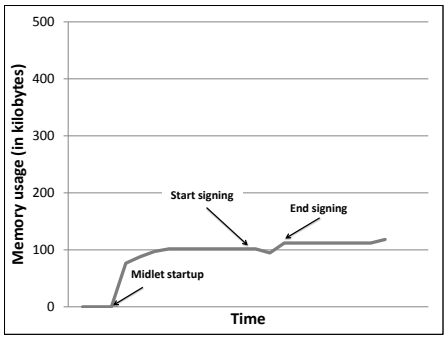

Figure 4.7: Memory usage profile of **RSA** when processing a 1024-bit signature

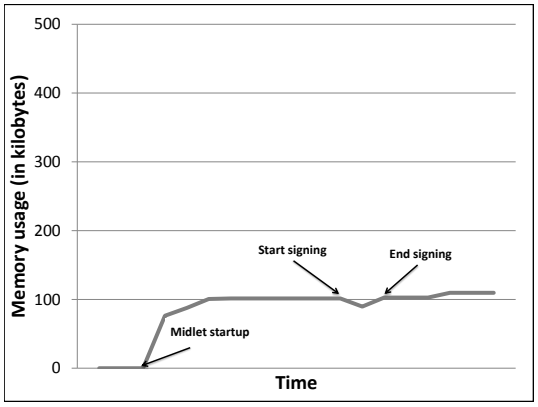

Figure 4.8: Memory usage profile of **DSA** when processing a 1024-bit signature

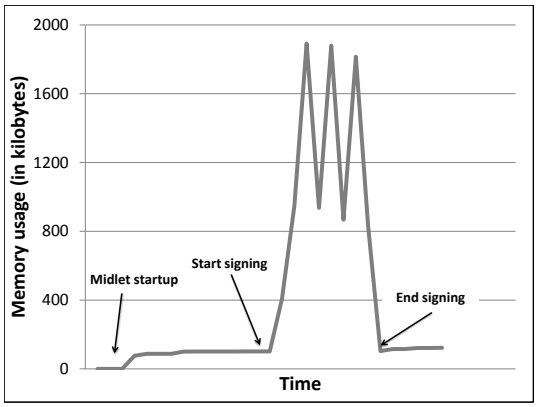

Figure 4.9: Memory usage profile of **ECDSA** when processing a 160-bit signature

4.7.2 Energy efficiency

A cryptosystem running on a mobile device may put its CPU on a heavy load and significantly drain the underlying battery, as witnessed by several contributions in this field (56, 57, 58). This consumption is proportional to the execution time of the cryptosystem and to the complexity of the involved cryptographic operations. The expectations are that performing a signature using ECDSA instead of RSA or DSA is less energy-expensive because this cryptosystem uses simpler operations and shorter keys. When performing a signature verification, it is also expected that RSA is much more energy saving than DSA and ECDSA since it is able to perform this operation faster.

Starting from these premises, we measured the energetic consumption of the three

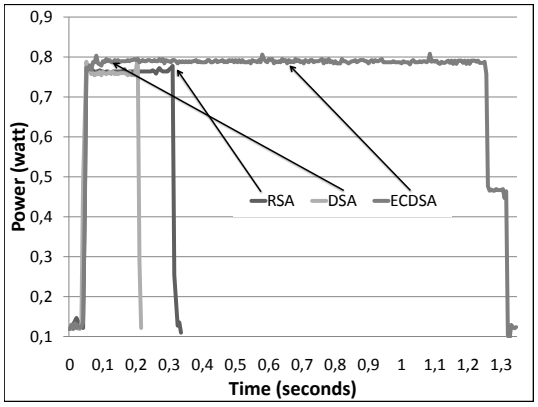

Figure 4.10: N95-8GB power consumption when signing a message using a 1.024 bits key

cryptosystems when performing a signature and a verification on a message using a security level equivalent to 1.024 bits RSA key.

The measurements have been taken by running the SEESMS client application on a Nokia N95-8GB using the Nokia Energy Profiler tool. Figure 4.10 shows the energy required to perform one signature using the cryptosystems currently supported by SEESMS. Despite the expectations, the energetic cost of the ECDSA algorithm ($\sim 0,79$ Watts) is higher than RSA and DSA algorithms ($\sim 0,76$ Watts). Moreover, since ECDSA execution time is longer than the other two algorithms, its overall energy consumption ($\sim 1,04$ Joule) results to be larger than RSA ($\sim 0,25$ Joule) and DSA ($\sim 0,15$ Joule).

Figure 4.11 shows the energy consumption of one signature verifications using the supported cryptosystems with a key strength of 1.024 bits. Even in this case the Watts consumption for the ECDSA algorithm ($\sim 0,77$ Watts) is higher than RSA (~ 0.70 Watts) and DSA algorithms ($\sim 0,68$ Watts). Moreover, it is interesting to observe that the overall energetic consumption of the ECDSA algorithm ($\sim 1,23$ Joule) is higher than RSA ($\sim 0,03$ Joule) and DSA ($\sim 0,20$ Joule), due to its longer verification time.

4.8 Optimization

The poor overall performance of ECDSA and its suspicious memory usage graph motivated us in investigating the quality of the implementation we have been using for

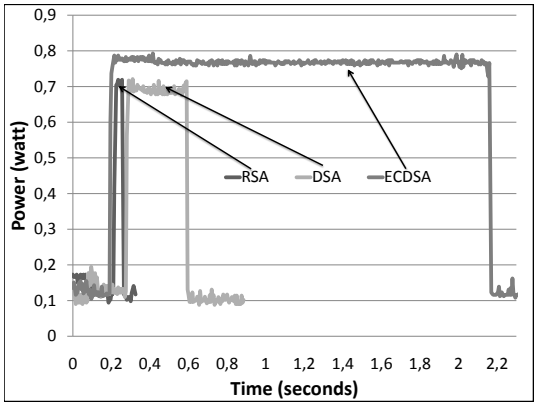

Figure 4.11: N95-8GB power consumption when verifying a message using a 1.024 bits key

this algorithm. The purpose of this investigation was to understand if these bad performance were due to algorithmic reasons or to some implementation defects. In this section, we further profile the inner behavior of the ECDSA implementation we have been using, we pinpoint some serious performance issues and, finally, we experiment with several optimizing algorithmic techniques in order to improve its speed. The discussion is organized according to the chronological order we have followed when trying the different optimizations, with all the succeeding optimization techniques applied in an incremental way over the original ECDSA and, in some case, RSA cryptosystems.

4.8.1 Optimizing memory usage

In our previous experiments on the memory usage of ECDSA (see, e.g., Figure 4.9), we have observed that there may be some issues with the way this cryptosystem manages its own memory. Namely, we have seen that ECDSA requires about 100 times the memory used by the equivalent RSA implementation. Moreover, we noticed that during its execution, the algorithm performs many memory allocations/deallocations. Indeed, this behavior may have a strong negative influence on the performance of the algorithm because of the time overhead required to perform memory related operations.

Starting from these observations, we focused our attention on the data types used by the ECDSA implementation available with the Bouncy Castle library, and on the way they are used in the implementation. A quick analysis revealed that the most

resource-consuming data type used by this algorithm is the one implemented by the `BigInteger` class, which serves to store an integer number of arbitrary length. The weak point of this data type is that is implemented as an *Immutable* object, that is: every time an instance of this object has to change the value it represents, a new instance has to be created to store the new value. It is interesting to note that the original `BigInteger` implementation available with the J2SE framework relies on the existence of an additional `BigInteger` implementation which is *mutable*. Instead, the J2ME framework does not implement the `BigInteger` class, in fact it is provided by the BC library.

As a confirmation of our intuition, we have seen that during the processing of a 160 bit based signature on a mobile device, ECDSA required the allocation of approximately 100.000 `BigInteger` objects. Instead, by running the same code on a desktop environment and using the native J2SE `BigInteger` implementation, we have seen that the overall number of allocations dropped to 3.700.

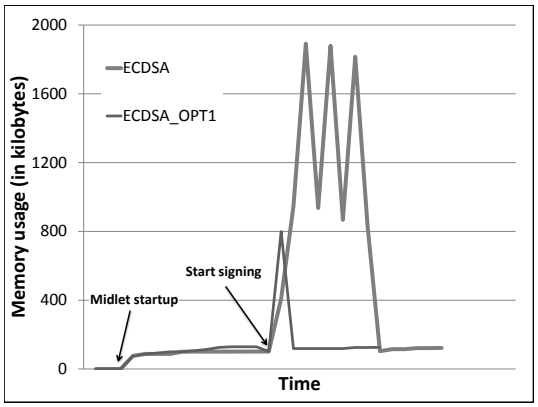

Figure 4.12: Memory usage profile of **ECDSA** and **ECDSA_OPT1** when processing a 160-bit signature

We then optimized **ECDSA** by replacing the original `BigInteger` class coming with BC with a porting of the mutable one available with the J2SE framework. The resulting code, which we called **ECDSA_OPT1**, exhibits a much more regular memory usage pattern than **ECDSA** and requires about 800 kbytes instead of 5.400 kbytes, as shown in Figure 4.12. This optimization has also a significant impact on the running times of the algorithms, which are now about six times faster than the original implementation (see Table 4.2). We performed the same optimization on the RSA

implementation by replacing the original `BigInteger` class with the new one. The resulting implementation, **RSA_OPT1**, is about the 30% faster than the original RSA implementation.

Table 4.2: ECDSA signature times (in ms) on an N95-8GB device

Curves	ECDSA	ECDSA_OPT1	RSA	RSA_OPT1
secp112r2	461	112	49	27
secp128r2	596	132	86	63
secp160r2	1020	184	236	164
secp192r1	1651	332	735	499
secp224r1	2343	481	1666	1127
secp256r1	3096	568	5503	3588

4.8.2 Optimizing running times

Despite the memory optimizations described in Section 4.8.1, **ECDSA_OPT1** still remains significantly slower than **RSA** when used to perform digital signatures with large keys. A careful profiling of this algorithm revealed that, in this case, it spends more than the 95% of its execution time to perform *scalar multiplications*, i.e., the product of a big scalar value and the representation of a curve point in affine coordinates. According to (59), given a scalar $k = \{k_0, k_1, \ldots, k_{m-1}\}$, this operation can be defined as:

$$kP = \sum_{i=0}^{m-1} k_i(2^i P)$$

Considering that the expected number of ones in the binary representation of k is about $m/2$[1], the expected number of operations needed to carry out a scalar multiplication is approximately $m/2$ point additions and m point doublings.

A common approach to the optimization of this operations uses, first, the Non-Adjacent Form (59) (NAF) technique to minimize the number of points additions to do, and then, the Fixed-base Windowing method (60) to precalculate some of the intermediate values required by a scalar multiplication.

[1] In our context k is the private key.

4.8.2.1 The NAF algorithm

According to (59), a NAF of a scalar k is a *signed digit representation* in the form of $k = \sum_{i=0}^{l-1} k_i 2^i$ where $k_i \in \{0, \pm 1\}$, $k_{l-1} \neq 0$, and no two consecutive digits k_i are nonzero. The *length* of the NAF is l.

For every integer positive k, the NAF expression has the following properties which can be exploited to improve the performance of the elliptic curve scalar multiplier:

1. k has a unique NAF denoted NAF(k).
2. NAF(k) has the fewest nonzero digits of any signed digit representation of k.
3. The length of NAF(k) is at most one more than the length of the binary representation of k.
4. If the length of NAF(k) is l, then $2^l/3 < k < 2^{l+1}/3$.
5. The average density of nonzero digits among all NAFs of length l is approximately 1/3.

NAF(k) can be efficiently computed using an algorithm that generates the digits of the NAF(k) by repeatedly dividing k by 2.

NAF can be defined in a more general way using a parameter w (window) and hence processing w digits of k at time.

Let $w \geq 2$ be a positive integer. A *width-w* NAF of a scalar k is a *signed digit representation* in the form of $k = \sum_{i=0}^{l-1} k_i 2^i$ where each nonzero coefficient $|k_i|$ is odd, $|k_i| < 2^{w-1}$, $k_{l-1} \neq 0$, and at most one of any w consecutive digits is nonzero. The *length* of the *width-w* NAF is l.

Let k be a positive integer.

1. k has a unique *width-w* NAF denoted $NAF_w(k)$.
2. $NAF_2(k) = NAF(k)$.
3. The length of $NAF_w(k)$ is at most one more than the length of the binary representation of k.
4. The average density of nonzero digits among all *width-w* NAFs of length l is approximately $1/(w+1)$.

4.8.2.2 The fixed-base windowing method

The *fixed-base windowing* method for point multiplication exploits the fact that if the point P is known (as in the case of ECDSA) and some storage is available, then the point multiplication can be sped up by precomputing some data which depends only on P. For example, if the points $2P$, 2^2P, $2P$. ..., $2^{m-1}P$ are precomputed, then the expected time required for the scalar multiplier would be $m/2$ additions.

Let $(k_{d-1}, \ldots, k_1, k_0)_{2^w}$ be the 2^w-ary representation of k, where $d = \lceil m/w \rceil$, and let $Q_j = \sum_{i:k_i=j} 2^{wi}P$. Then

$kP = \sum_{i=0}^{d-1} k_i(2^{wi}P) = \sum_{j=1}^{2^w-1}(j \sum_{i:k_i=j} 2^{wi}P) = \sum_{j=1}^{2^w-1} jQ_j$.

Hence

$$kP = Q_{2^w-1} + (Q_{2^w-1} + Q_{2^w-2}) + \ldots + (Q_{2^w-1} + Q_{2^w-2} + \ldots + Q_1). \tag{4.1}$$

The *fixed-base windowing method* is based on (4.1) and its expected running time is approximately $((d(2^w - 1)/2^w - 1) + (2^w - 2))A$.

Table 4.3: Overall number of ECPoint objects initializations, additions and doublings required by **ECDSA** and **ECDSA_OPT2**

	ECDSA	ECDSA_OPT2
ECPoint Init	239	54
Addition	35	29
Doubling	159	0

4.8.2.3 Experimental results

We implemented another variant of **ECDSA**, called **ECDSA_OPT2**, that uses the window-NAF coding for the scalar k and the fixed-base windowing method. The size w of the window has been chosen in such a way to optimize the trade-off between the number of multiplications saved by precomputation and the number of fields operations required to perform it. In Table 4.3 we report some statistics about the improvement on the overall number of operations to be performed when producing an ECDSA signature using **ECDSA_OPT2** rather than **ECDSA**. This improvement affects also the performance of **ECDSA_OPT2** which results to be much faster than **ECDSA_OPT1**,

as clearly visible in Table 4.4: the performance improvement is noteworthy as this new variant requires less than the 10% of the time required by **ECDSA_OPT1** and about the 2% of the time required by **ECDSA**. We also observe a significant improvement on the memory usage of **ECDSA_OPT2**, which is approximately the 10% of the memory used by **ECDSA_OPT1** (see Figure 4.13).

Table 4.4: ECDSA signature times (in ms) on an N95-8GB device

Curves	ECDSA	ECDSA_OPT1	ECDSA_OPT2
secp112r2	461	112	17
secp128r2	596	132	24
secp160r2	1020	184	38
secp192r1	1651	332	43
secp224r1	2343	481	46
secp256r1	3096	568	61

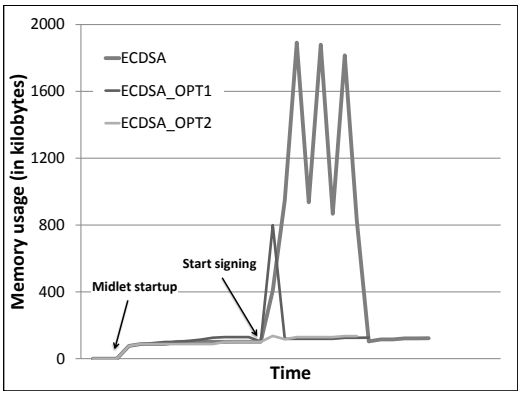

Figure 4.13: Memory usage profile of **ECDSA**, **ECDSA_OPT1** and **ECDSA_OPT2** when processing a 160-bit signature

4.8.3 Overall experimental results

The several optimizations discussed so far led to **ECDSA_OPT2**, a variant of **ECDSA** that exhibits a significant performance improvement with respect to the original implementation. We have been able to improve as well the performance of **RSA** through the implementation of **RSA_OPT1**, a variant of the original implementation that uses

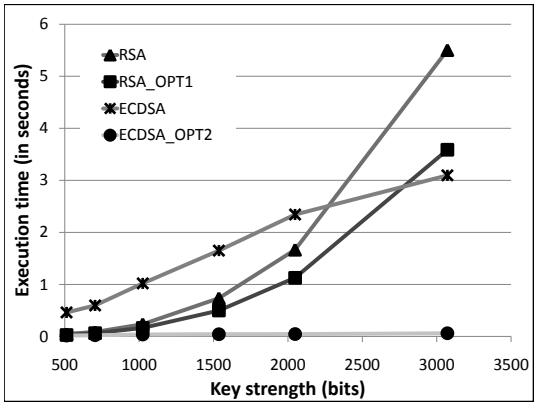

Figure 4.14: RSA and ECDSA signature generation times (in ms) on an N95-8GB device using optimizations

the optimized version of the `BigInteger` data type. We now turn our attention to the way these new implementations compare to each other from differ viewpoints. Concerning time performance, **ECDSA_OPT2** is extremely more efficient than **ECDSA** when signing messages and performs much better than the RSA-based implementations, also for short keys. This is shown in Figure 4.14. We also observed a significant speed-up of **ECDSA_OPT2** over **ECDSA** when verifying the signature of a message, as visible in Figure 4.15. In this case, the RSA-based implementations remain the fastest cryptosystems, however **ECDSA_OPT2** exhibits approximately the same order of growth. Concerning memory usage, the space requirements of **ECDSA_OPT2** are slightly higher than those of **RSA_OPT1** because of the overhead to be paid for precalculating and storing in memory the points to be used by the *fixed-base windowing* method (see Figure 4.16).

Finally, we briefly discuss the impact of these optimizations on the overall amount of energy consumed by the considered algorithms. The Watt consumption per second is almost unaffected by all the optimizations we have introduced however, the shorter execution times imply smaller amount of energies to be spent for performing the signature and verification operations, as it can be seen in Figures 4.17 and 4.18. Anyway, the energetic cost of these operations does not have a significant impact on the typical battery life of a smartphone device.

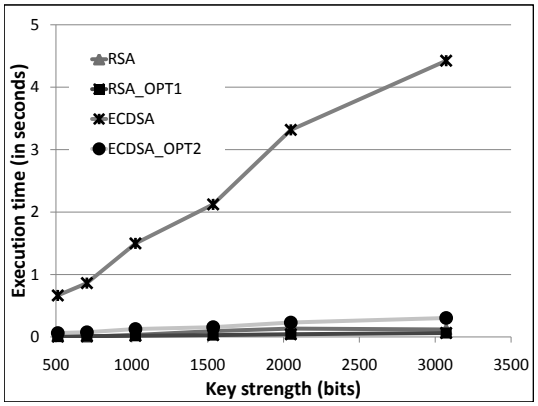

Figure 4.15: RSA and ECDSA signature verification times (in ms) on an N95-8GB device using optimizations

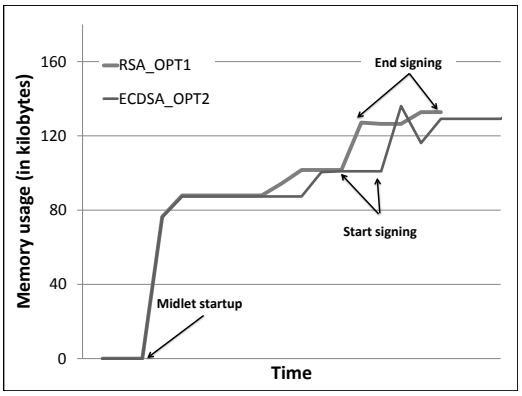

Figure 4.16: Memory usage profile of **RSA_OPT1** and **ECDSA_OPT2** when processing a 1024-bit and a 160-bit signature respectively

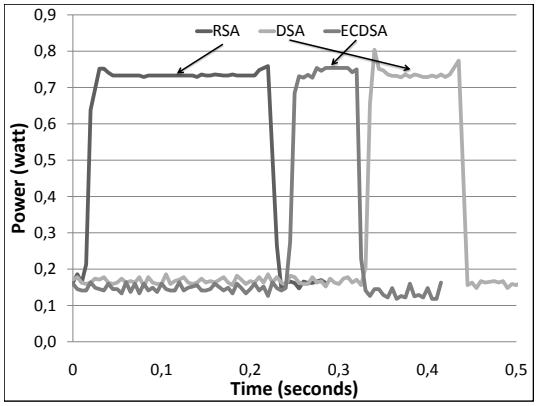

Figure 4.17: N95-8GB power consumption when signing a message using a 1.024 bits key using optimized algorithm and implementation

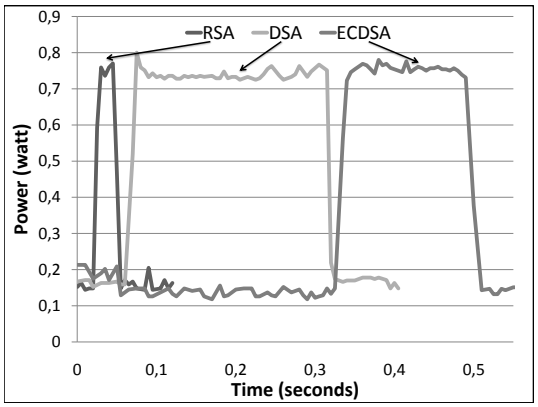

Figure 4.18: N95-8GB power consumption when verifying a message using a 1.024 bits key using optimized algorithm and implementation

5

A System for Secure Communication over 3G Networks

5.1 Introduction

Public and private sectors extensively rely upon 3G mobile networks for communicating sensitive technical, financial, political and personal informations. Videoconference is a service increasingly used for business and top-management affairs. A reliable videoconference service might be available always and everywhere. For example, a company manager might be able to perform a business videoconference in high-mobility environments as traveling by train or in isolated places as staying on boat quite far from the coast. 3G mobile networks provide circuit-switched channels with bandwidth and Quality of Service (QoS) constraints which cannot always be satisfied by IP-based connections in relation to reachability and availability requirements.

Contrary to IP networks, terminals in 3G networks are not affected by addressing problems: every device equipped with an USIM is automatically connected to the mobile-phone network and available through its telephone number to other mobile devices. Moreover, the UMTS network provides a suitable Quality of Service for real-time communications: guaranteed (even though low) bandwidth (64Kb/s), latency constraints and network-level transparency. The circuit-switched connection between terminals is carried out and managed by the network, simplifying the video-telephony

application structure.

Telecommunication companies have spent a big effort for the spreading, the global coverage, and the amount of different services provided to the users, such as videotelephony, text messaging and data communication, but less effort have been spent to provide suitable end-to-end security mechanism for those services. In fact, voice, text and video communications carried by means of 3G networks have been shown to be vulnerable to eavesdropping and unauthorized access. Both wireless data link and wired part of the network suffer several security threats.

Services provided by the modern third-generation (3G) telecommunication technology can be split into two classes, each in charge of two different network domains: Circuit Switched (CS) and Packet Switched (PS) domain. Different domains provide different services, for example, videotelephony is routed to the CS domain while IP networking take place in the PS domain.

As long as videotelephony is essentially a data communication, both CS and PS domains on the 3G networks are suitable for this service. The adoption of the PS has many advantages but also some important drawbacks with respect to the CS. The biggest advantage consists in the use of the IP protocol, which ensure interoperability with almost all network applications for both mobile and desktop devices. IP security protocols such as SSL, IPSEC and VPN, and already designed systems for end-to-end security at application level can be re-used almost without modifications. Even if effective and easy-to-use solutions have been presented for voice/video communications over IP networks, an effective and reliable videotelephony service has reachability and availability requirements which can't be always satisfied by an IP based mobile network. As long as the the mobile networks have not a global coverage, handover processes and high Bit Error Rate (BER) on the wireless links tend to make the channel unstable. On the contrary, the CS network connections are more stable in high-mobility and high-BER conditions, granting reachability and availability requirements to the users.

In this work we defined a metric to evaluate the communication service based on the following key performance indicators: voice and video quality, user data privacy and application impact on the battery life. These metrics are often related to the domain that provide the service. Generally, videotelephony over the CS domain results in a better quality and battery saving with respect to the PS domain. In fact, the CS connection between the endpoints is provided by the telecommunication network

with a higher QoS value and is managed by the BSs, instead of the IP connection that is to be managed by the peers. To the contrary, applications running over the CS domain must comply with inflexible network protocols through which is hard to provide the same security level of the applications running over IP. Scope of this work is to improve security of videotelephony over CS domain, in order to achieve an optimal quality/security/battery-life tradeoff.

5.1.1 Trusting

When an end-user subscribes a cell-phone contract implicitly trusts the network operator which is supposed to preserve the communication privacy. Therefore, mobile operator must guarantee adequate security measures to accomplish this task. In 3G mobile networks, whereas channel encryption has been adopted for wireless links, user data on wired links is transmitted in clear introducing several threats. Moreover, 3G networks extensively rely on roaming following agreement signed between two or more operators. This means that end-users should trust the roaming operator too as long as the physical telecommunication channel is managed by another company.

Network and intra-network domain security are actually covered by Mobile Application Part Security (MAPSec)(15) protocol, that provide security support for the MAP(61) protocol. The MAP protocol plays a central role in the signaling communications between the Network Elements (NEs). User profile exchange, authentication, and mobility management are performed using MAP. MAP typically runs over the Signaling System number 7 (SS7) protocol stack(62). MAPSec protocol protects MAP messages trough a packet-encapsulation mechanism.

It is important to note that the mobile station is not affected by network domain security. The two communicating NEs may both be in the same network administrated by the same telecommunication operator or they may belong to two different networks administrated by two different companies. Because MAPSec only provides encryption of MAP signalling messages but not user traffic, unauthorized communication interceptions, performed by malicious users who have access to the physical telecommunication infrastructure, are a real threat.

Encryption at network level may raise some issues. Assuming that an end-to-end user communication pass through a lot of network elements, interoperability between different mobile operator networks may be harder to reach because cryptographic keys

management and security protocols should be deployed among network elements managed by different providers. Moreover, there exists authorities that must be able to perform phone tapping, for example in many countries there are laws against terrorism which enforce the ability of the operators to intercept the communications related to a suspect user. Encryption at network level and cryptographic keys management between each couple of network elements may also interfere with these tasks.

Substantially, user traffic may be transmitted encrypted through all network links and entities to avoid unauthorized interceptions, encryption at network-level may cause administrative issues due to VNOs, and at the same time a third-trusted-part must be able to easily decrypt communications to allow authorized phone tapping.

The solution proposed in this chapter is an end-to-end security mechanism for 3G video-telephony based on a cryptosystem able to include fair mechanisms for key-escrow allowing authorized network eavesdropping.

5.1.2 Security background

In 1999, with the standardization process of third-generation (3G) mobile-phone networks, the 3GPP[1] consortium, which is in charge of producing technical specifications for the 3G mobile networks, proposed improved security mechanisms for wireless channel with respect to the GSM ones, which were shown to be vulnerable. The 3GPP introduced a stronger authentication and key agreement (AKA) protocol performed by mobile device and network, and a stronger cryptosystem for wireless data encryption (A5/3 based on KASUMI(63)). However, effective attacks which can lead to the decryption of the communication channel has been discovered. Keller and Shamir presented an attack on KASUMI which allows to recover a full A5/3 key using a related-key attack(64). Karsten et al.(65) showed how to perform a semi-active attack jamming UMTS frequencies and forcing the mobile device to switch in GSM mode. Using the Karsten technique an attacker can ask the USIM to reuse a previous A5/3 key for a breakable obsolete GSM encryption protocol, as A5/1, and can decrypt a previously intercepted conversation.

There exists some projects addressing application-level end-to-end security of voice communications over mobile-telephony networks, as SPEECH which is discussed in this Thesis. However, until this Thesis, it doesn't seem to exist any public research

[1]The 3rd Generation Partnership Project (3GPP), `http://www.3gpp.org/`

addressing application-level security of *video-telephony* communications over 3G mobile networks.

5.1.3 Outline

The rest of the chapter is organized as follow. Section 5.2 presents the state of the art in the voice and video communication security solutions. In Section 5.3 the 3G-324M communication protocol is described while in Section 5.5 the integrations aimed to enhance the security of the protocol are discussed.

5.2 Requirements

The main goal of this work is to present a system that realize a secure video-call over 3G mobile networks with following requirements:

- **Strong end-to-end user authentication through digital certificates.** To achieve the strong end-to-end user authentication requirement, we suppose that the user has requested and received a X.509 digital certificate issued by a trusted Certification Authority (CA) stored securely on the mobile equipment.
- **End-to-end user communication encryption.** The audio, video and data channel encryption requirement can be achieved using robust and well-known encryption algorithms. The authentication protocols must also implement key-agreement mechanisms in order to initialize the channel encryption.
- **Compatibility with video-telephony protocols.** The proposed solution must be compatible with existing applications and must be possible to implement it with no modifications to the video-telephony protocol.
- **Infrastructure-side transparency.** It is required that no extra effort should be spent by the network elements to realize the described end-to-end security mechanisms, therefore the protocol data and encrypted data must be transmitted between users as normal network traffic.
- **Limited impact on system performance.** The introduction of security mechanisms should not considerably affect the system performance and the user experience. For instance, the initial handshake should not delay the communication

for an amount of time longer than few seconds. Analogously, during the conversation, communication delays due to data encryption should not be longer than 300 milliseconds.

- **Device constraints.** In order to cope with the low processing power and to save battery life of the mobile devices, local computations and end-to-end communications must me minimized. To meet this, the secure video-call system should implement public-key encryption schemes based on the Elliptic Curve Cryptography[1] (ECC) instead of traditional public key cryptosystems.

5.3 Video-telephony over UMTS

In this section a briefly introduction to the 3G video-telephony protocol is presented in order to explain as it can be extended with support for security mechanisms.

3G-324M(66) is the 3rd Generation Partnership Project (3GPP) umbrella protocol for video telephony in 3G mobile networks. The 3G-324M protocol operates over an established circuit switched connection between two communicating peers. 3G-324M is based on the International Telecommunication Union - Telecommunication standardization sector (ITU-T)[2] H.324M specification for multimedia conferencing over Circuit switched networks.

3G-324M does not support end-to-end user-level security mechanisms but security only depends on network-level protocols in charge of the network.

3G-324M is a derivative of the existing ITU-T recommendation H.324 for low bitrate media communication, which was initially intended for video telephony using modem-based communication over the GSTN.

One of the key concepts of H.324 is that of logical channels. *Logical channel number 0* (LCN0) is dedicated to control information. Using LCN0 the terminal can immediately start sending and receiving control information following the syntax and semantics defined in ITU-T recommendation H.245, allowing it to declare its capabilities and discover the capabilities of the remote terminal.

[1] With respect to RSA, ECC offers equivalent security with smaller key sizes, which results in faster computations, lower power consumption as well as memory and bandwidth savings.

[2] International Telecommunication Union - Telecommunication Standardization Sector, `http://www.itu.int/ITU-T/`

H.223 provides the multiplexing function for H.324.

The *simple retransmission protocol* (SRP) layer between the H.245 control process and the multiplexer is designed to provide reliable, acknowledged transmission of control information.

Annex C of H.324 was introduced as a result of studies on how H.324 could be adapted for use over wireless and mobile networks, initially focusing on such technologies as *digital enhanced cordless telecommunications* (DECT). H.324 with Annex C has become known as H.324M. Annex C removes the modem requirements for H.324 and assumes that a transparent digital channel is available. Annex C provides enhanced mechanisms for robustly multiplexing and error resilience with respect to H.324.

Annex C also provides enhanced procedures for more reliable delivery of H.245 control messages by specifying the use of numbered SRP (NSRP) instead of SRP. A layer known as the *control segmentation and reassembly layer* (CCSRL) is introduced between the H.245 signaling entities and the NSRP layer. The purpose of CCSRL is to segment larger control messages, to reduce the susceptibility of control messages to error.

5.3.1 3G-324M

3GPP has taken H.324M as starting point and modified it to create 3G-324M. For Release 99 the principal differences between 3G-324M and H.324M are in the codecs supported. The mandatory audio codec for 3G-324M is GSM-AMR(67), H.263(68) remains the mandatory video codec and MPEG4 Simple Profile Level 0 is added to the list of optional video codecs.

The Fig.5.1 shows the extended 3G-324M architecture: the white modules compose the original protocol, the gray ones are added to provide support for security mechanisms.

The Application Layer (AL) represents the existing video-telephony application and is not part of the 3G-324M protocol stack. It interacts with the H.245 module which provide the support to set up and manage the call. The Control of Security Layer (CSL) is an added middleware which provide to the AL the possibility to initialize and manage security protocols. It uses the reliable H.245 protocol to interact with the peer. The AL also sends to and receives data from the Audio/Video (A/V) modules, which take in charge the communication of the multimedial streams. A/V data can

pass trough an Encryption Layer (EL) which provide the encryption functions for the outgoing streams and the decryption functions for the ingoing streams.

The multiplexing and de-multiplexing procedures are addressed by the H.223 protocol, which directly interact with the 3G modem to communicate with the network.

It is important to note that the proposed extensions do not impact over the existing video-telephony protocol and can be transparently integrated within 3G-324M.

5.3.1.1 Considerations on the video codecs

The video codecs employed in the 3G-324M protocol have particular characteristics which are to be considered for the correct design of the 3G-324M security framework.

On the transmitter side, H.263 and MPEG-4 codecs produce variable-length frames which are divided into Group Of Blocks (GOBs) to be sent at the multiplexing level. Every GOB has a resynchronization marker to reduce the error propagation caused by the nature of Variable Length Code (VLC) into single frame. The resynchronization marker is inserted at the top of a new GOB with the header information so that decoding can be done independently.

On the receiver side, blocks are received one-by-one from the multiplexing layer and are buffered until the reception of a resynchronization marker, which indicates the begin of a new GOB. The received GOB is recomposed using the information of its header and is passed to the video codec for the decoding.

5.4 A secure video-calling system

The 3G-324M-Security project (3G-324M-Sec) consists of a framework which extend 3G-324M and allows to integrate security mechanisms into the existing video-telephony protocol, without modifications to the video-telephony protocol. The 3G-324M-Sec high-level architecture is shown in the Fig.5.1. The white components represent the existing 3G-324M modules, explained in the Section 5.3, and the grey components are the 3G-324M-Sec meta-modules. The Security Control Layer includes the configuration module and the key-agreement and authentication protocols. The Encryption Layer includes the audio, video and text encryption mechanisms. 3G-324M-Sec supports a lot of security-related functions as user authentication through digital certificates, communication encryption and data integrity.

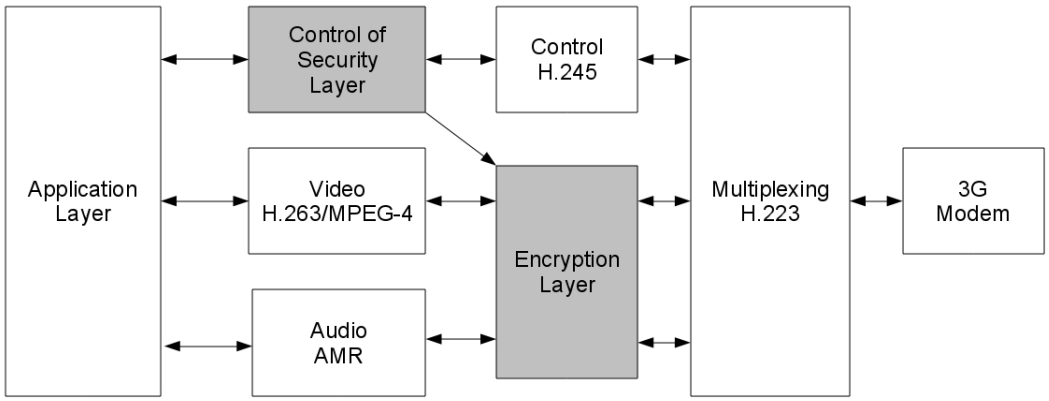

Figure 5.1: 3G-324M-Sec high-level architecture

Efficiency requirements are addressed employing robust and well-known implementations of the encryption protocols and techniques to reduce the communication overload. Cryptographic algorithms themselves have been chosen according to these requirements, for example, elliptic-curve based algorithms have been adopted as long as they guarantee the same security level of traditional encryption algorithms more efficiently.

3G-324M-Sec is compatible with UMTS network protocols. The 3G-324M-Sec control data is exchanged through the H.245 control protocol and the encrypted data is encapsulated in fully UMTS-compatible network packets. The transmissions are network-side transparent. The system also performs an auto-discovery procedure to determine if the peer is compatible or not with the security extensions.

Robust, well-known and efficient technologies have been employed:

5.4.1 Authentication and key-agreement

Whenever the two video-calling applications established the communication channel through the 3G-324M protocol, the users can initiate a secure conversation running a key agreement protocol through the 3G-324M-Sec extensions. The purpose of these protocols is to generate a common session key to be used to encrypt voice, video and text data streams and, optionally, to verify the identity of the parties of the conversation.

3G-324M-Sec supports three different forms of user authentication and key agreement schemes, each with a different level of security.

Those schemas produce a shared random secret which is processed by the PBKDF2 (Password-Based Key Derivation Function) - which is part of RSA Laboratories' Public-Key Cryptography Standards (PKCS) series, specifically PKCS #5 v2.0 (69) - to generate the cryptographic keys for the symmetric ciphers currently in use.

- **Elliptic Curve Diffie-Hellman key agreement.** Whenever two users initiate a new conversation, 3G-324M-Sec permits to run the 521-bit prime Elliptic Curve Diffie-Hellman (ECDH) (70) key-exchange protocol to agree on a common secret key. This form of agreement does not guarantee to the user the identity of the other end-point of the conversation but it is enough when we are just interested in guaranteeing the confidentiality of a conversation.

- **Passphrase based key agreement.** Two users interested in having a secure conversation choose a common passphrase. Whenever a new secure conversation has to be initiated, they will generate each a new common secret using the shared passphrase. The reuse of the same passphrase is always possible, because the generated common secret, and consequently the session keys, will be never the same as the key-exchange algorithm is based on the exchange of encrypted random values. This approach provides with a basic form of authentication since it is expected that the passphrases are known only by their legitimate owners.

- **Certificate based key agreement.** Two users initiating a new secure conversation own a legitimate X.509 digital certificate which has been previously loaded in their device. Moreover, the certificates of the root CAs must be available on the devices in order to verify the validity of the peer certificate. If these conditions are met, the two parties use the standard TLS 1.0 protocol[DiAA99] to perform the mutual authentication and keys agreement. The call originator plays the role of client in the TLS protocol while the receiver of the call play the server role. According to the TLS specification, each client submits its X.509 certificate and provides its verification.

5.4.2 Encryption

AES with 256-bit key in OFB mode has been chosen to encrypt the communication channels. The AES algorithm is one of the most commonly employed encryption standard and has been chosen due to its proven robustness and efficiency. The OFB mode

avoids bit error propagation and does not affect error resilience mechanism of underlying communication levels.

The audio, video and text user data can be encrypted after the previous key-agreement phase.

5.4.3 Data integrity

HMAC-MD5 function is implemented to provide a data-integrity mechanism.

5.4.3.1 Keys

The PBKDF2 function derives the cryptographic keys from the secret shared through the key-agreement protocol. It generates six 256-bit cryptographic keys, each employed for a single unidirectional data stream:

- Output audio encryption
- Input audio decryption
- Output video encryption
- Input video decryption
- Output IM encryption / HMAC generation
- Input IM decryption / HMAC verification

5.4.4 Side effects

A textual Instant Messaging (IM) protocol with security extensions has been implemented exploiting the H.245 control channel. The IM protocol is a proof-of-concept utility which demonstrates that arbitrary data can be exchanged between the video-call participants.

This result opens the way to a large number of promising applications, for example, it could replace the actual SMS technology used for device-to-device alerting systems. Unlike SMS, the designed messaging protocol can guarantee reliability and real-time delivery.

5.4.5 3G-324M-Sec prototype

A 3G-324M-Sec prototype has been developed in order to test the robustness and the efficiency of the system in a real environment. The experiments has been conducted with success using personal computers equipped with UMTS Tokens. The prototype do not require a high level of technical knowledge therefore can be used by non-skilled users in a large set of real use-cases.

5.5 Designing a security framework for UMTS video-telephony

3G-324M-Sec has been designed using a bottom-up approach, that is, lower-level modules have been first projected.

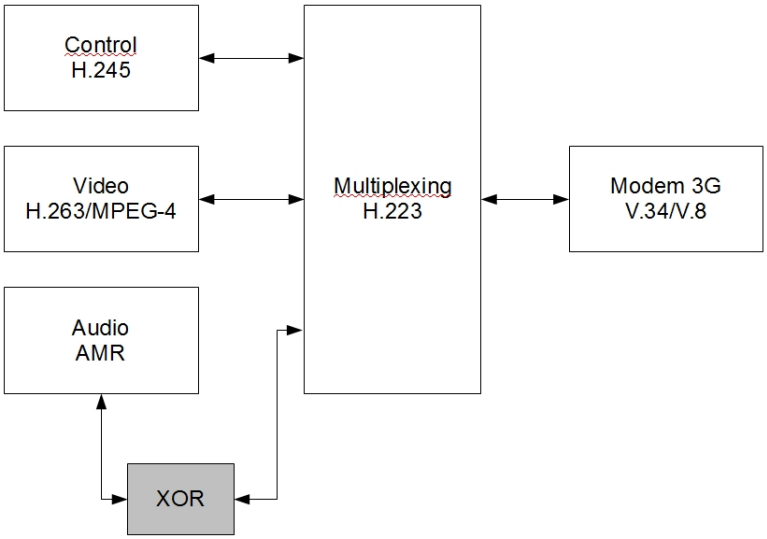

Figure 5.2: XOR-module

5.5.1 Proof of concept

The 3G-324M-Sec protocol expects that non-standard data can be transmitted between the video-call users, that is, audio/video encrypted packets can be routed through the telecommunication network and the network entities can treat them as common packets. To confirm this hypothesis a preliminary experiment has been conducted.

As shown in Fig.5.2, a XOR-module is integrated in the existing protocol stack to test the reaction of the network to the transmission of audio packets containing encrypted payload data. The new module is placed between the audio codec module and the multiplexing module. It performs a one-time-pad xor encryption and decryption, respectively, of the output and input audio data using a pre-loaded key.

A Video XOR module similar to the Audio XOR module has been introduced in the 3G-324M protocol to verify the possibility to exchange encrypted video packets during a video-call. Whereas the Video XOR module can be placed between the video codec module and the multiplexer module as the Audio XOR (Fig.5.2), it must have a different behavior. In fact, it must consider the particular characteristics the video codec, as explained in 5.3.

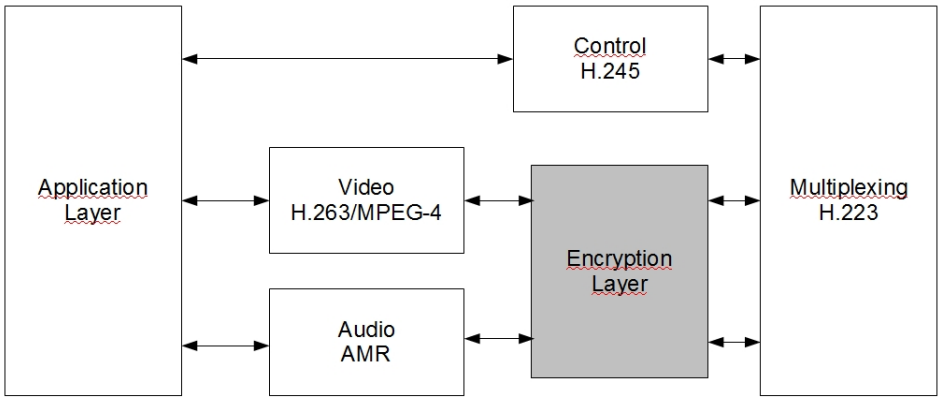

Figure 5.3: Encryption Layer

5.5.2 Encryption layer

The Audio and Video XOR experiment have been conducted with success and confirmed that the network-level protocols are completely unaware of the application-level traffic, so application-level encryption is possible.

The XOR module shown in Fig.5.2 has been replaced by the Encryption Layer (EL), as shown in Fig.5.3. The EL is composed of two sub-modules which are the Audio EL and the Video EL.

The EL implements AES to encrypt/decrypt the audio/video streams. Four cryptographic keys are generated using the PBKDF2: audio encryption key, audio decryption

key, video encryption key, video decryption key. These are derived from a common pre-shared secret.

On the transmitter, the Audio EL takes the AMR audio as input and encrypt it using AES. The Video EL runs a similar procedure encrypting not the entire video frame but the single blocks, as explained in 5.5.1.

On the receiver, the EL behavior is reversed. The EL takes encrypted packets coming from the multiplexing module as input and use the appropriate key to decrypt them. The plain-text data is subsequently passed to the higher protocol layers.

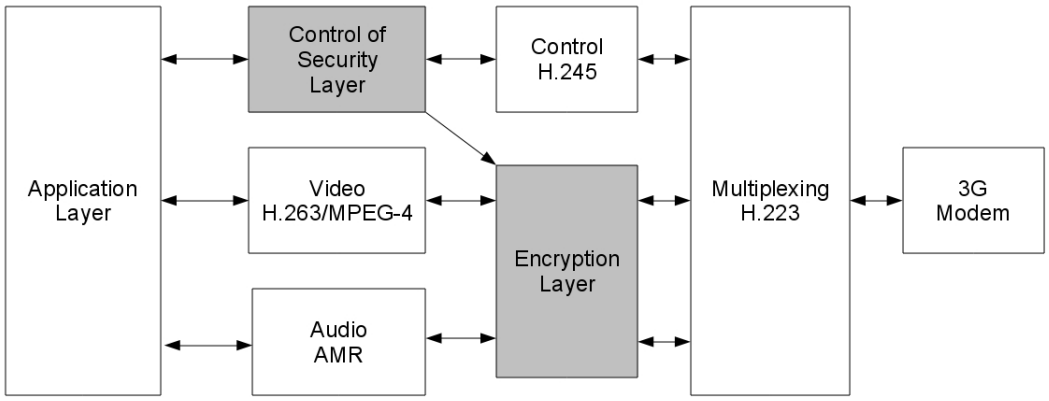

Figure 5.4: Control Layer

5.5.3 Control layer

The EL described in 5.5.2 has a basic behavior. It derives the cryptographic keys from a common pre-shared secret not performing any key-agreement or authentication protocol. The Control Layer (CL) is introduced to provide procedures to configure and manage the security mechanisms. As shown in Fig.5.4, it directly interacts with the AL and the EL.

At this design level, it provides a procedure to specify the passphrase used by the PBKDF2 function which generates the stream-specific cryptographic keys. Moreover, it provides a function to enable/disable on-the-fly the security extensions. The on-the-fly activation function can be useful, for example, to avoid the cryptographic overhead if encryption is not necessary.

5.5.4 Reliable transport adapter layer

Guaranteed data delivery is crucial for the proper functioning of Authentication and Key-Agreement protocols, that is, an only single bit error causes the entire re-execution of the protocol. The extremely high bit error rate of the wireless links ($10^{-3} \leq BER \leq 10^{-2}$) is worked around implementing a reliable transport layer over the H.245 control protocol.

As explained in the section 5.3, H.245 provides reliable, acknowledged transmission of control information and segmentation of larger messages. Moreover, it provides support for user-defined communication through the UserInputIndication message, originally introduced to simulate the transmission of DTMF tones over digital networks. In particular, the UserInputIndication message is defined in the H.245 standard using the ASN.1 encoding and permits to exchange arbitrary alphanumeric strings between the peers.

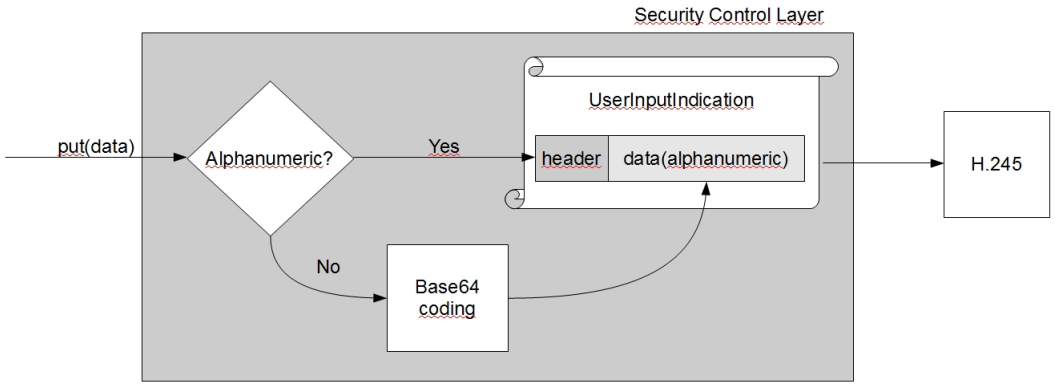

Figure 5.5: Reliable Transport Adapter Layer - message transmission

The Reliable Transport Adapter Layer (RTAL) exploits the UserInputIndication mechanism to provide procedures to open virtual streams between the peers and transmit general-purpose data structures. The RTAL interface exposes a basic symmetric non-blocking *put()*, which send data on an opened virtual stream, and a symmetric blocking *get()*, which receive data from a opened virtual stream. The Fig.5.5 schematize the RTAL operation of sending data.

5.5.5 Authentication

The introduction of the RTAL opened the way to the implementation of a largest set of cryptographic protocols over 3G-324M. In particular, the system incudes some robust well-known authentication and key-exchange protocols as SSL and ECDH (see the section 5.4 for the specifications). To integrate an existing implementation of the SSL protocol, it is enough to create an adapter which instantiates the virtual streams through the RTAL functions and binds the imported I/O streams to the new ones.

5.5.6 Session control layer

The Session Control Layer (SCL) is subsequently introduced to completely abstract the cryptographic protocols from the 3G-324M-Sec system. It performs all the session-specific operations as cryptographic keys management, EL initialization, virtual streams initialization. At the same time, it abstracts the AL from the underlying authentication protocol.

With the introduction of the RTAL and the growing complexity of the cryptographic protocols implemented in 3G-324M-Sec, the old CL became divided in two sub-layers: the previously described RTAL and the new Session Control Layer (SCL). The overall 3G-324M-Sec architecture is shown in Fig.5.6.

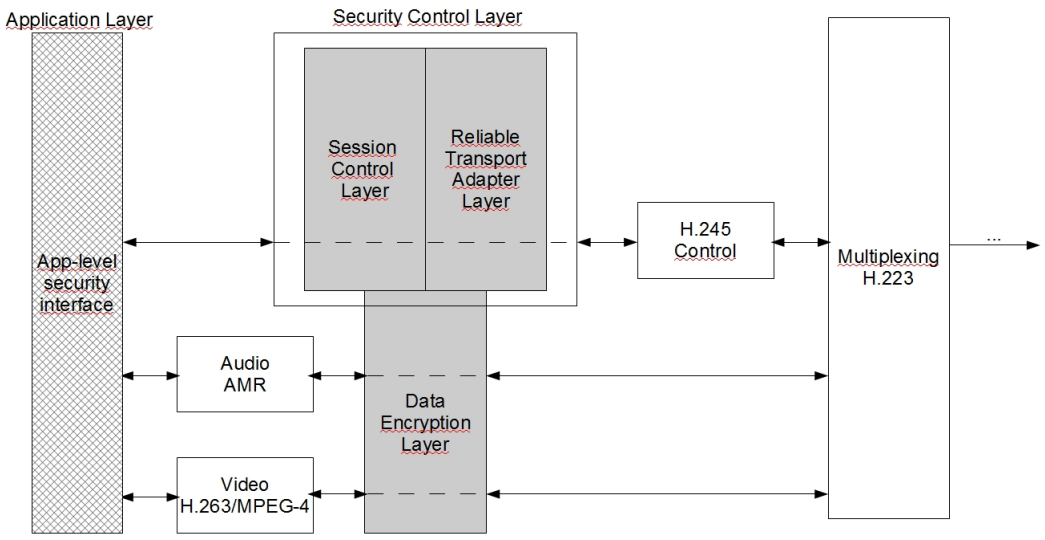

Figure 5.6: Overall 3G-324M-Sec structure

5.5.7 Secure instant messaging protocol

A Secure Instant Messaging (SecIM) protocol has been also designed exploiting the H.245 control channel. The SecIM protocol provides encryption and data integrity for textual messages. It demonstrates the flexibility of the RTAL managing general-purpose communication protocols.

The SecIM protocol can be used during both secured and non-secured video-calls, without service interruption. It can be useful, for example, to communicate a credit card number, payment informations etc.

5.5.8 Implemented protocol stack

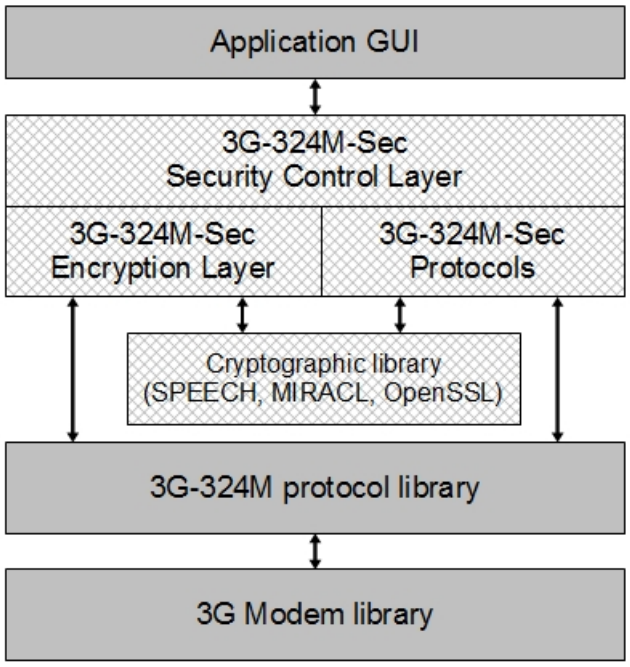

Figure 5.7: 3G-324M-Sec Implementation Stack

A 3G-324M-Sec prototype working in a real environment has been developed. The prototype runs on PC equipped with UMTS usb tokens and has been developed for Windows XP/7 platforms.

The overall implementation is in standard ANSI C++ and the used IDE is Visual Studio 2008. The project consists of more than three thousands of source files and headers. OpenSSL has been used to implement the SSL handshake. Implementations

of ECDH, Passphrase authentication and AES has been imported from the SPEECH project, which employs the MIRACL[1] library for high-precision operations on elliptic curves.

The Fig.5.7 shows the 3G-324M-Sec implementation stack. The video-call users interact with the 3G-324M-Sec GUI module, which sends commands to the 3G-324M-Sec application logic. The SCL module manages the authentication and key-agreement procedures, and interacts with the EL, which creates the encrypted data channels. The data delivery is demanded to the 3G-324M protocol implementation, which directly interacts with the hardware.

5.6 System performance

In order to gain a deeper understanding about the impact produced by our extension to the original 3G-324M protocol we arranged an experimental analysis of our prototype. Performance evaluation in such context is known to be hard task. This is mainly due to the unpredictable side-effects introduced by the network elements (BS) used to connect the two endpoints. In fact, depending on the BS's load and distance this kind of links can only increase the system entropy. Therefore our testing methodology is only aimed to measure the effect introduced by the security modules/protocols on the overall system performances compared with respect to the performances obtained using the standard (unencrypted) protocols. Whereas we are not concerned in channel bandwidth and reliability measurements.

The realized prototype has been experimented in a real environment and has shown to be robust and flexible enough to be adopted in many network conditions. The key-agreement and authentication protocols have been tested with success and the network has never rejected the encrypted packets. The employed 3G-324M protocol implementation has not shown to be performance effective for this specific application. The overall system performance can be improved employing a more performing 3G-324M protocol implementation.

A large set of experiments have been performed to measure the delay introduced by the 3G-324M-Sec cryptographic extensions with respect to the common video-call. In this section a significant set of experimental results are commented.

[1]Multiprecision Integer and Rational Arithmetic C/C++ Library at `http://www.shamus.ie/`

5.6.1 Methodology

The methodology adopted to measure the system performances rely on the key performance indicators discussed in 5.1.

The security strength of the secsec324 framework rely on the robustness of the cryptographic protocols that have been employed. Even though the cryptanalysis of such algorithms is out of the scope of this work, it is possible to consider the secsec324 system secure as long as the employed cryptosystems are considered secure.

The battery consumption strongly depends on the implementation of the secsec324 specifications and protocols. The realized prototype do not run more processes or threads with respect to the standard video-call implementation, and the amount of computations and data-exchanges is minimized since the implemented protocols rely on the ECC cryptography. It has been empirically measured that a secure video-call performing a SSL handshake and lasting 5 minutes do not discharge the device battery more than 1/10 of the original video-call application.

The service quality has been evaluated empirically, and the secsec324 implementation has not shown to cause any detectable degradation of the user experience with respect to the standard video-call. The empiric results have been confirmed by the analysis of the delays introduced by the protocols and the encryption procedures.

5.6.2 Experimental setup

The prototype has been experimented using these hardware platforms:

- PC1: Notebook with CPU Intel Core Duo T2300 at 1.66GHz, SDRAM DDR2 1GB
- PC2: PC with CPU Intel Core Duo 2 E6400 at 2.13GHz, SDRAM DDR3 2GB

These devices have been used for the 3G connectivity:

- UMTS Pendrive Onda MSA523HS
- UMTS Pendrive Onda MDC502HS

The prototype has been tested on these software environments:

- Windows XP SP3

- Windows 7 Professional

The execution time of the prototype procedures has been measured using the high-resolution performance counter library, included in the Windows API:

- QueryPerformanceCounter()

About a hundred of experiments has been conducted for each protocol in different environments having different UMTS network coverage, capacity and traffic load. For example, it can be assumed that the UMTS signal strength is good in the department laboratory and the traffic load varies during daytime. It can be also assumed that the experiments conducted at home, located in a small provincial town, are affected by worse UMTS signal coverage.

For the sake of simplicity, PC1 and PC2 were always nearby during the experiments, so it is possible to suppose that they were in the same cell and received the same UMTS signal. As explained in 5.6.3, the system performance strongly depends on the signal quality, which is supposed to be equivalent for both endpoints. The experiments discussed in 5.6.3 have been conducted in the department laboratory, having good UMTS signal coverage quality, so it is possible to suppose that the measured performances are the best possible for that specific signal quality. In other words, these results indicate an upper bound for the performance achievable by the 3G-324M-Sec prototype in a real usage.

5.6.3 Protocols performances

Concerning the protocols performance, the execution time strongly depends on the amount of data exchanged between the peers and the instantaneous BER of the communication channel ($10^{-2}...10^{-3}$).

In substance, if $T(p)$ is the execution time of the protocol p:

$$T(p) = T_c(p) + C$$

where $T_c(p)$ is the time utilized by the protocol p to exchange its data structures between the peers and C is the time utilized for the local computations.

The C value is strongly dependent on the processing power of the underlying hardware, and can be assumed constant for a specific protocol running on a specific hardware

platform. The minimum values always have been measured on PC2, the maximum values on PC1.

The $T_c(p)$ value depends on the UMTS signal quality and the instantaneous BER. Generally, the experiments have shown that $T_c(p)$ dominates C. In substance, $T_c(p)$ is the most relevant term of $T(p)$, therefore it is possible to approximate $T(p) \approx T_c(p)$.

It is important to note that the video-telephony service is not interrupted during the protocols execution, which runs over the out-of-band H.245 control channel. As long as the control channel has limited bandwidth [1] and the 3G-324M protocol do not provides mechanisms to increase its capacity, the performances of the 3G-324M-Sec protocols decrease with increasing amount of data to be exchanged. For example, the SSL handshake results in a significative performance loss if large size certificates are used.

5.6.3.1 Passphrase authentication protocol

Both the peers generates a cryptographic key using the pre-shared passphrase and use it to encrypt a data structure containing their session-specific key share and initialization vector, then send it to the other peer. The session key is generated through the appropriate combination of the local and the remote key shares. Each transmitted data structure is 304 byte.

The time required to transmit and receive the data structures is varying between 1678,73 and 2100,03 milliseconds, depending on the instantaneous BER. The time required for the local computations is negligible: $C < 1msec$ both on PC1 and PC2. The results are summarized in Tab.5.1.

Table 5.1: Delay introduced by the Passphrase Authentication Protocol

	$\mathbf{T_c(Passphrase)}$	$\mathbf{C_{passphrase}}$
Maximum	2100,03 msec	$<$ 1 ms
Minimum	1678,73 msec	$<$ 1 ms

[1] The 3G-324M standard (66) do not specify the bandwidth reserved for the control channel, so it depends on the specific protocol implementation.

5.6.3.2 Diffie-Hellman key agreement

The Elliptic Curve Diffie Hellman protocol for a 521 bit key agreement consists of some local computations (which most significant are a pair of scalar multiplications) to calculate the ECDH parameters, which are exchanged among the peers to obtain the common secret. Each exchanged parameter is 70 bytes.

Table 5.2: Delay introduced by the ECDH-521 data exchange

	$\mathbf{T_c(Diffie-Hellman)}$	$\mathbf{C_{Diffie-Hellman}}$
Minimum	1171,92 msec	4 msec
Maximum	1363,27 msec	12 msec

The Tab.5.2 summarizes the measured maximum and minimum delay: it varies between 1171,92 and 1363,27 milliseconds depending on the instantaneous BER. In this case the C parameter is more significant: the execution time of the local computations varies between 4 (the minimum measured on PC2) and 12 milliseconds (the maximum measured on PC1).

It is important to note that the ECDH-521 average communication time is lesser than the Passphrase protocol because the data to be exchanged is significantly smaller.

5.6.3.3 SSL handshake

The delay introduced by the SSL handshake has been tested using basic X.509 digital certificates and extended X.509 certificates containing user's photos. The execution time strongly depends on the certificate size, which is significantly greater if photo is added.

All the experiments have been conducted using the following certificates:

- *ClientCert* - 512-bit RSA certificate: 645 bytes
- *PhotoClientCert* - 512-bit RSA certificate with photo: 2622 bytes
- *ServerCert* - 512-bit RSA certificate: 645 bytes
- *PhotoServerCert* - 512-bit RSA certificate with photo: 2622 bytes

The SSL handshake messages have different size depending on the employed certificates:

1. *ClientHello*: 94 byte.
2. *ServerHello + Server certificate*: 1525 byte using ServerCert, 3502 bytes using PhotoServerCert.
3. *Client certificate*: 1651 bytes using ClientCert, 3756 bytes using PhotoClientCert.
4. *Finish*: 75 bytes.

The performance analysis has shown a communication delay of 6325,2 - 9227,11 milliseconds using the basic X.509 certificate and 17950,45 - 27876 milliseconds using the X.509 certificates with photo.

The local timings mainly contains the execution time of the ECDH key exchange (i.e. two scalar multiplications on the client) and the signature verification. The remaining time is spent for protocol processing and for administrative purposes.

The Tab.5.3 shows the measured maximum and minimum timings.

Table 5.3: Delay introduced by the SSL handshake

	$\mathbf{T_c(SSL)}$		$\mathbf{C_{SSL}}$	
	Basic X.509	*Photo X.509*	*Basic X.509*	*Photo X.509*
Maximum	17950,45 ms	27876 ms	12 ms	17 ms
Minimum	6325,2 ms	9227,11 ms	9 ms	11 ms

5.6.4 Encryption delay

The EL introduces delay in local computations, therefore delays the audio/video packets transmission and reproduction. Since the AES decryption algorithm is symmetric with respect to the encryption one, and in general all the pre-transmission procedures are symmetric to the post-reception ones, it can be considered the same delay interval Δ for encryption and decryption. It is important to note that Δ only depends on the local computations, therefore is strongly related to the computational capability of the underlying hardware.

The Tab.5.4 shows the average per-packet delay measured on PC1 during an encrypted video-call, and the average delay introduced every 60 seconds.

It is important to note that the total delay is proportioned to the quantity of the processed packets, resulting lower for the video traffic which has a lower sampling rate.

Table 5.4: Delay introduced by the local computations on PC1 during a video-call

	Data Type	
Average	**Audio**	**Video**
per-packet delay	0.00821606 ms	0.0112831 ms
packets sent every 60sec	3000	900
delay every 60sec	24,65 ms	10,15 ms

The local computations introduce delay having minimum variability with respect to the communication delay, therefore the local audio and video per-packet delays can be considered constant on the specific hardware:

$$\Delta_{audio}^{PC1} \approx 0.008 milliseconds$$

$$\Delta_{video}^{PC1} \approx 0.01 milliseconds$$

The Tab.5.5 summarizes the same experiment conducted on PC2. As expected, the local delays on PC2 are lesser than the local delays on PC1, because PC2 has a greater processing power. The constant per-packet delays on PC2 are:

$$\Delta_{audio}^{PC2} \approx 0.0042 milliseconds$$

$$\Delta_{video}^{PC2} \approx 0.0046 milliseconds$$

Table 5.5: Delay introduced by the local computations on PC2 during a video-call

	Data Type	
	Audio	**Video**
Call duration	77012.8 ms	77012.2 ms
Number of packets	3837	1170
Average per-packet delay	0.00423322 ms	0.00467853 ms
Total delay	16.2428 ms	5.47388 ms

The Encryption Layer delays the audio and video streams by an amount of time undetectable by an human user. As stated in 5.6.3, the most of the delay is introduced by the authentication and key-agreement protocol execution. A number of improvements for the 3G-324M-Sec performances and features are discussed in 5.7. However,

assuming that the user authentication only happens once at the beginning of the conversation, and the subsequent media stream encryption do not deteriorate the overall system performance, it can be concluded that the implemented 3G-324M-Sec prototype is suitable for deployment in a real environment.

5.7 Future works

The experimental results of the 3G-324M-Sec protocol appear promising and encourage further research. In particular, the 3G-324M-Sec project is a starting point to design mechanisms for the authenticity, integrity and non repudiability of the conversation, in order to provide a multimedial communication service which have legal validity and can be used for remote interrogations, contracts signing, remote purchases etc. Aiming on this purpose, we are currently working on the extensions presented below.

5.7.1 User certificate in the SIM card.

The achieved implementation of the SSL protocol over 3G-324M for user-authentication and key agreement is an important result and can motivate the introduction of digital certificates within USIMs, both simplifying and reinforcing the realized security infrastructure. However, this task is demanded to the mobile telephone companies.

5.7.2 Audio/Video integrity.

A possible extension of 3G-324M-Sec is the support for data integrity at Encryption Layer. This task may be difficult due to the inflexibility of the communication protocols. Audio and video packets have fixed size and it is possible to append information about data integrity, for example a HMAC code, reducing the payload size. This solution led to audio/video quality loss and do not agree with the 3G-324M specifications.

A possible solution is to exploit the H.245 control channel to send the integrity information separately. However, it important to note that audio/video packets are not numbered and packet loss is highly probable due to the BER, which increase the difficult to realize this solution.

5.7.3 Non-repudiation.

It would be useful for both parties to have, at the end of the communication, the same identical copy of the conversation. Such a task is not easy as it seems because both the underlying communication channel and the transport protocol may be unreliable, and so audio/video packets can be damaged or be lost.

Non-repudiation mechanisms can be designed over the 3G-324M-Sec framework. The out-of-band H.245 channel could be used to perform the protocol, for example, sending the digital signature and the integrity information of the packets.

5.7.4 Performance improvements.

The H.245 control channel represents a performance bottleneck due to its limited bandwidth. As discussed in 5.6.3, the most of the delay is introduced by the protocol data transfers. It is possible to speed up the protocol data transfers designing a reliable transport layer over the audio/video channels, which have more bandwidth with respect to the control channel.

6

Conclusions

As the functionality of mobile devices becomes more diverse, Security in wireless cellular communication has become an increasingly critical issue among consumers and providers. Financial transactions as well as confidential information are daily transmitted on wireless networks and the majority of users are unaware of the security issues they may face. The main problem is that mobile communication networks do not provide any end-to-end security mechanism. Hence, eavesdropping or phishing attacks are relatively easy to carry out. In this Thesis the security limits of the most deployed and spread cellular communication technologies have been presented and security systems for voice, video and text communications have beed designed and developed: SPEECH, SEESMS and SECR3T.

The first part of this Thesis provided an introduction to Security concepts related to wireless communications. Specifically, the first chapter showed an historical overview of the three generation of cellular networks and basic information on what is meant by Security in general for telecommunication systems. A short introduction on our secure communication tools is also given. Chapter two introduced a technical view about the Security systems adopted by network carriers to protect the wireless part of the communication path between two communicating users and the attacks to the Security algorithm employed.

The second part of the Thesis was focused on the design and the implementation of SPEECH, SEESMS and SECR3T, our Security systems designed and developed for secure, end-to-end communication over GSM (voice), SMS (text) and UMTS (voice, video and text) communication channels. Those systems are software-only solutions

and in particular do not require any specialized hardware configuration or device. All of them work at application level on mobile devices and desktop PC.

SPEECH is a software tool installable on modern handheld devices, which allows to communicate in encrypted, authenticated and signed mode using the GSM communication channel. The part of SPEECH concerning Security is completely independent from the bearer service in use. It currently supports TLS 1.0 and preshared passphrase as authentication/key agreement protocols, Diffie-Hellman as key agreement protocol and channel encryption with AES256. The voice is encoded with the Speex codec which allows to converse in fullduplex mode still with 9600 bps of band usage as upper bound. The non-repudiation of the conversation was the main challenge. To achieve it, we needed to design and implement a reconciliation protocol on the conversation content which is applicable if a limited number of errors occurs. Moreover a signature mechanism on the agreed conversation content has been provided as proof.

SEESMS is a software framework that allows two peers to exchange encrypted and digitally signed SMS messages. SEESMS differs from the other frameworks presented so far in literature, because it allows users to choose which cryptosystem and which degree of Security to use. Three cryptosystems are built-in, RSA, DSA and ECDSA and further can be added thanks to his modular architecture. An experimental analysis of the cryptosystems available in SEESMS using several different metrics has been conducted. A careful profiling of this library revealed some performance issues that were responsible for the bad performance of ECDSA. We then tried some algorithmic and programming optimization techniques for improving the performance of ECDSA. As a result of these optimizations, we obtained two variants of the original RSA and ECDSA implementations coming with the Bouncy Castle library which exhibit substantially faster execution times and a reduced memory footprint.

SECR3T project demonstrated that it is possible to integrate cryptographic mechanisms within the 3G-324M (and most generally H.324) protocol in a totally transparent way for the telecommunication company, preserving compatibility with the 3G network specifications, with a minimal delay of the communication. The SECR3T framework consists of an implementation of an extended 3G-324M protocol which includes Security features. Authentication and Key-Agreement protocols have been designed using the reliable H.245 control channel. The SSL handshake protocol provides strong user authentication employing X.509 digital certificates. The SECR3T framework provides

encryption for voice, video and textual communication. It supports the installation of new cryptographic protocols through its extendibility interface.

Our three software tools allow to overcome the Security limits of mobile communications networks. PKI infrastructures and X.509 digital certificates are used to allow end-to-end user authentication and proper Security algorithms and protocols ensure the confidentiality of the communication content. Digital signature by means of X.509 digital certificates guarantees the non-repudiation of the communication content. Using this tools, users can accomplish critical transactions (such as m-commerce, trading online, phone banking, homeland Security) using low cost handhelds, standard communication channels and without requiring additional dedicated hardware.

References

[1] Hideki Imai. *Wireless Communications Security*. Artech House, Inc., Norwood, MA, USA, 2005. 3

[2] Balderas-Contreras. **Security Architecture in UMTS Third Generation Cellular Networks - Technical Report**, 2-2004. http://ccc.inaoep.mx/Reportes/CCC-04-002.pdf. 4

[3] 3GPP-The 3rd generation partnership project. http://www.3gpp.org/, January 2011. 5

[4] H. Yang, F. Ricciato, S. Lu, and L. Zhang. **Securing a Wireless World**. *Proceedings of the IEEE*, **94**(2):442 –454, feb 2006. 5, 6

[5] Eduardo B. Fernandez, Imad Jawhar, Maria M. Larrondo-Petrie, and Michael VanHilst. **An overview of the security of wireless networks**, 2004. http://polaris.cse.fau.edu/~ed/WirelessSecSurv4.pdf. 7

[6] Eduardo B. Fernandez, Saeed Rajput, Michael VanHilst, and Maria M. Larrondo-Petrie. **Some Security Issues of Wireless Systems**. **3563**:388–396, 2005. 8

[7] Olivier Benoit, Nora Dabbous, Laurent Gauteron, Pierre Girard, Helena Handschuh, David Naccache, Stéphane Socié, and Claire Whelan. **Mobile Terminal Security**. Cryptology ePrint Archive, Report 2004/158, 2004. http://eprint.iacr.org/. 19, 27

[8] Alex Biryukov, Adi Shamir, and David Wagner. **Real Time Cryptanalysis of A5/1 on a PC**. In Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, and Bruce Schneier, editors, *Fast Software Encryption*, **1978** of *Lecture Notes in Computer Science*, pages 37–44. Springer Berlin / Heidelberg, 2001. 20

[9] David Wagner Ian Goldberg. **Rump session of Crypto '99**. In *CRYPTO*, 1999. 20

[10] K. Boman, G. Horn, P. Howard, and V. Niemi. **UMTS security**. *Electronics Communication Engineering Journal*, **14**(5):191 – 204, October 2002. 20

[11] Elad Barkan, Eli Biham, and Nathan Keller. **Instant Ciphertext-Only Cryptanalysis of GSM encrypted communication**. In *CRYPTO*, pages 600–616. Springer-Verlag, 2003. 20, 27

[12] Overview of 3GPP release 99. http://www.3gpp.org/article/release-1999, January 2011. 21

[13] Overview of 3GPP release 4. http://www.3gpp.org/article/release-4, January 2011. 21

[14] Overview of 3GPP release 5. http://www.3gpp.org/article/release-5, January 2011. 21

[15] 3GPP TS 33.200, 3G Security; Network Domain Security (NDS); Mobile Application Part (MAP) application layer security (release 6) V6.1.0 (2005-03). http://www.3gpp.org/ftp/specs/html-info/33200.htm, January 2011. 24, 67

[16] Kent and Atkinson. **Security Architecture for IP**. http://www.ietf.org/rfc/rfc2401.txt, 1998. 25

[17] 3GPP TS 33.210, 3rd Generation Partnership Project; Technical Specification Group Ser- vices and System Aspects; 3G Security; Network Domain Security; IP network layer secu- rity (release 6) V6.5.0 (2004-06). http://www.3gpp.org/ftp/specs/html-info/33210.htm, January 2011. 25

[18] M. Rahnema. **Overview of the GSM system and protocol architecture**. *Communications Magazine, IEEE*, **31**(4):92 –100, April 1993. 26, 27

[19] Jovan Dj. Golic. **Cryptanalysis of alleged A5 stream cipher**. In *Proceedings of the 16th annual international conference on Theory and application of cryptographic techniques*, EUROCRYPT'97, pages 239 – 255, Berlin, Heidelberg, 1997. Springer-Verlag. 27

[20] Casper Technology. http://www.caspertech.com/, January 2011. 29

[21] **Gesellschaft fur Sichere Mobile Kommunikation mbH**. http://www.cryptophone.de/, January 2011. 29

[22] **General Dynamics C4 Systems**. http://www.gdc4s.com/, January 2011. 29

[23] **VectroTEL**. http://www.vectrotel.ch/, January 2011. 29

[24] **Global Teck**. http://www.global-teck.com/, January 2011. 29

[25] **SecureGSM**. http://www.securegsm.com/, January 2011. 29

[26] Frederic P. Miller, Agnes F. Vandome, and John McBrewster. *Advanced Encryption Standard*. Alpha Press, 2009. 29

[27] **Skype phone**. http://www.skype.com, January 2011. 29

[28] Jean marc Valin. **Speex: A Free Codec For Free Speech**. http://www.speex.org/, January 2011. 30

[29] M. Schroeder and B. Atal. **Code-excited linear prediction(CELP): High-quality speech at very low bit rates**. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '85.*, **10**, pages 937 – 940, April 1985. 32

[30] **Nautilus Secure Phone**. http://nautilus.berlios.de/, January 2011. 33

[31] W. Diffie and M. Hellman. **New directions in cryptography**. *Information Theory, IEEE Transactions on*, **22**(6):644 – 654, November 1976. 35

[32] T. Dierks and C. Allen. **The TLS Protocol Version 1.0**. http://www.ietf.org/rfc/rfc2246.txt, 1999. 35

[33] Adi Shamir. **How to share a secret**. *Commun. ACM*, **22**:612–613, November 1979. 37

[34] D. Eastlake, 3rd and P. Jones. **US Secure Hash Algorithm 1 (SHA1)**. http://www.ietf.org/rfc/rfc3174.txt, 2001. 38

[35] Portio Research. http://www.portioresearch.com/MMF10-14_press.html, July 2010. 39

[36] Tele-Log. http://www.tele-log.com/domotica-e.html, 2010. 39

[37] Aginova Inc. http://www.aginova.com/, 2010. 39

[38] Clickatell SMS API. http://www.clickatell.co.za/blog/2009/04/sms-beer-alert-you-are-in-danger-of-running-out/, 2010. 39

[39] Carta SI. http://www.cartasi.it/download/AM_Rimini.pdf, 2010. 39

[40] Aniello Castiglione, Roberto De Prisco, and Alfredo De Santis. **Do You Trust Your Phone?** In *EC-Web 2009: Proceedings of the 10th International Conference on E-Commerce and Web Technologies*, **LNCS 5692**, pages 50–61, Berlin, Heidelberg, 2009. Springer-Verlag. 40

[41] A. Hossain, S. Jahan, M.M. Hussain, M.R. Amin, and S.H. Shah Newaz. **A proposal for enhancing the security system of short message service in GSM**. In *Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on*, pages 235–240, Aug. 2008. 41

[42] He Rongyu, Zhao Guolei, Chang Chaowen, Xie Hui, Qin Xi, and Qin Zheng. **A PK-SIM card based end-to-end security framework for SMS**. *Computer Standards & Interfaces*, **31**(4):629–641, 2009. 41

[43] IPCS Group. http://www.ipcslive.com/pdf/IPCSSMS.pdf, 2010. 41

[44] M. Toorani and A.A. Beheshti Shirazi. **SSMS - A secure SMS messaging protocol for the m-payment systems**. In *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pages 700–705, July 2008. 41

[45] Marko Hassinen, Konstantin Hyppönen, and Keijo Haataja. **An Open, PKI-Based Mobile Payment System**. In *ETRICS*, pages 86–100, 2006. 42

[46] M. Hassinen. **SafeSMS - end-to-end encryption for SMS**. In *Telecommunications, 2005. ConTEL 2005. Proceedings of the 8th International Conference on*, **2**, pages 359–365, 15-17, 2005. 42

[47] Ugo Chirico. **Message in a Bottle**. http://www.ugosweb.com/miabo/, 2010. 42

[48] D. Lisonek and M. Drahansky. **SMS Encryption for Mobile Communication**. In *Security Technology, 2008. SECTECH '08. International Conference on*, pages 198–201, Dec. 2008. 42

[49] Marko Hassinen and Smile Markovski. **Secure SMS messaging using Quasigroup encryption and Java SMS API**. In *SPLST*, pages 187–200, 2003. 42

[50] A. Grillo, A. Lentini, G. Me, and G.F. Italiano. **Transaction Oriented Text Messaging with Trusted-SMS**. In *Computer Security Applications Conference, 2008. ACSAC 2008. Annual*, pages 485–494, Dec. 2008. 42

[51] Johnny Li-Chang Lo, Judith Bishop, and J.H.P. Eloff. **SMSSec: An end-to-end protocol for secure SMS**. *Computers & Security*, **27**(5-6):154–167, 2008. 42

[52] Don Johnson, Alfred Menezes, and Scott A. Vanstone. **The Elliptic Curve Digital Signature Algorithm (ECDSA)**. *Int. J. Inf. Sec.*, **1**(1):36–63, 2001. 43

[53] Certicom Research. **Standards for efficient cryptography – SEC 1: Elliptic Curve Cryptography**. http://www.secg.org/download/aid-385/sec1_final.pdf, 2010. 43

[54] The Legion of Bouncy Castle. **Bouncy Castle Crypto API**. http://www.bouncycastle.org/java.html, 2010. 44

[55] Certicom Research. **Standards for efficient cryptography – SEC 2: Recommended Elliptic Curve Domain Parameters**. http://www.secg.org/download/aid-386/sec2_final.pdf, 2010. 50

[56] Nachiketh R. Potlapally, Srivaths Ravi, Anand Raghunathan, and Niraj K. Jha. **A Study of the Energy Consumption Characteristics of Cryptographic Algorithms and Security Protocols**. *IEEE Transactions on Mobile Computing*, **5**(2):128–143, 2006. 54

[57] Kihong Kim, Jinkeun Hong, and Jongin Lim. **Analysis of the Power Consumption of Secure Communication in Wireless Networks**. In *DEXA*, pages 894–903, 2006. 54

[58] A.S. Wander, N. Gura, H. Eberle, V. Gupta, and S.C. Shantz. **Energy analysis of public-key cryptography for wireless sensor networks**. In *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, pages 324–328. IEEE Computer Society Washington, DC, USA, 2005. 54

[59] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone. *Guide to Elliptic Curve Cryptography*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003. 58, 59

[60] Michael Brown, Darrel Hankerson, Julio López, and Alfred Menezes. **Software Implementation of the NIST Elliptic Curves Over Prime Fields**. In *CT-RSA*, pages 250–265, 2001. 58

[61] Group Core Network 3rd Generation Partnership Project, Technical Specification. **Mobile Application Part (MAP) Specification (Release 5), 3GPP TS 29.002, 2002.** http://www.3gpp.org/ftp/Specs/html-info/29002.htm, 2002. 67

[62] ITU-T. **ITU-T Recommendation Q.700**. http://www.itu.int/rec/T-REC-Q.700/en, 1993. 67

[63] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects. **3G Security; Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification**, 12-2000. http://www.3gpp.org/ftp/Specs/html-info/35202.htm. 68

[64] Orr Dunkelman, Nathan Keller, and Adi Shamir. **A Practical-Time Attack on the A5/3 Cryptosystem Used in Third Generation GSM Telephony**. Cryptology ePrint Archive, Report 2010/013, 2010. http://eprint.iacr.org/. 68

[65] Nohl Karsten and Chris Paget. **GSM: SRSLY?** In *26th Chaos Communication Congress*, 2009. 68

[66] 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects. **Codec for circuit switched multimedia telephony service; Modifications to H.324 (Release 9)**. http://www.3gpp.org/ftp/Specs/html-info/26111.htm, 2009. 70, 85

[67] 3GPP technical specification. **3GPP TS 26.190; Transcoding functions**. http://www.3gpp.org/ftp/Specs/html-info/26190.htm, 2009. 71

[68] ITU-T. **ITU-T Recommendation H.263**. http://www.itu.int/rec/T-REC-H.263/en, 2007. 71

[69] RSA Laboratories. **PKCS5: Password-Based Cryptography Specification Version 2.0**. http://tools.ietf.org/html/rfc2898, 2000. 74

[70] NIST Special Publication 800-56A. **Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography**. http://csrc.nist.gov/publications/nistpubs/800-56A/SP800-56A_Revision1_Mar08-2007.pdf, 2007. 74

Declaration

I herewith declare that I have produced this paper without the prohibited assistance of third parties and without making use of aids other than those specified. Notions taken over directly or indirectly from other sources have been identified as such. Some results in this Thesis are present in the following papers.

- F. Petagna, A. Castiglione, G. Cattaneo, G. De Maio, SECR3T: Secure End-to-End Communication over 3G Telecommunication Networks, Innovative, Mobile and Internet Services in Ubiquitous Computing (IMIS 2011), International Conference on, June 30th-July 2nd, 2011 Korean Bible University (KBU), Seoul, Korea (to appear).
- F. Petagna, A. De Santis, A. Castiglione, G. Cattaneo, M. Cembalo, U. Ferraro Petrillo, An Extensible Framework for Efficient Secure SMS, Complex, Intelligent and Software Intensive Systems (CISIS 2010), International Conference on, February 15-18, 2010, Krakow (PL).
- F. Petagna, G. Cattaneo, L. Catuogno, An Implementation of Interoperable Security Features for a Smart Card enabled VoIP Softphone, Proceedings of Workshop on ICT at 2008 International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2008), September 25-27, 2008, Split-Dubrovnik (HR).
- F. Petagna, G. Cattaneo, L. Catuogno, G. Di Matteo, L. Romano, iToken: a Wireless Smart Card Reader which Provides Handhelds with Desk Top Equivalent Security. The 2007 International Workshop on Secure and Multimodal Pervasive Environments (SMPE'07), September 17, 2007, Nice (FR).

- F. Petagna, A. De Santis, G. Cattaneo, A. Castiglione, U. Ferraro Petrillo, SPEECH - Secure Personal End-to-End Communication with Handheld. Proceedings of The Independent European ICT Security Conference and Exhibition (ISSE 2006), October 10-12, 2006, Rome (IT).

Fisciano (SA), Italy