MOBILE COMMERCE OVER GSM:

A BANKING PERSPECTIVE ON SECURITY

by

Pieter Ben van der Merwe

Submitted in partial fulfilment of the requirements for the degree

Master of Science (Electronics)

in the

Faculty of Engineering

UNIVERSITY OF PRETORIA

October 2003

Summary

Mobile Commerce over GSM: A Banking Perspective on Security

By

P.B. van der Merwe and W.T. Penzhorn

Department of Electrical, Electronic and Computer Engineering

University of Pretoria

MSc with specialisation in Electronics

Indexing Terms: GSM, GSM Security, Mobile Commerce, Mobile Commerce Security, m-Commerce, Cryptography, Wireless Application Protocol, WAP, Wireless Internet Gateway, WIG, SIM Application Toolkit, STK.

GSM has changed the face of communication and information exchange, much as the Internet did. With the advances made in the mobile technology arena, new opportunities are created. Mobile Commerce (m-Commerce) is one such opportunity. Each new advance in technology brings with it associated risks. This dissertation focuses on the risks involved with m-Commerce for the banking industry.

This dissertation provides a detailed overview of basic services that any m-Commerce application should provide to the banking industry. These principles provide the foundation for securing any financial transaction over untrusted networks. Several mechanisms to provide these services are also discussed. Examples of such mechanisms include hash functions, Message Authentication Codes and Digital Signatures.

The security of GSM networks has come under attack in the past. This is largely due to the fact that the GSM consortium opted to develop their security technologies in secret, rather than in the public domain. This dissertation aims to evaluate the security offered by GSM and assess potential attacks in order to further understand risks associated with m-Commerce applications over GSM.

In recent years there have been significant additions to the GSM enabling technology family. The arrival of the SIM Application Toolkit and the Wireless Application Protocol promised to again change the face of commerce. Although market acceptance of these technologies proved to be initially slow, usage is set to increase exponentially within the next couple of years. A detailed analysis of these enabling technologies is presented in the dissertation. Possible attacks on these technologies are discussed in the latter part or this document.

Based on the findings of the research, some changes to either the application architectures or the processing of the data have been suggested in order to enhance the security offered by these services. It is not the intent of this dissertation to redesign these applications, but to rather leverage off the current technologies in order to enable secure m-Commerce over these channels.

Samevatting

Mobiele Handel oor GSM: Sekuriteit vanuit 'n Bank se oogpunt

deur

P.B. van der Merwe en W.T. Penzhorn

Departament Elektriese, Elektroniese en Rekenaar Ingenieurswese

Universitiet van Pretoria

MSc met spesialiseering in Elektronika

Indekseringsterme: GSM, GSM sekerheid, Mobiele Handel, Mobiele Handel sekerheid, m-Handel, Kriptografie, WIG, "SIM Toolkit".

GSM het die gesig van komminukasie en die uitruil van inligting verander, soos die Internet dit tevore gedoen het. Met die vooruitgang in tegnologie kom nuwe geleenthede na vore, soos byvoorbeeld Mobiele handel. (m-Handel). Elke vooruitgang in tegnologie bring egter risiko's daarmee saam. Hierdie verhandeling sal fokus op die risiko's wat m-Handel na die finansiële sektor toe bring.

Die verhandeling verskaf 'n breedvoerige oorsig oor die basiese beginsels wat enige m-Handel applikasie moet verskaf aan die finansiële sektor. Hierdie basiese beginsels is die grondslag van enige finansiële transaksie oor onbetroubare netwerke.   Sekere meganismes wat die dienste kan verskaf, word ook bespreek. Die meganismes sluit in: hutsfunksies, Boodskap Stawings Kodes and Digitale Handtekeninge.

Die sekuriteit van GSM netwerke het in die verlede onder skoot gekom. Dit is grootliks as gevolg van die feit dat die GSM-groep verkies het om hulle sekuriteits-tegnologie in afsondering te ontwikkel, eerder as om die publiek te betrek. Hierdie verhandeling het ten doel om die sekuriteit wat GSM bied te ondersoek en potensiële aanvalle te evalueer, ten einde die risiko's verbonde aan m-Handel applikasies te kan meet.

In die afgelope paar jaar het GSM noemenswaardige toevoegings gekry tot sy bemagtigings-tegnologie "familie". Die aankoms van die "SIM Toolkit" beloof om weer die gesig van handel te verander. ’n Breedvoerige studie van die bemagtigingstegnologië word in die verhandeling onderneem. Sommige aanvalle op die tegnologië word later ondersoek.

Gegrond op die bevindings van die navorsing word sommige veranderings aan die argitektuur, of die hantering van data van die applikasies voorgestel, ten einde die sekuriteit van die dienste te verhoog. Dit is nie die bedoeling van die verhandeling om hierdie applikasies te herontwerp nie, maar eerder om dit wat deur hulle gebied word te benut en te bevorder om sodoende sekuur m-Handel oor die kanale te bedryf.

Acknowledgements

I would like to make use of this opportunity to thank the following individuals:

My friends and family, too many to mention, for their continued support throughout this project.

A special word of thanks goes to my father, mother and sister. Throughout my life you have been there to support me. Thank you for the amazing opportunities you gave me. God's light will shine on thee.

To a very special someone, for her continued support throughout the toughest parts of this project. Thanks RLK. You made it special.

Johan du Toit for his unceasing encouragement and for going out of his way to accommodate my requirements.

David Goosen for spending late nights with me to help me program the simulator.

Linda Rautenbach for helping me spell better.

The management of Nedcor Bank Ltd for the time afforded me in order to complete this dissertation.

It is not necessary to hope in order to undertake, nor to succeed in order to persevere.

Charles the Bold (1433 – 1477)

LIST of TABLES

LIST of FIGURES

CONTENTS

CHAPTER 1: INTRODUCTION

1.1 BACKGROUND

In order for us to discuss Mobile-Commerce, we first need to look at some definitions:

Definition 1.1: e-Commerce:

Electronic commerce (e-Commerce) can be defined as the mutual exchange of perceived or monetary value by electronic means over openly accessible networks. This basically means communication over the Internet for some or all of the transaction processes [1].

Definition 1.2: m-Commerce:

Mobile commerce (m-Commerce) can be defined as any transaction with added value for the user, which is carried out by means of mobile/wireless devices or infrastructure [1].

Definition 1.3: m-Finance:

Mobile financial services are a subset of m-Commerce that offers a range of banking, share dealing and insurance services [1].

1.1.1 The growth of e-Commerce/ m-Commerce

Internet use has grown on the strength of PC networks. The Yankee group predicts that the installed base of PCs will reach 500 million by 2003. This base is essential in driving the continued growth of e-Commerce and communication applications. Due to the fact that these devices have greater power, storage and price-performance ratios, more powerful and sophisticated applications are likely to emerge for desktop computing and the Internet [2].

Impressive as they may be, these systems still have the limitation that users have to sit in front of them and dial into, or physically connect to a network in order to launch their Internet applications. The aggregate PC installation is substantial, but even more mobile communication devices are in use [2]. This scenario is depicted in Figure 1.1.

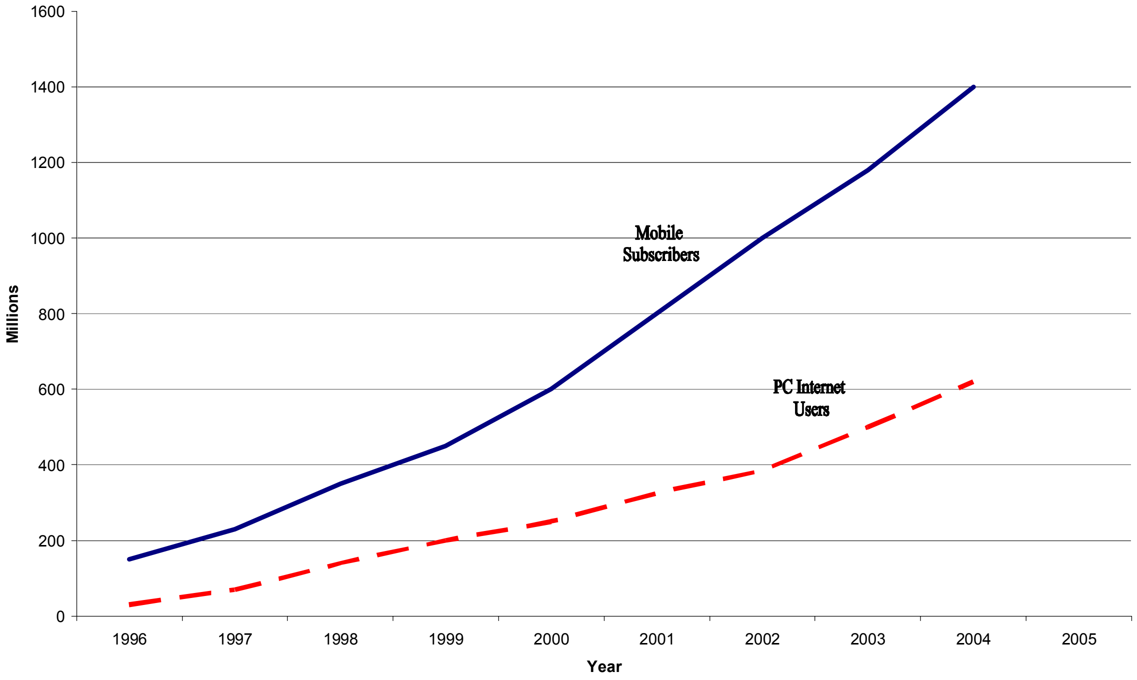


Figure 1.1: Projected installed base of PCs and mobile telephones. [2]

With mobile communications reaching the mass market, network operators are facing a decreasing ARPU (Average Revenue per User) [3]. Price erosion accelerates with every new network operator entering into a local market [4]. This is very evident in the number of specials and price reductions introduced by Vodacom and MTN in the South African market prior to the activation of Cell C. It is commonly accepted that mobile tariffs will come down to the same level of fixed line tariffs in the next 2 to 3 years [3] [4]. Network operators must therefore continuously implement new and improved services on their networks if they want to slow down, or reverse the rate of decreasing ARPU [4]. Mobile data and SMS services have not been very successful at doing this in the past. These services usually generate no more than 2-3% of a network operator's turnover [4].

The exponential growth of wireless and mobile networks has affected huge changes in mobile devices, middleware development, and standards for network implementation and user acceptance. It is estimated that more than 350 million mobile devices are in use worldwide and this number is expected to rise to 1 billion in the coming years [5]. Analysts also predict that by 2002 at least 70 percent of wireless subscribers worldwide will access data applications via their mobile phones. Further to this it is perceived that by the year 2004, 50 percent of Internet hits will originate from wireless devices in the USA [2]. Thus the continually increasing numbers of handheld terminals make mobile networks an ideal channel for offering personalised services to mobile users. This in turn drives the rapid development of e-Commerce conducted with portable devices [6].

With the coming of advanced and sophisticated services, mobile communications combined with e-Commerce propositions are heightening the attractiveness of m-Commerce. The key drivers of this are:

- Ubiquity: The 'anytime anywhere' advantage of m-Commerce [1]. Smart phones may fulfil the need for both real-time information and communication, independent of the user's physical location [4].
- Reachability: Using a mobile terminal, a user can be contacted anywhere anytime. Mobile handsets also provide users the ability to restrict their reachability to certain people [4].
- Personalisation: Handsets are effective personal accessories that are capable of holding data and enabling access to information and services tailored to the needs of each individual [1].
- Localisation: Noting where the user is and providing information relating to that location adds a unique value to mobile services [1].
- Convenience: Mobile subscribers have become accustomed to their devices that store data and are always at hand. More advanced applications are driven by technology further enabling the mobile subscriber [4].
- Convergence: Technological applications can be deployed on the move. This is blurring the divide between mobile phones and PCs. Ever increasing sophistication and functionality sustains further handset development [1].

- Internet Access: Instant connectivity to the Internet from a mobile device is fast becoming a reality and will take off with the introduction of GPRS. This will mean faster access to the Internet than from a PC based application. It is suggested that mobile devices will become the preferred means of accessing information on the Internet [4].

However there are also several factors that may slow or constrain the progress of m-Commerce. These inhibitors include the following:

- Interoperability: Due to the range of handset functionalities and operating systems, there are inherent costs associated with delivering a range of services. This may deter some content providers from making the investment and carrying the overheads associated with such a service [1].
- Usability: The Internet provides rich content via the large screens and multimedia capabilities of PCs. The constraints imposed on the mobile handset might limit its appeal to users [1].
- Security: The public has serious concerns about the security of the Internet. This has been a major constraint to consumer e-Commerce. This negative perception may be transferred, or potentially magnified, to the mobile arena [1]. Mobile security technology is however emerging, like SSL in closed end-to-end systems. The SIM also provides for user authentication by leveraging of the Smart Card technology [4].

1.1.2 e-Commerce and m-Commerce in Africa

The Internet has been described as the great equaliser, as people can now get information about any conceivable topic at the click of a button. The limitation of this kind of information is that most of it runs over wired networks. In Africa, access to wired networks is not as readily available as in other parts of the world. Wireless networks could be the answer to delivering Internet information services to the masses in Africa.

Further to this, Africa has traditionally been an under-banked market, as access to banks is very limited. Combining the wireless space with m-Commerce can change many of these facts. Not only can one deliver content to the users over GSM, one could also give them access to financial services they never had access to before.

1.2 PROBLEM STATEMENT

With the emergence of m-Commerce applications came m-Finance applications. The Financial sector has always put a high priority on confidentiality of customer information as well as integrity of transmission data.

During investigations of m-Commerce applications and security offered to financial institutions by various vendors, it has come to the attention of the author that little interest is paid to securing these kinds of transactions. Vendors evangelise the use of GSM for new and exciting payment solutions and the security offered to the institution by using GSM as the carrier of these transactions.

It is widely accepted in the mobile community, and specifically by [7][8][9] that the following services should be available when providing m-Commerce opportunities to consumer. The list includes:

1. Confidentiality: Only the communicating parties can view the communicated data.
2. Integrity: Unauthorised parties cannot tamper with the traffic flow between communicating parties, i.e. they cannot alter the data without the knowledge of the communicating parties.
3. Non-Repudiation: Neither of the communicating parties can deny ever receiving or sending any of the communications between them.
4. Authentication: Each communicating party needs to be able to authenticate the other party in the communications ensuring they are who they say they are.
5. Authorisation: Ensuring a party performing the transactions is entitled to perform these actions.

Each of these elements is discussed in detail in Chapter 2.

In their article titled:” Enhancing Security of GSM” [10] states that although voice data is protected on the radio link between the mobile handset and the GSM Base Station, there is no protection offered during the transmission of the data through the fixed network of the mobile communications provider. This fact could lead to eavesdropping of the voice data, as well as tampering with Short Message Service (SMS) messages sent between communicating entities.

In [11] the author argues that although SMS messages are encrypted over the air link, it is highly accessible for attacks from the Network Provider side due to its store and forward technology. He also states that injection of malicious and false SMS messages is possible through poorly protected SMS gateways. This indicates that Authentication, Authorisation, Confidentiality and Integrity of the SMS communications are at risk.

On the contrary, [12] and [13] proposes the security requirements for use of the SIM Application Toolkit, and even shows implementations of some of the solutions. Further investigation has revealed that *not* all that was promised by GSM, [12] and [13] has yet been delivered.

Different network operators implement various specifications in differing ways due to the high cost involved in securing these networks. Some networks operators even have some proprietary applications running on their networks to try and limit the cost involved in applying the GSM standards. These factors imply that one cannot blindly trust the security offered by GSM and its accompanying applications. An alternative solution is required, and this thesis investigates the possibilities and proposes a secure solution that is cost effective and applicable to the African environment.

1.3 ASSUMPTIONS

Some assumptions have been made in the writing of this dissertation in order to limit the scope of work. These assumptions are listed below in no specific order.

Assumptions made in writing this dissertation include:

1. Any information or transaction data flowing across an IP network is secure. As there are numerous well-known mechanisms for securing of data traversing IP networks, it is felt that they are outside of the scope of this dissertation. Some examples include IP Security (IPSec) and the Secure Socket Layer (SSL).
2. The backbone networks of financial institutions are secure. Although this might not always be the case, for the sake of highlighting the relevant security vulnerabilities in m-Commerce this assumption had to be made.
3. Any party that has a communication channel to a financial institution has to pass their network traffic through the firewall of that institution. In some instances firewalls are shown in diagrams. In these diagrams the author felt the need to include them for the sake of completeness.
4. All SIM cards have the Triple DES algorithm programmed into them, and each SIM card possesses some derived unique keys that can be used for encryption of other data apart from voice.
5. Wherever a wireless solution is proposed, it is assumed that the intended user of this solution has the correct equipment to perform the functions defined by the solution. For instance, when a solution based WIG is discussed, it is assumed that the user has a phase 2+ GSM handset fitted with a 32k SIM Card.

1.4 OBJECTIVES

The objectives of this dissertation are:

1. Understanding the basic principles of Information Security and how to obtain them.
2. Analysing the security offered by the GSM network, of both over-the-air and the fixed network, with special emphasis on message integrity.
3. Analysing the security offered by Wireless Internet Gateway (WIG) and Wireless Application Protocol (WAP) applications.
4. Applying the basic principles of Information Security to WIG and WAP architecture in order to propose enhancement to the security of these solutions.
5. Design and build a secure m-Commerce simulator over WIG and prove the success of the suggested changes to the architecture to enhance security.
6. Evaluate the proposed solution against acceptable international security standards.

1.5 SCOPE OF WORK

This dissertation only focuses on the security of current GSM applications. These include:

1. GSM security in both over-the-air and the fixed network.
2. Wireless Internet Gateway security over GSM

This dissertation specifically only covers applications running over GSM, as the author is of the opinion it will still take at least a year for the network operators in Southern Africa to provide a commercially sustainable GPRS network.

A further reason is the specific market this technology can be aimed at. Even today in South Africa there are a lot of potential banking customers in rural areas. These areas are not easily accessible to the traditional banking fraternity.

These potential customers do however have access to GSM technology and handsets. These handsets are usually not capable of accommodating advanced technologies like GPRS etc.

This dissertation does not evaluate future technologies like GPRS, EDGE, or UMTS, as the author is of the opinion that these are merely different bearer channels and that the arguments described in this dissertation will still hold.

1.6 METHODOLOGY

The research was conducted by the following means:

1. Bibliographical research from various literature sources.
2. Information searches on the Internet and participation in Internet based discussion forums on the topic of GSM security and its associated commerce enabling technologies.
3. Practical work experience gained in the financial sector.
4. Participation in technical m-Commerce project groups in the banking industry, where problems in current GSM systems was identified and analysed.
5. Investigation of the available cryptographic applications provided by the mobile technologies were studied, in order to assess they availability of technologies like the Data Encryption Standard, digital signatures etc.
6. Subsequent investigations then led to the choice of cryptographic solution to the identified problems. The cryptographic solution chosen should be able to function on the highly restrained mobile processing environment.
7. Once all the pieces of the puzzle were in place, the actual workable solution was proposed which took all the previous research into consideration.
8. Once a workable solution was identified, the design and production of an m-Commerce simulator over WIG commenced. This was done in order to prove that the proposed solution was in fact a workable solution.

9. Experimental results were then verified in order to ascertain that the integrity of the transactional messages was still intact.
10. Finally the proposed solution was evaluated against acceptable International standards of good practice.

This dissertation is the culmination of practical work experience and theoretical research over a period of more than two years.

1.7 OUTLINE

The dissertation follows a logical flow. It is the objective of this dissertation to take the reader through the same steps the author took in order to understand m-Commerce security. To this end the dissertation covers the following topics:

Chapter 2 looks at the basic requirements that information security should achieve. To this end, some applications that can provide these services are discussed. These include hash functions, Message Authentication Code algorithms, and Digital Signatures.

In Chapter 3, we discuss the security provided by the GSM network. We specifically look at the authentication of subscribers and at user- and signalling data integrity and confidentiality.

Chapter 4 covers numerous attacks on, and vulnerabilities of, the GSM network.

Chapter 5 looks at the WIG application and its enabling technologies in great detail. In this chapter possible attack scenarios are described and solutions to these attacks are offered.

In Chapter 6 a solution is proposed in order to facilitate secure m-Commerce transactions.

In Chapter 7, an analysis of the solution proposed in Chapter 6 takes place.

Chapter 8 discusses the experimental setup and design used to evaluate the proposed solution.

In Chapter 9 the author ties all the strings together in the Conclusion.

CHAPTER 2: MESSAGE INTEGRITY

2.1 INTRODUCTION

With the introduction of the computer and the advent of computer networks, the need for protecting information became very evident. Company and customer data traversed the open networks and this information fell into the wrong hands or got altered without the knowledge of the author of the message. Automated tools were required for protecting the sensitive data flowing over these networks. Cryptography came as a clear answer to all the concerns.

In this chapter we will look at transaction/ message security over open networks regardless of the transmission media used. Transaction security is critical in any commerce application but even more so in the e-Commerce and m-Commerce domains, as the electronic messages are transmitted over open, unsecured networks.

2.2 ATTACKS ON ELECTRONIC MESSAGES

An attacker might want to gain access to an electronic message for numerous reasons. Gaining unauthorised access to information in order to violate someone's privacy, impersonating another user in order to shift the responsibility or originate a fraudulent activity are some of the reasons an attacker might want to access the information [14].

There are four general categories of attacks on a transmitted message apart from normal transaction flow: [14]

- Interruption.
- Interception.
- Modification.
- Fabrication.

Each one of these will now be discussed in detail.

2.2.1 Normal message flow

In general there is a flow of information from a source to a destination. In a normal message flow, the information passes unhindered from the source to the destination as shown in Figure 4.1.

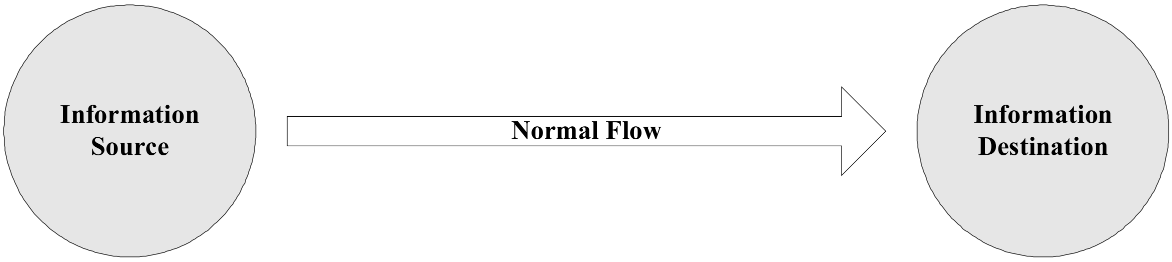


Figure 2.1: Normal message flow

2.2.2 Interruption

Interruption is the action of preventing a message from reaching its intended recipient. It can also occur when an asset of the system is destroyed or becomes unavailable or unusable. This is an attack on availability. Some examples of these kinds of attack include: [14][15]

- Destruction of a piece of hardware.
- The intentional cutting of a communication line.
- Disabling of the file management system.
- Denial of service attack.
- Figure 4.2 illustrates this graphically.

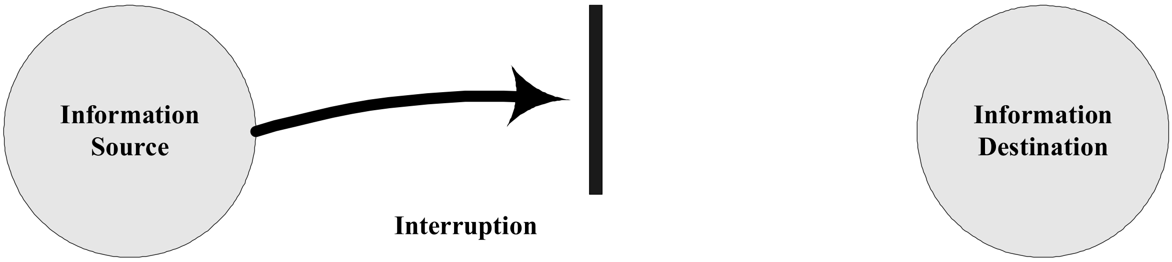


Figure 2.2: Interruption of a message. [14]

2.2.3 Interception

Interception is where an unauthorised party gains access to information. This is an attack on confidentiality [14]. The unauthorised party might be a person, program or a computing system [15]. A loss due to this kind of attack might be noticed quickly, but a silent interceptor might leave no traces by which the interception can be detected. Examples of these kinds of attacks include: [14][15]

- Wiretapping to capture data in a network.
- Illicit copying of files or programs.

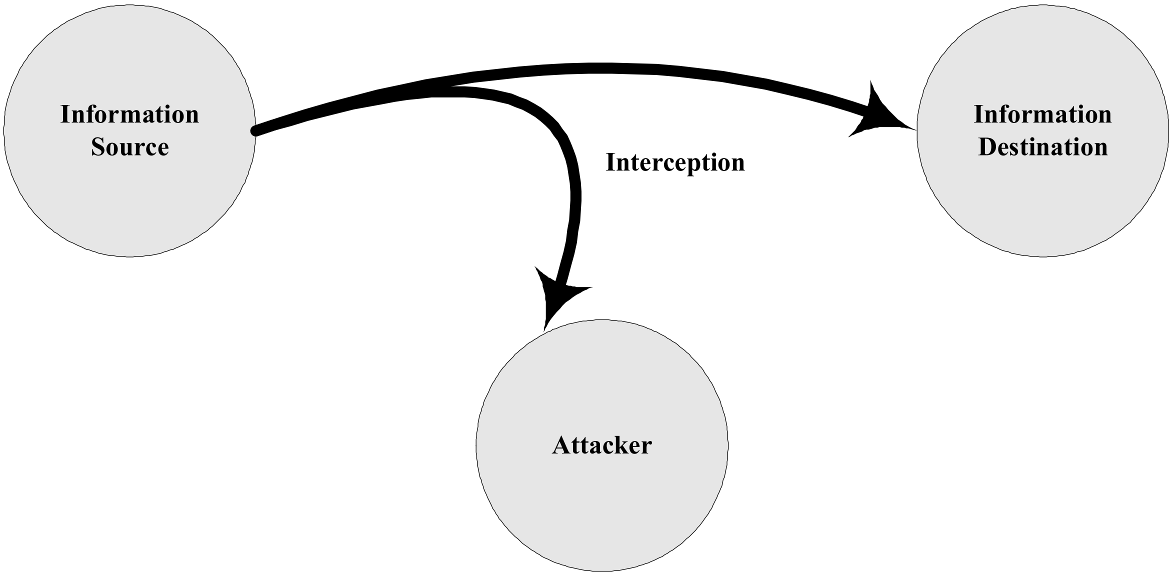


Figure 2.3: Interception of a message. [14]

2.2.4 Modification

Modification is where an unauthorised party not only gains access to an asset, but tampers with it. This is an attack on the integrity of the message [15]. Examples include: [14][15]

- Changing of values in a database for personal gain.
- Altering a program so that it performs an additional computation.
- Modifying the content of a message transmitted on a network.

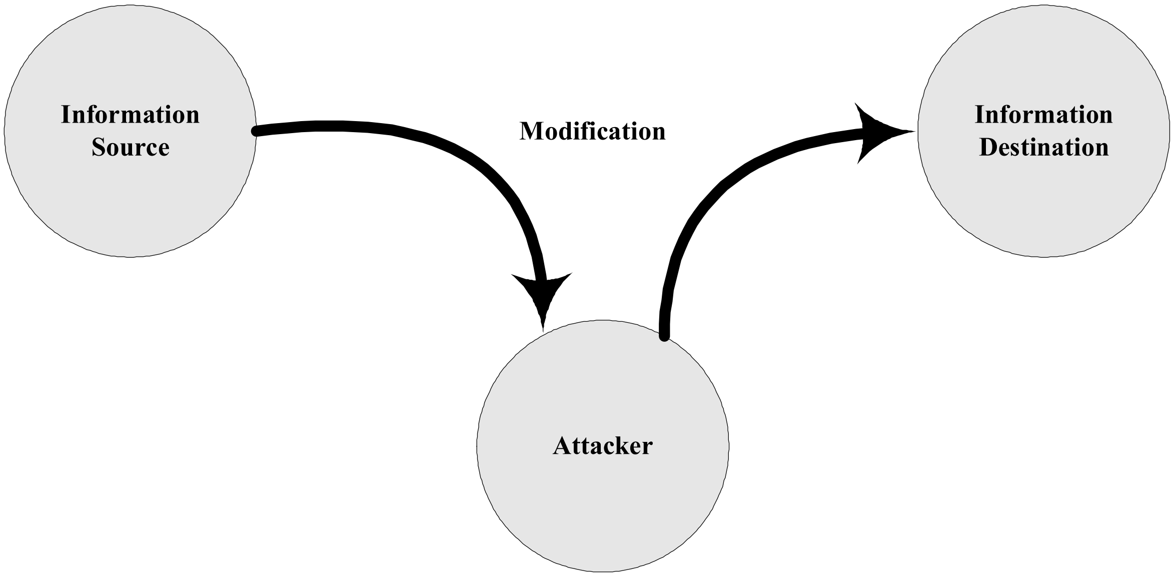


Figure 2.4: Modification of a message. [14]

2.2.5 Fabrication

Fabrication occurs when an unauthorised party inserts counterfeit objects into the computing system [15]. This is an attack on the authenticity of the message [14]. These insertions can sometimes be detected as forgeries, but if skilfully done, they are virtually indistinguishable from the real thing. Examples include: [14][15]

- Insertion of spurious information into the network communication system.
- Adding additional records to an existing file or database.

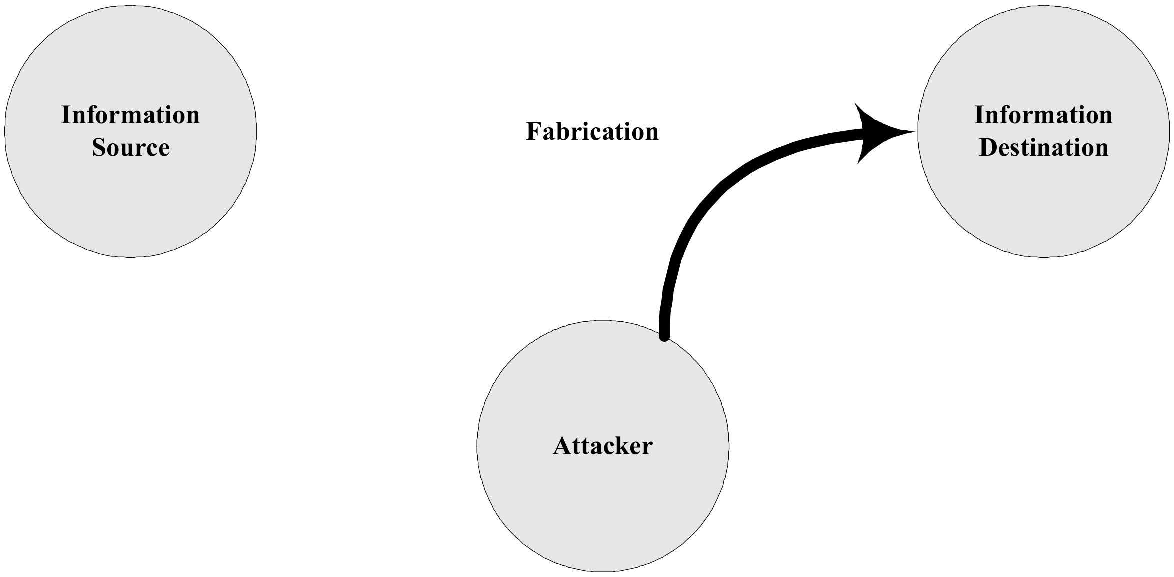


Figure 2.5: Fabrication of a message. [14]

2.3 OBJECTIVES OF SECURITY IN MESSAGE TRANSMISSION

In knowing and understanding the attacks on messages we can look at the goals of securing transmissions. Regardless of who is involved, all parties to a transmission must have confidence that certain objectives associated to message transmission have been met. One such means of ensuring this is by means of cryptography.

Cryptography is the study of mathematical techniques related to aspects of information security like confidentiality, message integrity, entity authentication etc. [16]. Cryptography provides the means to ensure that the objectives of communicating parties are met. These objectives are listed in Table 4.1 and briefly discussed.

Confidentiality	Keeping information secret from all but those who are authorised to see it.
Message integrity	Ensuring the transmitted message has not been altered by unauthorised or unknown means.
Authentication	Corroboration of the identity of an entity, like a person, a computer terminal etc. and corroboration of the origin of the message.
Non-repudiation	Preventing the denial of some previous commitments or actions by the communicating parties.
Availability	Availability provides functionality to ensure that resources or information are accessible and usable upon demand by authorised users.
Authorisation	Authorisation provides functionality to determine whether users or applications are permitted to use computer resources.

Table 2-1: Information security objectives [14][15][16]

2.3.1 Confidentiality

Confidentiality protects against the threat of revealing information to a user not authorised to have that information. Authentication and access control provide some level of confidentiality by allowing only those users or processes that are identified, authenticated, and authorised to gain access to information. However, other measures provide additional levels of protection against disclosure by concealing the information.

Confidentiality is defined as the property that ensures that information is not made available or disclosed to unauthorised users. Confidentiality mechanisms are intended to prevent information dissemination to users who are not authorised to receive it.

A confidentiality mechanism may prevent access to the information (physical protection or access control) or may conceal or alter the information (encipherment or data padding) to all but those who have privileges.

There may not be a requirement to provide confidentiality on a complete set of data. Selective field confidentiality provides confidentiality for selected fields within the data. For example, some text in the data may need to be clear text for processing or performance requirements.

In some environments, sensitive information may be extracted by monitoring communications. The sensitive information may be inferred by observing external characteristics such as a sudden increase in electronic traffic between two companies; however, the actual messages transmitted may not be compromised. This type of analysis may be thwarted by use of data padding mechanisms in conjunction with encipherment. Data padding mechanism may include adding blocks of data to a message as well as sending white noise or meaningless messages [17].

2.3.2 Message Integrity

Integrity protects against the threat of corruption or modification of information (either accidentally or intentionally). Authentication and access control provide some level of integrity by allowing only those users who are identified, authenticated, and authorise to gain access to the information. However, other approaches provide additional levels of integrity by detecting and, possible, correcting corruption or modification.

Integrity is defined as the property that information has not been altered or destroyed in an unauthorised manner. Integrity addresses the threat that the value or existence of information might be modified. If an integrity mechanism cannot prevent alterations of the information, it must be able to detect them.

There are two general categories of integrity protection:

- Preventing access to the information through such mechanisms as secure channels (for example access control) and routing control (for transient data or stored data).
- Detecting corruption or unauthorised modification of the information for example, through the use of cryptographic seals or digital signature (transient data or stored data).

Protecting information is often a function of the physical media in use. Some media are inherently difficult to protect; cellular network/ communications is an example. In those cases, detection of the loss of integrity is important.

Some examples of integrity mechanisms are cryptographic seals, digital signature, and error detection. These mechanisms usually rely on computing a value based on the content of the data and encrypting the result. A modification of the data is detected by re-computing the value and comparing it to the original. A mismatch indicates that the data has been altered. Replication of data in multiple locations may also be used for comparison to provide an integrity check [17].

2.3.3 Authentication

The authentication service is concerned with ensuring that communication is authentic [14]. This applies to both entities and the message itself [16]. The two communicating parties should identify each other. This ensures they are communicating with the correct entity and not someone pretending to be that entity. Information delivered over the channel should be authenticated as to origin, date of origin, data content, time sent, etc. [16]. For this reason this aspect of cryptography is subdivided into two major classes: entity authentication and origin authentication. By the nature of its definition data origin authentication implies data integrity. From this, one can deduce that, if a message is modified during transmission, the origin has changed [16].

2.3.4 Non-Repudiation

Non-repudiation is different from the other security functions in that it provides protection from legitimate users rather than from unknown attackers. This security function depends on authenticated, authorised identities and integrity-protected data to provide evidence to resolve disagreements about participation in an exchange or communication.

Non-repudiation provides protection against the denial by one of the entities involved in a communication of having participated in all or part of the exchange. This mechanism can also provide a means of rejecting invalid/ unauthorised communications. Non-repudiation can provide proof of delivery of data, as well as proof of origin, and proof of integrity. Non-repudiation with proof of origin protects against an originator denying originating the data or falsely claiming that the data was modified. Non-repudiation with proof of delivery protects against a recipient denying receiving the data or falsely claiming that the data received has been modified and is not what was originally sent.

Non-repudiation functionality is often specific to the individual application. For example, transaction security services associated with electronic data interchange are quite different from those used for electronic funds transfer and point of service applications. Generally, the industries concerned define formal or de facto standards appropriate to their needs [17].

2.3.5 Availability

Availability provides functionality to ensure that resources or information are accessible and usable upon demand by authorised users. Availability of computer systems is a broad area that may be addressed by a wide variety of methods ranging from well-defined backup policies to fault tolerant computer architectures. Availability functionality can be provided by using combinations of other security functions, for example by preventing or detecting security breaches that would result in diminished availability.

Authentication and access control provide aspects of availability by allowing only those users who are identified, authenticated, and authorised to gain access to the information.

Availability can be provided indirectly by using logging and audit functionality provided within the security administration and management function by progressively improving the access control. The use of available security functions, together with the established security policies, processes, procedures and standards, provide enhanced availability to the computer system, by

- other security services preventing unauthorised access resulting in a denial of service, or
- by other security services protecting data required for recovery
- by security processes and infrastructure providing for business continuity and disaster recovery.

The primary responsibility of availability would be to:

- guard against denial of service attacks to authorised users,
- assist in eliminating single points of failure in the security chain (for example fall back authentication services and fall back authorisations services in cases of accidental or malicious interruption of services), and
- ensuring correct interpretation of access rules; that is, access to resources is not denied to legitimate users with legitimate access rights [17].

2.3.6 Authorisation (Access Control)

Access control provides functionality to determine whether users or applications are permitted to use computer resources. Permissions or authorisation of users or processes are defined according to the policies of the business. Access permissions should be defined by making use of "role profiles". Role profiles map to the user's business role and access privileges within that role; it therefore makes more sense to the users/ business. These roles do not only include access permissions, but also user related restrictions, such as transfer limits.

The authorisation policy decisions of who is allowed use of resources and the authentication of a user must occur before access to resources may be allowed. The purpose of access control is to prevent unauthorised access to any resources.

Unauthorised access may include unauthorised:

- use of a system or resource;
- disclosure of information;
- modification of the system or information;
- destruction of information or programs;
- issuance of system commands;
- or in general, any use of a resource in an unauthorised manner.

Access control capabilities directly support confidentiality, integrity, and availability by ensuring that the computing resources are used by only the users and processes with the correct permissions [17].

2.4 ENSURING MESSAGE INTEGRITY DURING TRANSMISSIONS

In any financial transaction, over any untrusted network, it is important to ensure that the integrity of the message is not compromised. To this end we will look at ways to ensure the integrity of messages during transmission over said unsecured networks. It is important to note here that the transmission media is of no concern to the applications listed below.

Figure 2.6 below illustrates the procedure for verifying the integrity of transmitted messages that were protected by use of some cryptographic technique, the result of which is denoted by the term Authentication Code (AC).

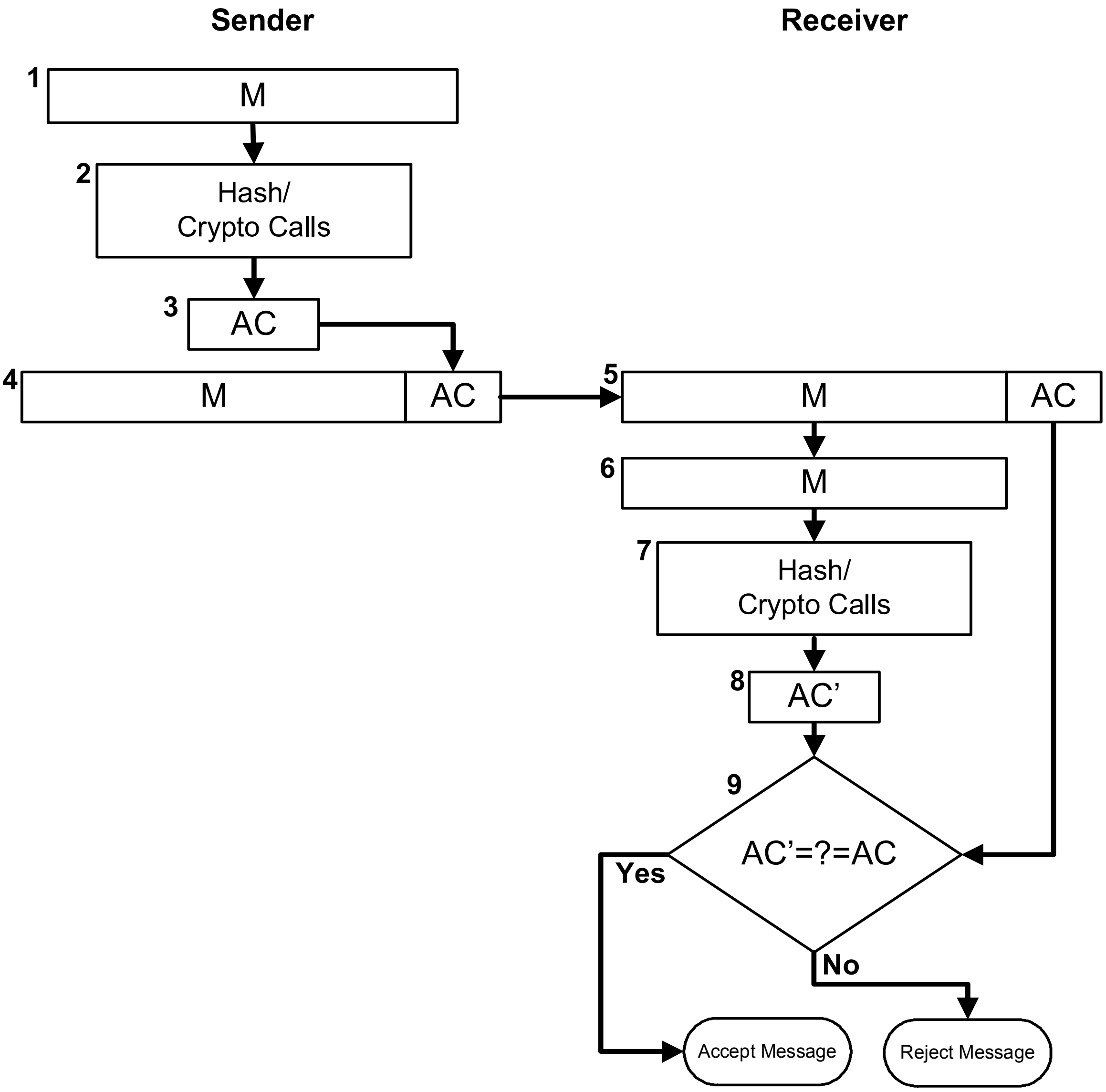


Figure 2.6: Message Integrity Verification Procedure

2.4.1 Hash functions

A hash function can be defined as a function that compresses an input string of arbitrary length to an output string of fixed length. There are numerous implementations of such hash functions like the MD4, MD5, HAVAL and SHA-1 algorithms. The current international standard for hash functions is the SHA-1 algorithm and as such when we refer to a hash function in the text, we are referring to the SHA-1 algorithm.

Figure 2.7 graphically illustrates how the SHA-1 algorithm gets applied to a message of arbitrary length.

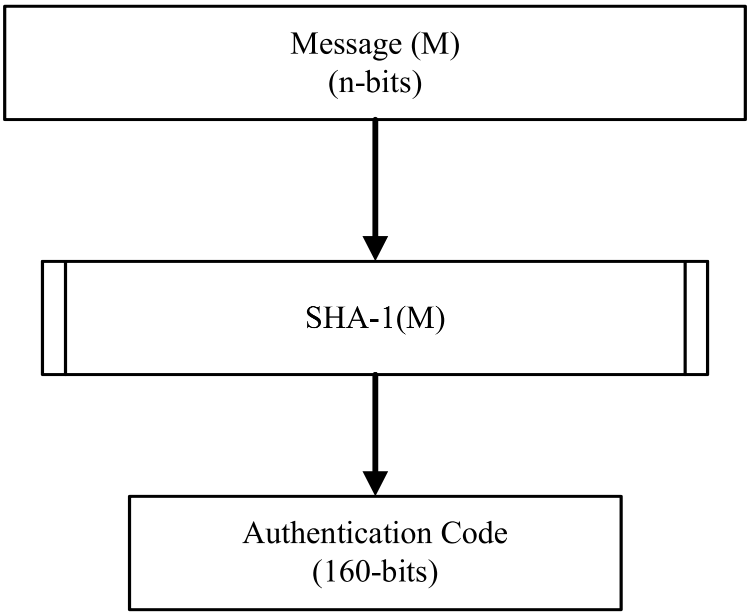


Figure 2.7: The use of SHA-1 hash algorithm

The output from the function then gets appended to the message

M || SHA-1(M)

The entire string [M || SHA-1(M)] then gets transmitted to the intended recipient. The recipient then verifies the received hash value with the received message as input, as illustrated in Figure 2.7.

Although a lot has been written about the security of SHA-1, it is not secure when used in this way. The reason should be quite obvious to the reader. Both the message and hash value are transmitted in the clear. If an attacker was to intercept the transmission of say, a payment instruction, he could easily alter the instruction for financial gain. All the attacker needs to know is the structure of the message, and the specifics of the hash algorithm used. He could then alter the message, compute a new hash value, and transmit the altered message and the new hash value on the network. To a payments engine, the message would seem legitimate and it would surely process the transaction to the benefit of the attacker. A more secure solution is needed. Is a keyed hash the answer?

2.4.2 Keyed Hash

A keyed hash is defined as a hash function that takes as an input a message of arbitrary length and a key of fixed length and computes the corresponding output string. In this instance we will make use of the SHA-1 hash algorithm and a secret 160-bit key shared between the sender and receiver. Figure 2.8 illustrates this procedure graphically.

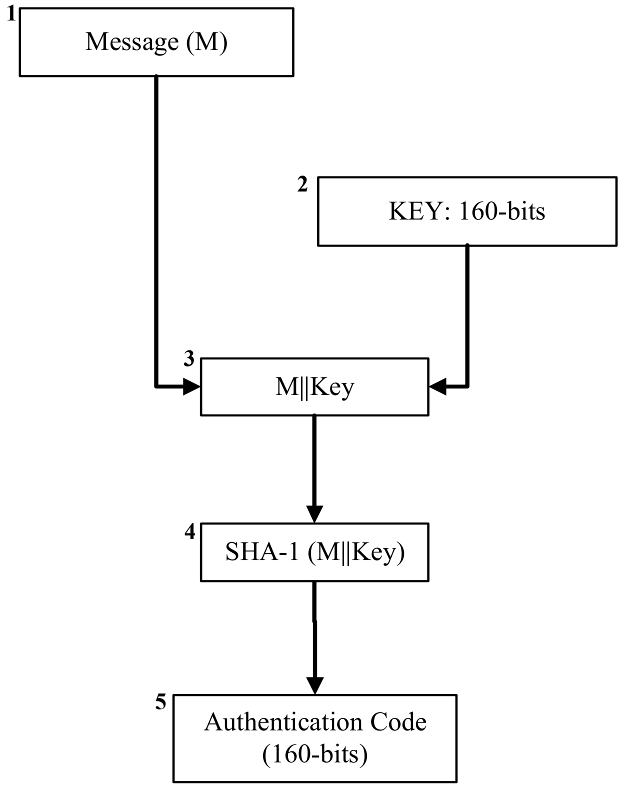


Figure 2.8: Keyed HASH

Although this method of using a hash function makes it more difficult for the attacker to construct a false message during transmission, it is still not cryptographically secure. The key appended to the message and the corresponding authentication code is sent in the clear. An attacker can easily extract the key, construct a new false message, and re-compute the corresponding authentication code. More security is needed to ensure the integrity of data during transmission for a financial transaction. HMAC is a new technology that we explain in the next section.

2.4.3 HMAC

HMAC is used in combination with a cryptographic hash function specified in the Federal Information Processing Standard (FIPS) document [Keyed-Hash Message Authentication Code (HMAC)]. HMAC uses a secret key for the calculation and verification of the MACs.

The main goals behind the HMAC construction are:

- To use available hash functions without modifications; in particular, hash functions that perform well in software, and for which code is freely and widely available,
- To preserve the original performance of the hash function without incurring a significant degradation,
- To use and handle keys in a simple way,
- To have a well-understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions on the underlying hash function, and
- To allow for easy replaceability of the underlying hash function in the event that faster or more secure hash functions are later available.

To determine the HMAC the following steps are to be followed:

1. If the length of K = B, set K_0 = K. Go to step 2. If the length of K > B, hash K to obtain an L byte string: K = H (K). If the length of K < B, append zeros to the end of K to create a B-byte string K_0 (e.g., if K is 20 bytes in length and B = 64, then K will be appended with 44 zero bytes 0x00).

2. Exclusive-Or K_0 with ipad to produce a B-byte string: **$K_0 \oplus$ ipad**.

3. Append the stream of data 'text' to the string resulting from step 4: **($K_0 \oplus$ ipad) || text.**

4. Apply H to the stream generated in step 5: **H (($K_0 \oplus$ ipad) || text).**

5. Exclusive-Or K_0 with opad: **$K_0 \oplus$ opad.**

6. Append the result from step 6 to step 7: **($K_0 \oplus$ opad) || H (($K_0 \oplus$ ipad) || text).**

7. Apply H to the result from step 8: **H (($K_0 \oplus$ opad) || H (($K_0 \oplus$ ipad) || text)).**

8. Select the leftmost t bytes of the result of step 9 as the MAC.

Where:

B Block size (in bytes) of the input to the FIPS-approved hash function; e.g., for SHA-1, $B = 64$.

H FIPS-approved hash function, e.g., FIPS 180-1, *Secure Hash Algorithm-1 (SHA-1)*.

ipad Inner pad; the byte x’36’ repeated *B* times.

K Secret key shared between the originator and the intended receiver(s).

K_0 The key *K* with zeros appended to form a *B* byte key.

L Block size (in bytes) of the output of the FIPS-approved hash function; for SHA-1, $L = 20$.

opad Outer pad; the byte x’5c’ repeated *B* times.

t The number of bytes of MAC.

text The data on which the HMAC is calculated; the length of the data is *n* bits, in this case 160 for SHA-1

x’*N*’ Hexadecimal notation, where each ‘*N*’ represents 4 binary bits.

|| Concatenation

$\oplus$ Exclusive-Or operation.

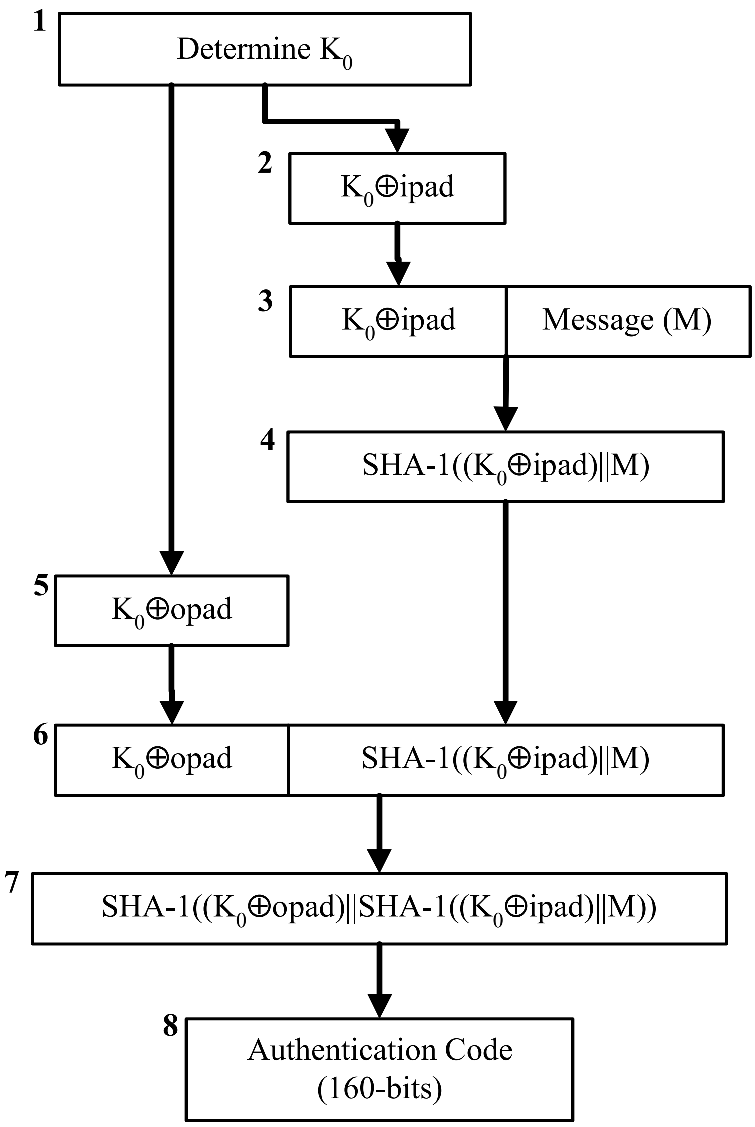


Figure 2.9: HMAC implementation

Although an HMAC is more secure than a hash or a keyed hash, it still does not satisfy our needs. The HMAC might seem complex, and certainly is not easy to decode, it is still not cryptographically strong. SHA-1 does not provide confidentiality, and the exclusive-OR function is certainly not difficult to replicate. Thus attackers can reproduce an HMAC given the clear text message and the authentication code, as the keys can be extracted with relative ease. We need a mechanism that can ensure that an attacker cannot derive the authentication code from the transmission of the message and the appended authentication code. The next section will look at such a mechanism.

2.4.4 Message Authentication Code (MAC)

In this section we will look a SHA-1 Triple DES MAC in CBC-mode. DES as we know, stands for the Data Encryption Standard. The CBC-mode points to the mode of operation in which we would like to implement the triple DES function, Cyclic Block Chaining in this instance.

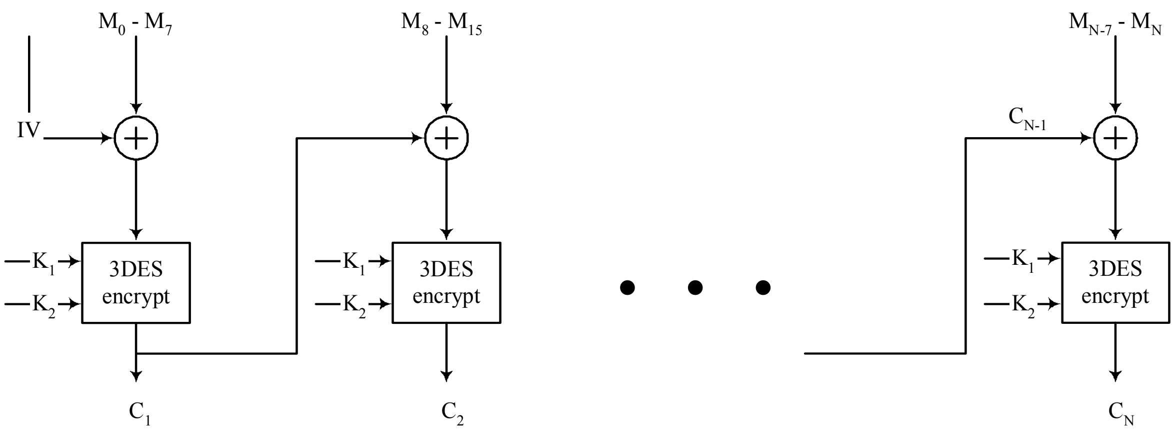


Figure 2.10: The CBC-mode of operation of 3DES [14]

In computing the authentication code in this method we follow the following simple steps:

1. Compute the SHA-1 hash value for the message being transmitted.
2. Encrypt the SHA-1 hash value using Triple DES in CBC-mode using two secret keys.
3. Take the last 8-byte block (C_N) output from the Triple DES function as the MAC.
4. Append the MAC to the message and transmit to the recipient.

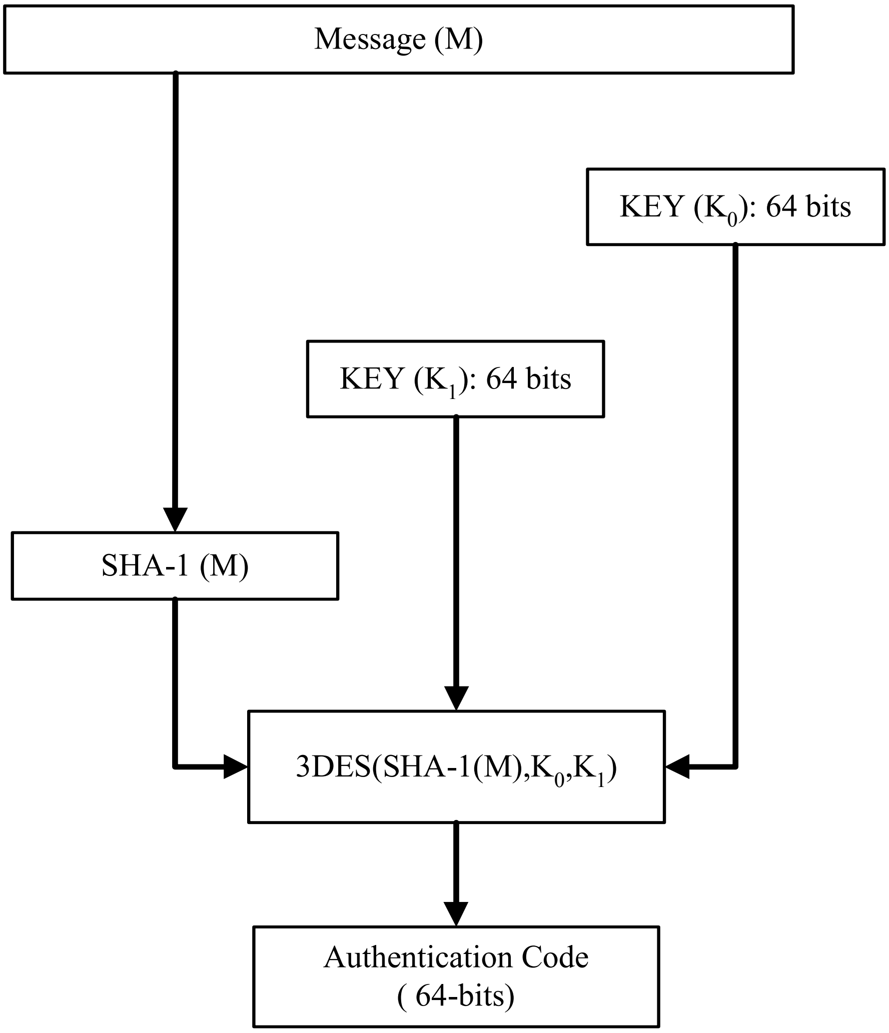


Figure 2.11: SHA-1 3DES MAC

The authentication code computed in this fashion can be regarded as cryptographically secure. Even when an attacker intercepts the clear text message and the MAC, he cannot extract the keys from the MAC. This implies that the attacker cannot alter the message and re-compute the MAC as he has no access to the keys needed. Another reason for computing the MAC in this fashion is that industrial Hardware Security Modules (HSM) can be used to perform this function.

In a typical industrial application, the SHA-1 will be computed in software, the SHA-1 value is then sent to a HSM to compute the MAC on this value. The reason for doing the computation this way is due to the nature of HSMs. A typical HSM is built for transactional encryption techniques, like PIN verification. If we now send a big file to the HSM to MAC in the normal 3DES CBC-mode, the HSM will take a long time to compute all the blocks required. While this MAC operation is taking place on the HSM, no other transactions can be verified or completed. In a high volume transactional environment, this can be catastrophic. By computing the SHA-1 value in software, we restrict the size of the data sent to the HSM for encryption, thus speeding up the computation of the MAC.

Although a MAC computed in this fashion is cryptographically secure, it does not give us non-repudiation. In the next section we will look at the RSA digital signature that will provide this service to us as well.

2.4.5 Digital Signatures

Digital signatures can only be accomplished by means of public key cryptography. Computing the hash of a message and encrypting that hash with the sender's private key achieves this. When the receiver receives the message, he decrypts the encrypted hash with the sender's public key, and verifies the received hash with the hash he himself computes from the received message. In this way he can be sure that the person that sent him the message is the authenticated sender, and that the message has not been tampered with during transmission. Figure 2.12 illustrates this graphically.

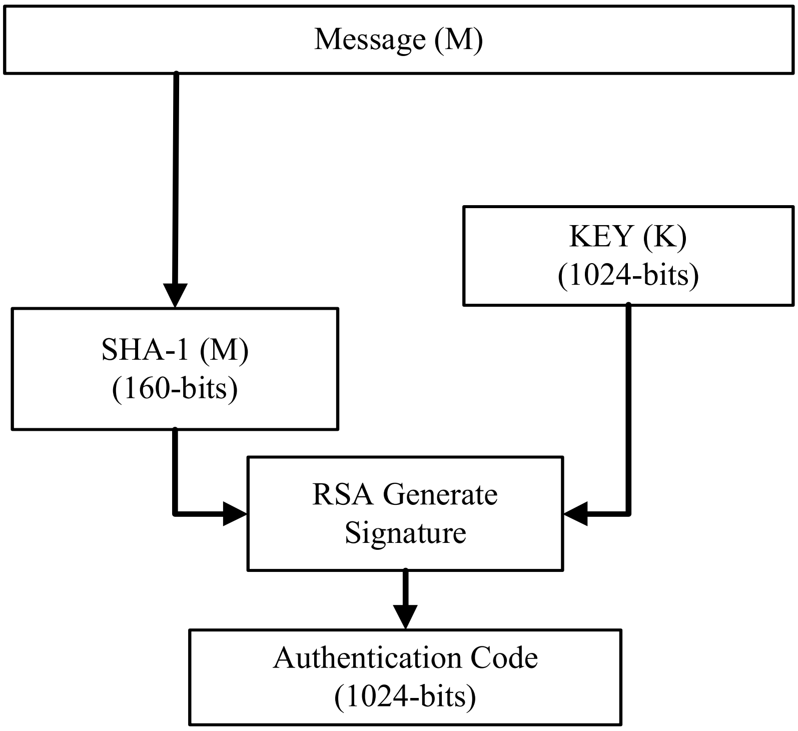


Figure 2.12: Generation of a RSA Digital Signature

In the industry, people are always transacting over untrusted networks. In order for an organisation to be able to communicate with other organisations or individuals, it is very important that they adhere to international standards. These standards assure that all communicating parties are using the same protocols to communicate. Just as TCP/IP became an international standard, so too have several cryptographic functions.

In the previous section we made use of SHA-1, 3DES and RSA as examples of these kinds of technologies, because of their standing as international standards. The author does not hereby imply that they are the only, or even the best tools to make use of. The selection of these functions comes from experience with communication failure due to the communicating parties making use of non-standards based options.

The public algorithms mentioned above have been scrutinised by subject matter experts. Their inherent vulnerabilities are therefore well known and documented. This is not the case with proprietary algorithms. Areas of vulnerability within proprietary algorithms are unknown and could pose serious threats. In a highly volatile environment such as banking, time-to-market is of utmost importance. Time and costs involved with programming proprietary algorithms may lead to a window of opportunity being missed.

Table 2.2 summarizes the services provided by each of the algorithms described above.

Crypto Tool	Authentication	Integrity	Non-repudiation	Confidentiality
HASH	0	1	0	0
Keyed HASH	1	2	0	0
HMAC	2	3	1	0
MAC	3	5	3	0
RSA Signature	5	5	5	0

Table 2-2: Services provided by cryptographic authentication

Where:

0 = No service provided,

5 = Full service provided.

CHAPTER 3: GSM NETWORK SECURITY

3.1 THE GSM NETWORK ARCHITECTURE

3.1.1 History of GSM

Analogue cellular telephone systems experienced rapid growth in Europe during the early 1980's. Each country developed their own system that was incompatible with all the others in both equipment and operation. This posed the undesirable problem of each system only being functional within national boundaries, limiting the possibilities of achieving economies of scale that would drive the prices of the equipment down to an affordable level.

Realising this, the Conference of European Posts and Telegraphs (CEPT) formed a study group called Groupe Spécial Mobile (GSM) to investigate and develop a pan-European public land mobile system [18]. The proposed system had to meet certain criteria: [19]

- Good subjective speech quality
- Low terminal and service cost
- Support for international roaming
- Ability to support handheld terminals
- Support for range of new services and facilities
- Spectral efficiency
- ISDN compatibility

These GSM networks offered enhanced features over analogue based systems, like [18]:

- **Total mobility**. The user has the advantage of being able to travel to different countries and still being able to communicate on his mobile device.

- **High capacity and optimal spectrum allocations**. By making use of smaller cells, the GSM networks could handle a higher capacity of calls. They utilise the assigned frequency bandwidth more efficiently than the older analogue systems.
- **Security**. Although not perfect, the security methods standardised for GSM systems makes it the most secure cellular telecommunications standard in use today.

GSM responsibility was transferred to the European Telecommunication Standards Institute (ETSI) in 1989 and phase I of the GSM specifications was published in 1990. The standards spread across the globe and by 1994 there were 1.3 million subscribers worldwide [19]. This figure grew exponentially and by October 1997 55 million subscribers were registered [19]. The new era has begun.

3.1.2 The GSM network

A GSM network is composed of several functional entities. Each of these entities' functions and interfaces are specified. Figure 6.1 shows the layout of the GSM network.

The GSM Public Land Mobile Network (PLMN) can be divided into three broad parts [19]:

3.1.2.1 <u>Mobile Station:</u>

The Mobile Station (MS) is carried by the subscriber and consists of the mobile equipment (terminal) and a Smart Card called the Subscriber Identity Module (SIM) [20]. The SIM can be transferred between numerous mobile devices allowing the user access to the subscribed mobile services [19].

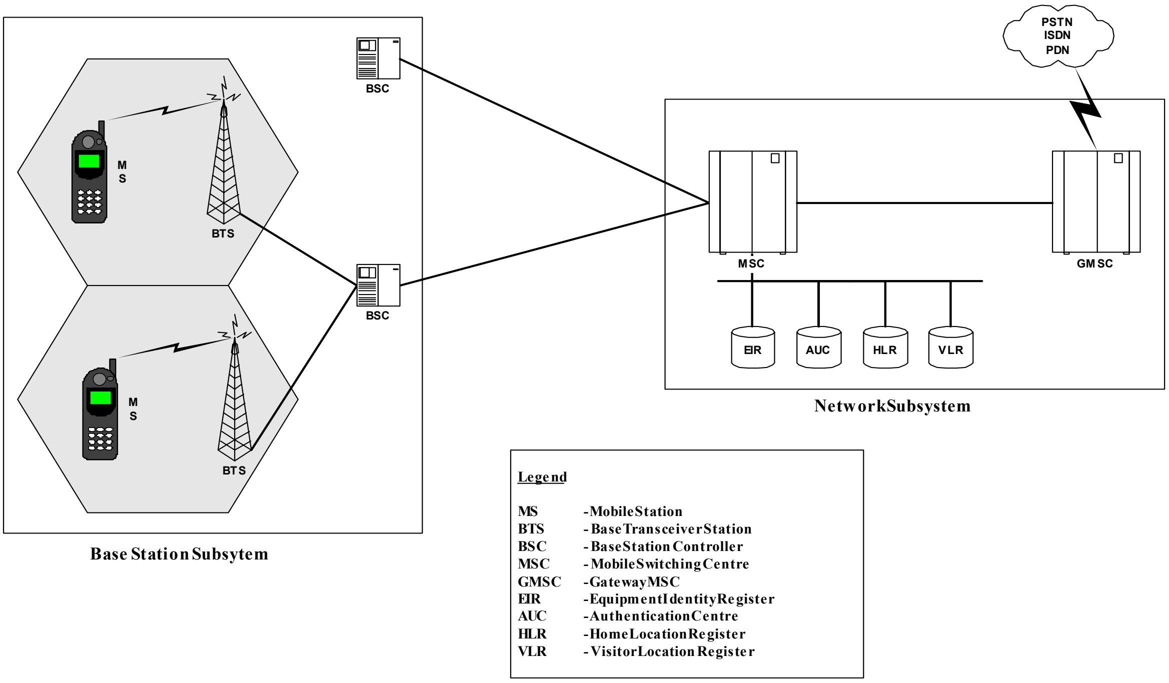

PSTN
ISDN
PDN
BSC
BSC
M
S
BTS
M
S
BTS
Base Station Subsytem
MSC
GMSC
EIR
AUC
HLR
VLR
NetworkSubsystem
Legend
MS -MobileStation
BTS -BaseTransceiverStation
BSC -BaseStation Controller
MSC -MobileSwitchingCentre
GMSC -GatewayMSC
EIR -EquipmentIdentityRegister
AUC -AuthenticationCentre
HLR -HomeLocationRegister
VLR -VisitorLocation Register

Figure 3.1: The GSM Network. [21]

3.1.2.2 The Base Station Subsystem:

The Base Station Subsystem (BSS) controls the radio link with the Mobile Station and consists of the Base Transceiver Station and the Base Station Controller. The Base Transceiver Station houses the radio transceivers that define a cell and handles the radio-link protocols with the Mobile Station [19]. The Base Station Controller on the other hand manages the radio resources for one or more BTSs [21]. Duties include radio channel set-up, frequency hopping, and handovers.

3.1.2.3 The Network Subsystem:

The Network Subsystem performs the switching of calls between mobiles and fixed network users and consists of [19] [20] [21]:

- *Mobile Switching Centre (MSC)*. It acts like a normal switching node of a PSTN or ISDN network. In addition it provides the functionality needed to handle a mobile subscriber, such as registration, authentication, location updating, and handovers and call routing to a roaming subscriber.

- *Home Location Register (HLR)*. The HLR contains all the administrative information of each subscriber registered in the corresponding GSM network. It also keeps track of the mobile's current location.
- *Visitor Location Register*. The VLR contains the selected administrative information from the HLR in order to provide call control for the subscribed services for each mobile within the geographical area controlled by the VLR.
- *Equipment Identity Register (EIR)* is a database that contains a list of all valid mobiles in the network, where each mobile is defined and its International Mobile Equipment Identity (IMEI). The IMEI can be marked as invalid if it is reported stolen for instance.
- *Authentication Centre (AuC)* is a protected database that stores a copy of the secret key stored in each subscriber's SIM card. This key is used for authentication and encryption over the radio channel and is discussed in detail later in this chapter.

3.2 THE GSM SECURITY MODEL

The motivations for security in cellular telecommunication systems are to secure conversations and signalling data from interception as well as to prevent cellular telephone fraud. This is especially true in the more traditional analogue systems mostly found in the U.S.A. The GSM System promises to provide security over the air interface that is equal to the security offered by guided media.

These security functions, which are adequate for normal cellular communication, are however not suitable for mobile commerce applications traversing these networks. In order for us to understand these shortcomings we need to investigate the current security provided by GSM and then leverage off that to enable secure mobile commerce. This section will look at the security of the GSM network.

3.2.1 SIM Card Security

The GSM Subscriber Identity Module (SIM) card is a cryptographic Smart Card with the GSM system specific applications loaded onto it. Being a Smart Card, the SIM has some inherent security functions specific to Smart Cards.

Smart Cards provide computing and business systems the enormous benefit of portable and secure storage of data and value. At the same time, the integration of Smart Cards into any system introduce its own security management issues as people access card data far and wide in a variety of applications. Smart Cards are used primarily on applications requiring high security, such as facilities access or in applications handling sensitive information such as financial applications on accessing the GSM network. A criminal could then potentially benefit financially by trying to break the security controls that are designed into a Smart Card. There are many reasons to use Smart Cards, but one of the main reasons must be the built-in security features a Smart Card can offer us. The microprocessor of the card has encryption keys and encryption algorithms built-in for performing cryptographic functions on the data of the card. The operating system file structure prevents the secret keys from being read from an external source.

We will now look at some of these features:

3.2.1.1 Security Features of the Smart Card Chip

It is necessary during production for the Smart Card chip to test the microcircuit. After the chip has been tested, the chip must be irreversibly converted to a mode where it is impossible to access the internal chip circuit, for example directly accessing memory from the outside. One of the last processes in the chip fabrication is to apply an electric current to the selected chip to blow a fusible link on the chip. Sometimes, in concert with blowing this fusible link, some manufacturers modify a location in EEPROM that logically changes the chip. The card operating system detects the blown link and/or reads this memory location to determine the current mode. Once changed at the factory, it is impossible to change the operating mode back to the service mode. This is typically the location where a GSM operator would want to store the unique subscriber key Ki of each subscriber during personalisation [22].

Other on-chip security features include burying components deep inside the chip in inaccessible locations. The ROM can be buried in the lower layers of silicon to prevent reverse engineering of the card operating system. The internal address and data buses that connect the major components on the chip together are scrambled so that individual conductors are interchanged to make it more difficult for the attacker to deduce their function [22].

Communication between on-chip components is encrypted on some Smart Card chips. To prevent electrical signals emitted by the memory cells from being monitored externally, the area of the chip around the EEPROM is coated in a metal shield. Removing this shield will destroy the chip and it will no longer function. The chip is also coated with a passivation layer to stop ultraviolet light from erasing the memory on the chip. Smart Cards have circuits to detect external tampering with the chip. There are also circuits to detect too high or too low supply voltage, too high or too low external clock frequency or sometimes too low an operating temperature [22].

3.2.1.2 Security Features of the Card Operating System

One of the enormous strengths of Smart Cards is the card operating system. All memory accesses must flow through the CPU. Thus the design of the card operating system is critical for implementing security at a logical level. The logical organisation of the dedicated files in EEPROM memory forms a security barrier [22].

When a dedicated file (DF) is selected for instance, the card operating system prevents access to data in other DFs. Access to Smart Card files can be protected with a Personal Identification Number (PIN) or with cryptographic keys. If a PIN is entered at the card terminal incorrectly, then after a number of attempts, the Smart Card could be inactivated. The number of attempts is determined by the Smart Card issuer. Depending on the design of the Smart Card, it is possible for the card issuer to reset the Smart Card if it is deactivated. Different PINs can be used to protect information in different directories [22].

3.2.1.3 Security Features of the Card Interface

Any system design should take into account the accessibility of data in transit and protect it accordingly, or design the transport protocol such that tampering will not affect the overall system security. One easy access point in the network is at the Smart Card contact pads. An attacker can insert the Smart Card into a hostile Smart Card reader whose purpose is to exercise the Smart Card and intercept the data stream flowing between the Smart Card and the Smart Card reader [22].

Encrypting all sensitive data that will exit the Smart Card through the contact pads is one way to thwart this attack. If the card terminal can be physically secured by building it into a wall for example, then equipping the card terminal with a motorized Smart Card reader with shutter will enhance the network security. The motorized Smart Card reader will draw the Smart Card into the machine when the card is inserted and seal the Smart Card in the Smart Card reader while it is in use. Anyone who has used an automated teller machine is familiar with this security method. The card remains inaccessible until the transaction is complete. The card could be retained if the system determines it is being misused [22].

Modern card terminals are just a part of a larger, more complex network of communications links between computers. These communications links must be physically protected from tampering if data integrity is to be maintained. The Smart Card reader and any communications links can be physically protected by placing them in a secured environment where personnel or monitoring equipment continuously observe the use of the Smart Card reader and prevent tampering [22].

Apart from these fundamental features of a SIM and Smart Card, the SIM also has the following characteristics [23]:

- The SIM has a small form factor that enables it to be removed from one handset and installed in another quite easily.
- The GSM subscriber is authenticated via a Personal Identification Number (PIN).

- The SIM has a PIN Retry Counter which only allows the incorrect PIN to be entered a certain number of times, where after the SIM can only be unlocked by using the PIN Unlocking Key (PUK).
- The applications on the SIM can be updated via keyboard, attached terminal equipment, or even over-the-air.
- The SIM handed to the subscriber is owned by the network operator the user subscribes too. This enables the network operator to manage the applications allowed on the SIM as well as control the "real estate" on the card.
- The SIM stores the IMSI of the subscriber.

3.2.2 Security Interactions with the Network Operator.

According to [24] the following security standards need to be implemented by every PLMN:

- Subscriber Identity (IMSI) authentication. This feature protects the mobile network from attack by impostors.
- Subscriber Identity (IMSI) confidentiality. This feature protects the Subscriber ID (IMSI) from attacks by eavesdroppers.
- Signalling and data confidentiality. Describes the protection of a user's signalling and data traversing the network.

In the GSM world each subscriber is uniquely identified via an International Mobile Subscriber Identity (IMSI) [25]. Further to this each handset/ Mobile Station (MS) is identified with the International Mobile Equipment Identity (IMEI) that operates a lot like the MAC addresses we see in normal Ethernet. The IMSI along with the individual subscriber authentication key (Ki) constitutes sensitive identification credentials that needs to be protected during transmission [18].

Let's look at each of these services individually.

3.2.2.1 Subscriber Identity Authentication

The subscriber identity authentication is used to enable the fixed GSM network to authenticate the identity of mobile subscribers. It is also used to establish and manage the encryption keys needed to provide the confidentiality services [26]. The GSM network authenticates the identity of the subscriber through a simple challenge-response protocol [18]. When the mobile subscriber attempts to access the network, a 128-bit random number (RAND) is sent to the Mobile Station (MS). The MS computes a 32-bit response (SRES) to RAND using the one-way A3 authentication algorithm with the individual subscriber authentication key Ki [18] [26]. Ki is a unique key allocated to each subscriber at subscription time, and is shared only by the MS and the authentication centre, which serves the subscriber's home network [25]. On the MS, Ki is securely stored on the subscriber's SIM [27]. The calculation of SRES is processed within the SIM on the MS. This entails that Ki never leaves the SIM, which in turn enhances the security of the authentication process [26].

Upon receipt of the SRES from the subscriber, the network performs a similar operation using the same algorithm and key. If the results of the received SRES and computed SRES match, the subscriber is authenticated and the call is allowed to proceed. If the values are however different, access to the system resources is disallowed. A graphical representation is shown in Figure 3.2: GSM Authentication [18].

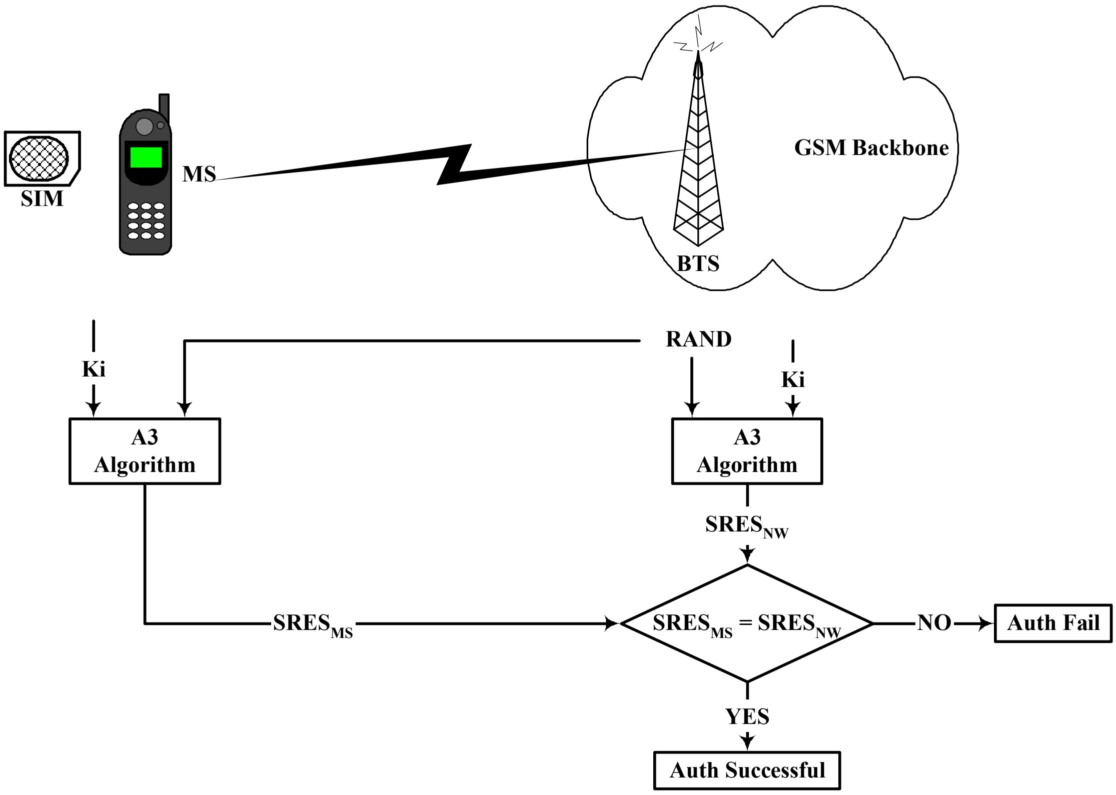


Figure 3.2: GSM Authentication [18].

3.2.2.2 Subscriber Identity Confidentiality

The purpose of this function is to avoid the possibility of an intruder to identify which subscriber is using a given resource on the radio path by listening to the signalling exchanges on the radio path. This provides confidentiality of the user's identity, signalling information as well as the user's location [25].

In order to provide the subscriber identity confidentiality service it is necessary to ensure that the IMSI or any information that allows the eves-dropper to derive the IMSI is not transmitted in clear in any signalling message on the radio path [26].

The designers of the GSM system opted for a Temporary Mobile Subscriber Identity (TMSI) to be used [18]. The TMSI is securely updated after each successful access attempt to the GSM system. The TMSI only has a meaning in the given location area, implying that the TMSI must be accompanied by the Location Area Identification (LAI) to avoid ambiguities. The mapping of the TMSI to the IMSI is done by the network and is typically handled by the VLR.

The TMSI updating mechanism functions in the following manner and is depicted in Figure 3.3: TMSI reallocation. [18]

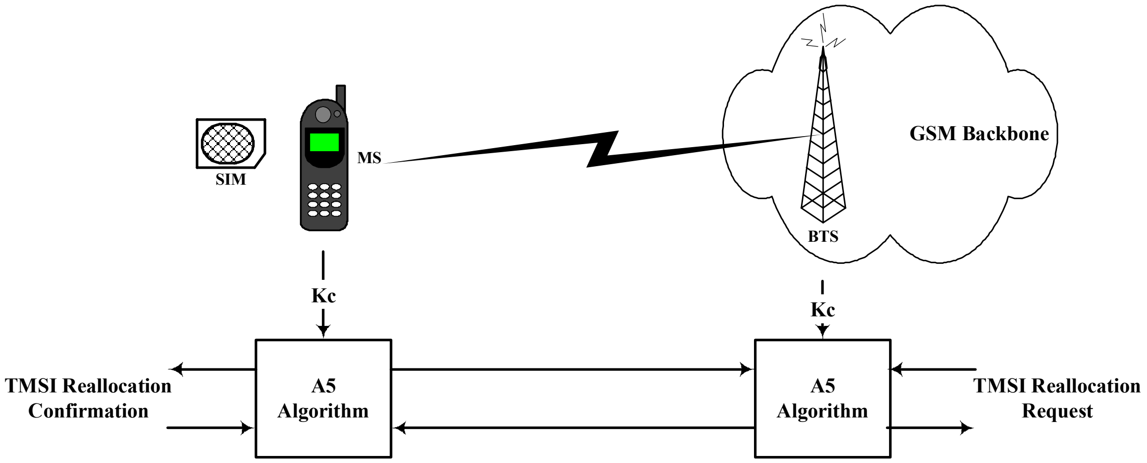


Figure 3.3: TMSI reallocation. [18]

For simplicity, let's assume the MS has been allocated a TMSI, denoted by $TMSI_o$. The network knows the association between $TMSI_o$ and the IMSI. The subscriber identifies itself to the network with $TMSI_o$. Immediately after authentication takes place the network generates a new $TMSI_n$ and sends it to the MS encrypted under the cipher key Kc. The MS receives the message and decrypts it and replaces $TMSI_o$ with $TMSI_n$ and uses this identifier on the network from now on [26]. Thus, if an eavesdropper uses a $TMSI_o$, the network will assume a relationship with an IMSI. The eavesdropper doesn't know which IMSI it should be associated with. A new TMSI must be encrypted using a key Kc that is only known by the correct SIM.

3.2.2.3 Signalling and data confidentiality

The same initial random number (RAND) and subscriber key (Ki) are also used to compute the ciphering key, Kc, by using an algorithm called A8 [28]. It is generally suspected that Kc has a length of 64-bits [25] [18] [29]. This ciphering key is used for the encryption and decryption of user and signalling data between the MS and the BTS [18] [26]. Kc is pre-computed for the network by the authentication centre that serves the subscriber's home network. This ensures that both the network and the MS have a fresh cipher key at the end of a successful authentication exchange [26].

The home network's authentication centre generates and stores a triplet, by using Ki, for each subscriber. The triplet contains RAND, SRES, and Kc [29]. This triplet is unique for each subscriber and is passed from the home network's authentication centre to visited networks on demand [26]. These random challenges are used just once, thus ensuring that the authentication centre never sends the same triplet to two distinct networks and that a network never re-uses the challenge [26]. This implies that after the subscriber leaves this location, his cipher key cannot be re-used by an attacker. An additional level of security is provided by having the ability to change the ciphering key at regular intervals as required by network design or security considerations [18]. Figure 3.4: Kc generation shows the key generation process graphically.

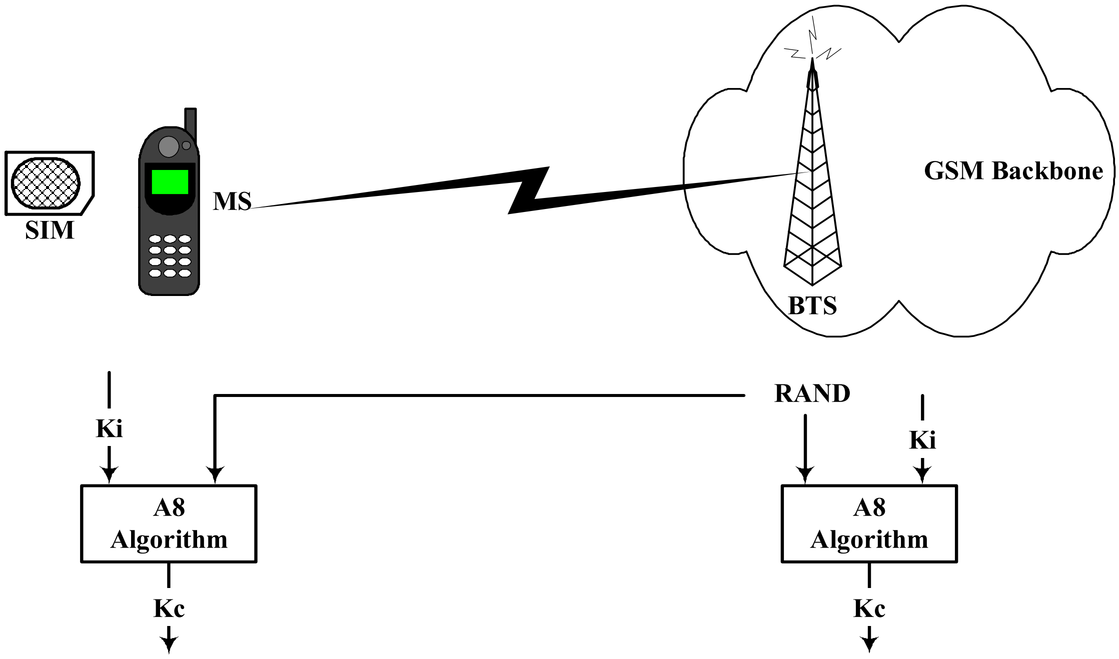


Figure 3.4: Kc generation. [18]

The signalling and data communication security service is provided by using the same encryption mechanism, and must be supported and used across all networks and mobiles [26]. The layer 1 data flow is encrypted by a bit-per-bit or stream cipher called A5 and the key Kc, which produces a key stream [25]. This key stream is the exclusive-or'ed (XOR) with the data transferred over the radio path between the MS and BTS [26]. The Encrypted communication is initiated by a ciphering mode request command from the GSM network. Upon receipt of this command, the MS begins encryption and decryption of data [18].

It is essential that the MS and BTS synchronise the starting of their cipher algorithms. The A5 algorithm is synchronised by using the TDMA frame structure of the radio sub-system, and adds very little complexity to the MS [29] [26].

The A5 algorithm is initiated with the session key (Kc) and the TDMA frame number as a message key [27] [26]. Although the same Kc is used throughout the call, the 22-bit frame number changes during the call, thus generating a unique key stream for each frame [27]. For each frame a total of 114-bits are produced for enciphering/ deciphering data transferred from the MS to the BTS, and an additional 114-bits are produced for deciphering/ enciphering data received at the MS from the BTS. A typical frame lasts for 4.6 ms, thus the cipher has to produce 228-bits in this time. This is graphically represented in Figure 3.5: A5 Key stream generation.

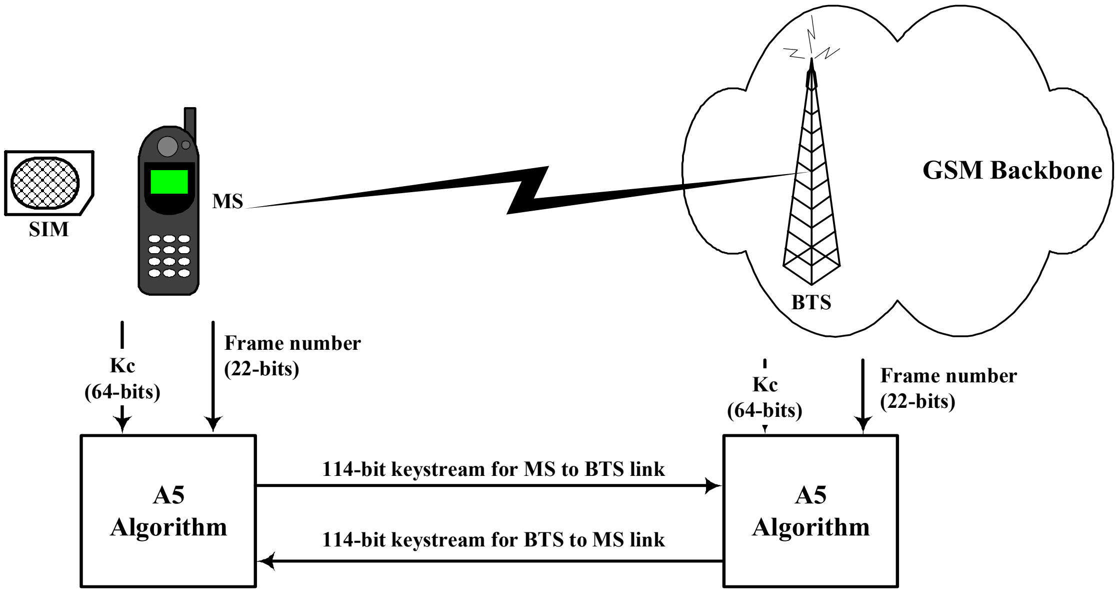


Figure 3.5: A5 Key stream generation. [18]

Detailed discussions on A3, A5 and A8 can be found in section 3.3

3.3 GSM AUTHENTICATION AND ENCRYPTION ALGORITHMS

In this section we will look at each of the GSM algorithms in more detail. It must however be noted that the implementation of these algorithms is a closely guarded secret. Speculations have been made with regards to the implementation of these algorithms and will be duly noted in the text.

3.3.1 The A3 Algorithm

The sole purpose of the one-way A3 algorithm is to allow authentication of a mobile subscriber's identity. This is done by a simple challenge response mechanism. This authentication is unilateral in nature, implying that the subscriber never knows if he is talking to a legitimate network.

The A3 algorithm receives a 128-bit random number (RAND) from the network, and must compute an expected 32-bit response (SRES) by making use of the unique 128-bit subscriber key Ki. A3 is contained in the SIM module at the MS, and in the AuC HLR on the network side. The run-time of the algorithm must not exceed 500ms [25].

It is perceivable that A3 works on the following principle:

- Take the 128-bit inputs of RAND and Ki and XOR them.
- Run the input through a hash algorithm like MD5 or SHA-1.
- Encrypt the output of the hash function with a symmetric key encryption algorithm like DES, and use the first 4-bytes of the output as SRES.

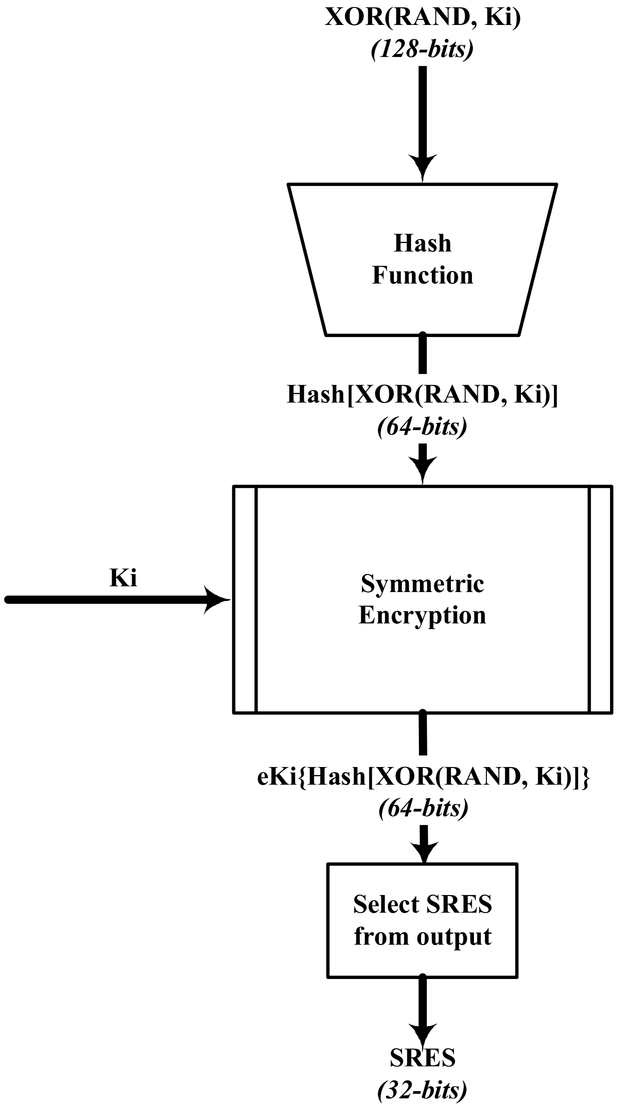


Figure 3.6: A possible implementation of A3

3.3.2 The A8 Algorithm

The A8 algorithm is a key generation algorithm and is used to compute the ciphering key Kc from the challenge 128-bit RAND sent during the authentication procedure and the unique 128-bit subscriber key Ki [25] [27]. The output of the A8 algorithm is the 64-bit session key Kc. The same Kc is generated by the HLR, as it has both Ki and RAND, and distributed to the BTS communicating with the MS. One session key is used until the MSC decides to authenticate the subscriber again, which can be anywhere from a couple of minutes to a couple of days.

It is generally accepted that nearly every GSM operator uses a single algorithm called COMP128 as an implementation of both the A3 and A8 algorithms in practice [27] [30]. The actual specification for COMP128 was never made public, but it was reversed engineered and cryptanalysed [30].

According to [30] COMP128 is a keyed hash function that takes a 128-bit key (Ki) and 128-bits of data (RAND) generating an output of 96-bits. This is in direct contradiction to the view of [27] which argues that COMP128 generates a 128-bit response. It is my opinion that the argument in [27] written in 1999 does not hold water and the implementation of COMP128 as described in [30] written in 2002 is the most likely solution. To this effect, we will look at the COMP128 as described by [30].

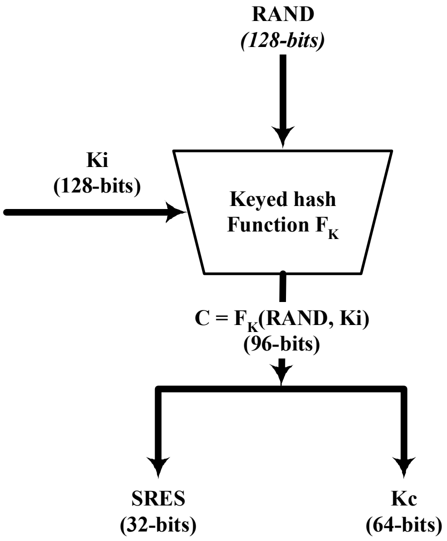


Figure 3.7: Logical Implementation of COMP128

COMP128 works in the following fashion:

- The COMP128 algorithm first loads Ki and RAND in a 32-byte vector X [].
  - Ki is stored in positions X [0...15] and RAND is stored in positions X [16...31].
- Eight iterative loops are applied on X [].
- Each iteration starts with a butterfly-structure like compression.
- The compression consists of five levels of table lookups using *T0 [512]*, *T1 [256]*, *T2 [128]*, *T3 [64]* and *T4 [32]* respectively.

- In each iteration except the last, a permutation follows the compression. Each Ti contains only (8-i)-bit values.
- Compression results in 32 4-bit values that are assembled into 16 bytes before the permutation is applied.
- These 16 bytes are stored into X [16...31] and Ki is loaded into X [0...15] before the next iteration begins.
- The resulting 128-bits after eight iterations are further compressed to 96-bits which forms the output of the algorithm.
- The first 32-bits of the output are SRES, and the remaining 64-bits are used as Kc.

3.3.3 The A5 Algorithm

Before we discuss the A5 algorithm itself, let's look at a couple of aspects of stream ciphers.

A stream cipher is a symmetric encryption algorithm. Stream ciphers can be designed to be exceptionally fast, much faster in fact than any block cipher. While block ciphers operate on large blocks of data, stream ciphers typically operate on smaller units of plaintext, usually bits. The encryption of any particular plaintext with a block cipher will result in the same ciphertext when the same key is used. With a stream cipher, the transformation of these smaller plaintext units will vary, depending on when they are encountered during the encryption process [14][15][16].

A stream cipher generates what is called a keystream and encryption is provided by combining the keystream with the plaintext, usually with the bitwise XOR operation. The generation of the keystream can be independent of the plaintext and ciphertext (yielding what is termed a synchronous stream cipher) or it can depend on the data and its encryption (in which case the stream cipher is said to be self-synchronizing). Most stream cipher designs are for synchronous stream ciphers [14][15][16].

Current interest in stream ciphers is most commonly attributed to the appealing theoretical properties of the one-time pad but there have been, as of yet, no attempts to standardize on any particular stream cipher proposal, as has been the case with block ciphers. Interestingly, certain modes of operation of a block cipher effectively transform it into a keystream generator and in this way; any block cipher can be used as a stream cipher. However, stream ciphers with a dedicated design are likely to be much faster [14][15][16].

A Linear Feedback Shift Register (LFSR) is a mechanism for generating a sequence of binary bits. The register consists of a series of cells that are set by an initialisation vector that is, most often, the secret key. A clock regulates the behaviour of the register and at each clocking instant, the contents of the cells of the register are shifted right by one position, and the XOR of a subset of the cell contents is placed in the leftmost cell. One bit of output is usually derived during this update procedure. LFSRs are fast and easy to implement in both hardware and software. With a judicious choice of feedback taps, the sequences that are generated can have a good statistical appearance. However, the sequences generated by single LFSRs are not secure because a powerful mathematical framework has been developed over the years, which allows for their straightforward analysis. Nevertheless LFSRs are useful as building blocks in more secure systems [16].

The over-the-air privacy of both signalling and user data is protected by the A5 stream cipher and is implemented into both the MS and the BSS [25][31]. A GSM conversation is sent as a sequence of frames every 4.6 milliseconds. Each frame consists of 114-bits representing the digitised A to B communication, and 114-bits representing the B to A communication. Each conversation, or SMS, can be encrypted with a new session key Kc. For each frame, Kc is mixed with a publicly known frame counter Fn. The result serves as the initial state of a generator that produces 228 pseudo random bits. These bits are XORed by the two parties with the 114+114 bits of the plaintext to produce 114+114 bits of ciphertext [31]. Let's look at the operation of A5/1.

- A5/1 is built from three short LFSRs of lengths 19, 22 and 23-bits. They are denoted by *R1*, *R2* and *R3* respectively. The rightmost bit in each register in labelled as bit zero. The taps of *R1* are at bit positions 13, 16, 17 and 18. The taps of *R2* are at bit positions 20 and 21. The taps of *R3* are at bit positions 7, 20, 21 and 22 as illustrated in Figure 3.8 [31].

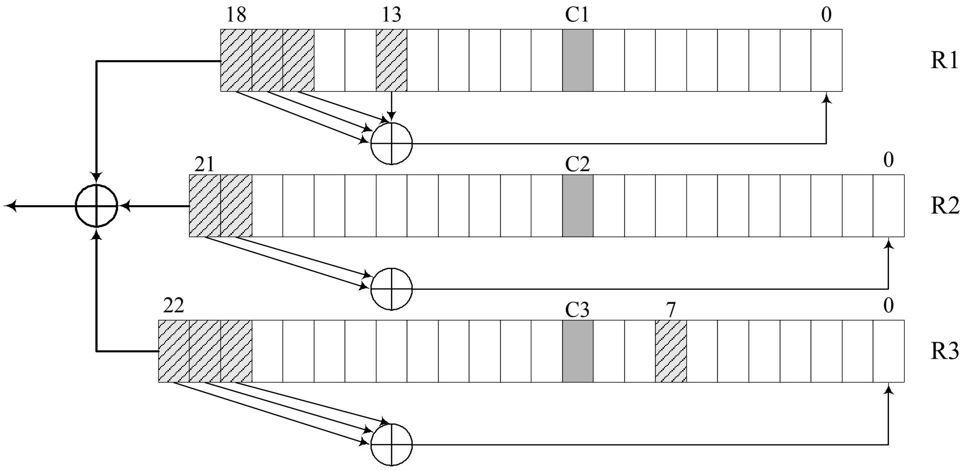


Figure 3.8: The A5/1 stream cipher. [31]

When a register is clocked, its taps are XORed together, and the result is stored in the rightmost bit of the left-shifted register. The three registers are maximal length LFSRs with periods 2^{19}-1, 2^{22}-1, and 2^{23}-1 respectively. The registers are clocked in a stop/go fashion using the following majority rule:

- Each register has a single "clocking" tap (bit 8 for *R1*, bit 10 for *R2* and *R3*). For each clock cycle the majority function of the clocking taps is calculated and only those registers whose clocking taps agree with the majority bit is actually clocked. This implies that at each step, two or three registers are clocked and that each register moves with probability ¾ and stops with probability ¼ [31].

The process of generating pseudo random bits from the session key Kc and the frame counter Fn is carried out in four steps [31]:

1. The three registers are zeroed, and then clocked for 64 cycles. During this period each bit of Kc is XORed in parallel into the least significant bits of the three registers.
2. The three registers are clocked for 22 additional cycles. During this period the successive bits of Fn are again XORed in parallel into the least significant bits of the three registers. The content of the registers at the end of this step is called the initial state of the frame.
3. The three registers are clocked for 100 additional clock cycles with the stop/go clock control but without producing any outputs.

4. The three registers are clocked for 228 additional clock cycles with the stop/go control in order to produce the 228 output bits. At each clock cycle one output bit is produced as the XOR of the most significant bits of the three registers.

CHAPTER 4: EVALUATION OF GSM SECURITY

4.1 INTRODUCTION

The GSM infrastructure for subscriber authentication and confidentiality of communication sessions represented a major advance over first-generation analogue cellular systems. However, as GSM has matured and expanded its reach across Europe and beyond, its basic security mechanisms have come under increasing criticism. Given the strong belief in the security community that only protocols that can be tested should be trusted (that security should depend on the secrecy of keys and not of algorithms), it was inevitable that GSM would be attacked for its dependency on the proprietary A3, A8, and A5 algorithms. Many security analysts view these algorithms as cryptographically weak, and subject to intrusion by government agencies, in addition to well-equipped hackers. In this chapter we will look at some of the problems faced by GSM security in an effort to understand what security GSM can provide us, and what security we have to build into our system in order to cater for the services not provided.

4.2 PROBLEMS WITH OTA SECURITY

GSM makes use of the A3 and A8 algorithms to authenticate the user and generate the session key for secure transmission of user and signalling data over the air. Both A3 and A8 are typically implemented by GSM service providers with an algorithm called COMP128. COMP128 was reverse engineered at Berkeley in 1998, and cryptanalysis by Berkeley researchers indicates that the protocol can be broken with 219 queries from a rogue base station to the GSM SIM card, which can be achieved in eight hours. Analysis of the GSM application of COMP128 further revealed that it had apparently been deliberately weakened. The algorithm calls for a 64-bit key, but of this total, ten key bits are consistently set to zero, dramatically reducing the security offered by the A8 implementation.

In [30] the authors develop a new kind of side-channel attack on the SIM card, which they call a partitioning attack. The proposed attack makes use of a side-channel attack strategy. Cryptographic algorithms are traditionally designed to withstand attacks that treat the implementation as a black box. These attacks usually focus on and exploit the subtle relationships between inputs and outputs from the black box. There are however more information available to a determined attacker than just inputs and outputs. Sensitive information could for instance be obtained from side-channels such as timing of operations, power consumption, electromagnetic emanations and the likes [30]. Using these techniques the authors in [30] show how they can extract the 128-bit subscriber key (Ki), by utilising only 1000 invocations with random inputs, or 255 chosen inputs, or only 8 adaptively chosen inputs. It is their opinion that an attacker only needs to gain physical access to the subscribers SIM card for 1 minute in order to extract the Ki value from the SIM.

For an attacker this is great news. Having access to the Ki of the subscriber, the attacker can easily clone the unsuspecting subscriber’s SIM card. For all intents and purposes, the attacker can become the legal subscriber on the GSM network with no one being the wiser. With access to Ki, the attacker can freely authenticate on the GSM network and generate session keys (Kc) by means of the COMP128 algorithm, or can he?

There is a slight kink in the attacker’s armour. GSM networks will only allow access to the network for one SIM card at any given time. This means that if the attacker and the original subscriber try to access the GSM network at the same time, the network will realise that duplicate cards exist, determine that they reside in different locations, and disable the affected account, thus denying access to both the attacker and the legitimate subscriber [32].

A further problem exists in the GSM authentication scenario. Under the GSM authentication protocol, the GSM base station authenticates the Mobile Station, which seeks to establish a communications session, i.e. unilateral authentication [25]. However, the opposite is not true. Thus the Mobile Station has no guarantee that it is not communicating with a node, which is impersonating a GSM base station. To make the situation worse, the same random challenge (RAND) that is used to authenticate a mobile station, when presented to the A8 algorithm, also becomes the seed for the generation of a session key Ki [23]. Furthermore, the authentication challenge-response message protocol does not include a time-stamp. Thus, if a rogue station does successfully impersonate a GSM base station, it may secure a session key that will allow the decryption of any messages sent with the same key over a potentially prolonged period.

The A5 encryption algorithm in itself is not so secure either. Of the two variants of the A5 data encryption algorithm, the weaker, called A5/2, can be exported anywhere in the world without restriction. It is believed that A5/2 can be broken in real time with approximately 2^{16} steps. A5/1, the stronger of the two variants, is susceptible to attacks that can break it with approximately 2^{40} steps [31]. This level of security makes it vulnerable against hardware-based attacks by large organisations. In [31] however, the authors suggest a new kind of attack on A5/1 exploiting the subtle flaws in the tap structure of the registers. After an initial 2^{48} parallelisable data preparation stage, the attacks can be carried out in real time on a single PC. The two attacks are related, but use different types of time-memory tradeoffs. Two attacks are demonstrated:

- The first attack requires the output of the A5/1 algorithm during the first two minutes of the conversation and computes the key, Kc, in roughly one second.
- The second attack requires the output of the A5/1 algorithm during about two seconds of the conversation and computes the key, Kc, in a couple of minutes.

As can be seen from the above discussion, there are numerous flaws in both the GSM authentication and encryption algorithms. This is largely due to the fact that the algorithms were developed in isolation and not put under public scrutiny. Even though the GSM forum tried to keep the algorithms secret, some of them were reverse engineered or even leaked to the public. Whenever security is obtained through obscurity, failure is guaranteed.

4.3 PROBLEMS WITH THE GSM BACKBONE

GSM authentication (and security in general) protects the wireless link between the Mobile Station (MS) and the GSM base station that is serving it. The security mechanisms do not protect the transmission of information between the Authentication Centre on the user's home network and the serving network for instance.

This lack of security in the wired network represents a major exposure for GSM, particularly in light of the fact that communications between GSM base stations and the true wired network are often transmitted over microwave links that make interception easy.

Mobile networks primarily use Signalling System no. 7 (SS7) for communication between networks for such activities as authentication, location update, and supplementary services and call control. The messages unique to mobile communications are Mobile Application Part (MAP) messages. The security of the global SS7 network as a transport system for signalling messages e.g. authentication and supplementary services such as call forwarding is open to major compromise [23]. Major SS7 vulnerabilities arise from the number and complexity of interfaces between distinct SS7 entities [33].

There is also exponential growth in the use of interconnection between the telecommunication networks and the Internet. Additional vulnerabilities arise from this interdependence and interconnection between SS7 networks and the Internet. SS7 systems are interconnected using Internet technologies and sometimes even the Internet itself.

SS7 incorporates very limited authentication procedures due to the fact that it was originally designed for a closed telecommunications community [33]. In the past, SS7 traffic was passed between major Public Telephone Operators covered under treaty organisation and the number of operators was relatively small and the risk of compromise was low. Networks are getting smaller and more numerous. Opportunities for unintentional mishaps will increase, as will the opportunities for hackers and other abusers of networks [23]. The problem with the current SS7 system is that messages can be altered, injected or deleted into the global SS7 networks in an uncontrolled manner. This poses the risk that anyone capable of generating SS7 messages and introducing them onto the network can disrupt services and even initiate transactions [33]. With the increase in different types of operators and the increase in the number of interconnection circuits there is an ever-growing loss of control of security of the signalling networks [23].

The IT community now has access to many protocol converters for conversion of SS7 data to IP. These protocol converters are primarily used for the transportation of voice and data over the IP networks in Voice over IP (VoIP) applications.

In addition, new services such as those based on integrated networks of the future, will lead to a growing use of the SS7 network for general data transfers. There have been a number of incidents from accidental action, which have damaged a network, and very few deliberate actions, which have been recorded [23].

The availability of cheap PC based equipment that can be used to access networks and the ready availability of access gateways on the Internet will lead to compromise of SS7 signalling and this will affect mobile operators. This is especially true for users with ISDN connections, as they can introduce digital messages directly into SS7 networks [33]. The risk of attack has been recognized in the USA at the highest level, which indicates the level of concern that exists about SS7 security. For the network operator there is some policing of incoming signalling on most switches already. These policing facilities are however dependent on the make of switch as well as on the way the switch is configured by operators. Some engineering equipment available is not substantially different from other advanced protocol analysers in terms of its fraud potential, but they are more intelligent and can be programmed more easily [23].

Operators ensure that signalling screening of SS7 incoming messages takes place at the ingress to their networks and that operations and maintenance systems alert against unusual or suspicious SS7 messages. There are a number of messages that can have a significant effect on the operation of the network and inappropriate messages should be controlled at the ingress to the network. Network operator's security engineers should on a regular basis carry out monitoring of signalling links for these inappropriate messages. In signing agreements with roaming partners and carrying out roaming testing a network operator should review signalling messages with the partner and also to seek appropriate confirmation that the partner is also screening incoming SS7 messages to his network in order to ensure that no rogue messages appear, or that they are handled in an agreed method [23].

Operators must be concerned about unauthorised access to their Network Elements and their Operations Support Systems. External access gained through Internet or dialup is a concern but also internal fraud such as modification of billing records. Very few operators audit security logs or have capabilities to detect intrusions in their network. Network Intelligence is also transferred from switches to UNIX platforms, increasing their exposure to "traditional" security issues associated with that operating system [15].

As numerous network elements in the GSM operator backbone are connected to IP networks in order to enhance the remote management capabilities of engineers, an attacker might access these elements. Unauthorised access to a HLR could result in activating subscribers that is not seen by the billing system, and therefore not chargeable. An attacker might also activate or deactivate certain services for a subscriber, thus allowing unauthorised access to services or denial of service attacks. In certain circumstances it is possible to use Man-Machine Language (MML) commands to monitor other HLR user's action which could allow for unauthorised access to data and even cryptographic information like a triplet [23].

An operator should not rely on the fact that an intruder's knowledge on particular vendor's MML language will be limited. Those attacks can be performed both by external intruders and by operator's employees. Access control to a HLR should be based on user profiles, using at least a unique username and a password as authentication data. Remote access to a HLR should be protected from eavesdropping, source and destination spoofing and session hijacking. An operator may therefore wish to limit the range of protocols available for communication with a HLR [23].

The number of employees that has physical and logical access to AuC should be limited. Operators should carefully consider the need for encryption of AuC data. Some vendors use default encryption with an algorithm that is proprietary and confidential. As stated earlier, the strength of such encryption could be questionable as it has not been publicly scrutinized [34]. If it is decided to use an off-the-shelf ciphering facility, the network operator should pay attention to cryptographic key management and should always store these cryptographic keys in an HSM. Careless use of off the shelf equipment could even compromise AuC security. A further vulnerability of the network exists in that authentication triplets can be obtained from AuC by masquerading as another system entity like an HLR. The threat is only present when a HLR and AuC are physically separated.

The MSC is one of the most important nodes of any GSM network. It handles all calls incoming to, or originating from subscribers visiting the given switch area. Unauthorised access to an MSC would likely result in the loss of confidentiality of user data and even a denial of service for large numbers of subscribers. Access to MSCs should be heavily restricted being it physical or logical access. The physical location of the MSC should not be made public as an attacker can try and disrupt services. A redundant deployment of several MSCs should be implemented in order to limit the impacts from accidents on one particular MSC (e.g. fire, flood etc.).

4.4 ADVANCED ATTACKS ON GSM

The author in [23] suggests some further advanced attacks on the GSM network. Although the author of this dissertation cannot confirm these attacks, it is included here in the interest of completeness. The author of this dissertation would like to focus the reader's attention to the fact that GSM is not secure enough for high value financial transactions and that measures needs to be put in place in order to enable secure m-Commerce over this channel.

The author in [23] suggests the following attacks on the GSM system are possible:

4.4.1 Eavesdropping:

This is the capability that the intruder eavesdrops signalling and data connections associated with other users. The required equipment is a modified MS.

4.4.2 Impersonation of a user:

This is the capability whereby the intruder sends signalling and/or user data to the network, in an attempt to make the network believe they originate from the target user. The required equipment is again a modified MS.

4.4.3 Impersonation of the network:

This is the capability whereby the intruder sends signalling and/or user data to the target user, in an attempt to make the target user believe they originate from a genuine network. The required equipment is modified BTS.

4.4.4 Man-in-the-middle:

This is the capability whereby the intruder puts itself in between the target user and a genuine network and has the ability to eavesdrop, modify, delete, re-order, replay, and spoof signalling and user data messages exchanged between the two parties. The required equipment is modified BTS in conjunction with a modified MS.

4.4.5 Compromising authentication vectors in the network:

The intruder possesses a compromised authentication vector, which may include challenge/response pairs, cipher keys, and integrity keys. This data may have been obtained by compromising network nodes or by intercepting signalling messages on network links.

4.4.6 De-registration spoofing:

An attack that requires a modified MS and exploits the weakness that the network cannot authenticate the messages it receives over the radio interface. The intruder spoofs a de-registration request (IMSI detach) to the network. The network de-registers the user from the visited location area and instructs the HLR to do the same. The user is subsequently unreachable for mobile terminated services.

4.4.7 Location update spoofing:

An attack that requires a modified MS and exploits the weakness that the network cannot authenticate the messages it receives over the radio interface. The user spoofs a location update request in a different location area from the one in which the user is roaming. The network registers in the new location area and the target user will be paged in that new area. The user is subsequently unreachable for mobile terminated services.

4.4.8 Camping on a false BTS:

An attack that requires a modified BTS and exploits the weakness that a user can be enticed to camp on a false base station. Once the target user camps on the radio channels of a false base station, the target user is out of reach of the paging signals of the serving network in which he is registered.

4.4.9 Camping on false BTS/MS:

An attack that requires a modified BTS/MS and exploits the weakness that a user can be enticed to camp on a false base station. A false BTS/MS can act as a repeater for some time and can relay some requests in between the network and the target user, but subsequently modify or ignore certain service requests and/or paging messages related to the target user.

4.4.10 Passive Identity Caching:

A passive attack that requires a modified MS and exploits the weakness that the network may sometimes request the user to send its identity in clear text.

4.4.11 Active Identity Caching:

An active attack that requires a modified BTS and exploits the weakness that the network may request the MS to send its permanent user identity in clear text. An intruder entices the target user to camp on its false BTS and subsequently requests the target user to send its permanent user identity in clear text perhaps by forcing a new registration or by claiming a temporary identity mismatch due to database failure.

4.4.12 Suppressing encryption between the target user and the intruder:

An attack that requires a modified BTS and that exploits the weakness that the MS cannot authenticate messages received over the radio interface. The target user is enticed to camp on the false BTS. When the intruder or the target user initiates a service, the intruder does not enable encryption by spoofing the cipher mode command. The intruder maintains the call as long as it is required or as long as his attack remains undetected.

4.4.13 Suppressing encryption between target user and the true network:

An attack that requires a modified BTS/MS and that exploits the weakness that the network cannot authenticate messages received over the radio interface. The target user is enticed to camp on the false BTS/MS. When a call is set-up the false BTS/MS modifies the ciphering capabilities of the MS to make it appear to the network that a genuine incompatibility exists between the network and the mobile station. The network may then decide to establish an un-enciphered connection. After the decision not to cipher has been taken, the intruder cuts the connection with the network and impersonates the network to the target user.

4.4.14 Compromised cipher key:

An attack that requires a modified BTS and the possession by the intruder of a compromised authentication vector and thus exploits the weakness that the user has no control upon the cipher key. The target user is enticed to camp on the false BTS/MS. When a call is set-up the false BTS/MS forces the use of a compromised cipher key on the mobile user.

4.4.15 Eavesdropping on user data by suppressing encryption:

An attack that requires a modified BTS/MS and that exploits the weakness that the MS cannot authenticate messages received over the radio interface. The target user is enticed to camp on the false BTS. When the target user or the intruder initiates a call the network does not enable encryption by spoofing the cipher mode command. The attacker however sets up his own connection with the genuine network using his own subscription. The attacker may then subsequently eavesdrop on the transmitted user data.

4.4.16 Suppression of encryption between target user and true network:

The target user is enticed to camp on the false BTS/MS. When the target user or the genuine network sets up a connection, the false BTS/MS modifies the ciphering capabilities of the MS to make it appear to the network that a genuine incompatibility exists between the network and the mobile station. The network may then decide to establish an un-enciphered connection. After the decision not to cipher has been taken, the intruder may eavesdrop on the user data.

4.4.17 Eavesdropping on user data by forcing the use of a compromised cipher key:

An attack that requires a modified BTS/MS and the possession by the intruder of a compromised authentication vector and thus exploits the weakness that the user has no control the cipher key. The target user is enticed to camp on the false BTS/MS. When the target user or the intruder set-up a service, the false BTS/MS forces the use of a compromised cipher key on the mobile user while it builds up a connection with the genuine network using its own subscription.

4.4.18 User impersonation with compromised authentication vector:

An attack that requires a modified MS and the possession by the intruder of a compromised authentication vector, which is intended to be used by the network to authenticate a legitimate user. The intruder uses that data to impersonate the target user towards the network and the other party.

4.4.19 User impersonation through eavesdropped authentication response:

An attack that requires a modified MS and exploits the weakness that an authentication vector may be used several times. The intruder eavesdrops on the authentication response sent by the user and uses that when the same challenge is sent later on. Subsequently, ciphering has to be avoided by any of the mechanisms described above. The intruder uses the eavesdropped response data to impersonate the target user towards the network and the other party.

4.4.20 Hijacking outgoing calls in networks with encryption disabled:

This attack requires a modified BTS/MS. While the target user camps on the false base station, the intruder pages the target user for an incoming call. The user then initiates the call set-up procedure, which the intruder allows to occur between the serving network and the target user, modifying the signalling elements such that, for the serving network it appears as if the target user wants to set-up a mobile originated call. The network does not enable encryption. After authentication the intruder cuts the connection with the target user, and subsequently uses the connection with the network to make fraudulent calls on the target user's subscription.

4.4.21 Hijacking outgoing calls in networks with encryption enabled:

This attack requires a modified BTS/MS. In addition to the previous attack this time the intruder has to attempt to suppress encryption by modification of the message in which the MS informs the network of its ciphering capabilities.

4.4.22 Hijacking incoming calls in networks with encryption disabled:

This attack requires a modified BTS/MS. While the target user camps on the false base station, an associate of the intruder makes a call to the target user's number. The intruder acts as a relay between the network and the target user until authentication and call set-up has been performed between target user and serving network. The network does not enable encryption. After authentication and call set-up the intruder releases the target user, and subsequently uses the connection to answer the call made by his associate. The target user will have to pay for the roaming leg.

4.4.23 Hijacking incoming calls in networks with encryption enabled:

This attack requires a modified BTS/MS. In addition to the previous attack this time the intruder has to suppress encryption.

CHAPTER 5: SECURITY OF WIG BASED MOBILE COMMERCE

5.1 INTRODUCTION

Wireless Internet Gateway (WIG) is a technology available on the GSM network that enables a network operator to effect wireless GSM based payment instructions to financial institutions. In this chapter the architecture of the WIG application, dependencies on other technologies, and the security thereof as implemented in the South African environment will be discussed.

We will firstly look at the security of the service technologies that can provide WIG applications to us, namely Unstructured Supplementary Services Data (USSD) and Short Message Service (SMS). Thereafter we will look at enabling applications by means of the SIM Application Toolkit (SAT) and the security of the SAT.

Once the whole architecture is built we will do a brief analysis of the WIG solution and then propose a model that will provide secure m-Commerce transactions via WIG.

5.2 UNSTRUCTURED SUPPLEMENTARY SERVICES DATA (USSD)

5.2.1 Introduction to USSD

USSD is a session-oriented technology, unlike SMS, which is a store-and-forward technology. Turnaround response times for interactive applications are shorter for USSD than SMS because of the session-based feature of USSD [35].

Users do not need to access any particular handset menu to access services with USSD, as they can enter the USSD command direct from the initial mobile phone screen. USSD commands are routed back to the home mobile network's HLR, allowing the ability for services, based on USSD, to work just as well and in exactly the same way when users are roaming. USSD works on all existing GSM mobile phones, and both SIM Application Toolkit and Wireless Application Protocol (WAP) support USSD [35][36].

5.2.2 Operation of USSD

In operation, USSD is used to send text between the user and an application. USSD should be thought of as a trigger rather than an application itself, as it enables other applications such as prepaid top-ups. USSD provides an ideal way for subscribers to request changes to their class of service, request that enhanced services are performed, or to perform a mobile originated payment instruction [36]. To achieve this, the sequence of operations is as follows:

- A Subscriber sends a mobile originating USSD message
- The USSD message is routed to the subscriber's HLR in accordance with the GSM recommendations
- The HLR forwards the USSD message to the USSD Gateway
- The USSD Gateway communicates the message to external applications using TCP/IP, which is more convenient for integration with commercial computing platforms.
- The external system interprets the message and performs the action indicated by the content of the message.
- Within a time-out period, the external system acknowledges successful receipt of the message to the mobile via the USSD Gateway. The external system can later asynchronously send further information to the mobile as a Short Message via an SMS.

5.2.3 Benefits offered by USSD

The primary benefit of USSD is that it allows for very fast communication between the user and an application. Most of the applications enabled by USSD are menu based and include services such as mobile prepay and chat. Some key benefits of USSD are [35]:

- Easy to use: Keying a digit string can be easier for a user than formatting a short message. Strings may be stored under abbreviated dial keys on the handset.
- USSD messages are very flexible in both length and content.
- USSD is faster than SMS.
- Roaming is supported. Because messages are exchanged with your HLR, services are still available when roaming.
- Service access codes and service names may be downloaded to the handset using Over the Air Programming. This makes it even easier for the user to get started.

5.2.4 Difference between USSD and SMS

The question might arise as to what are the differences between USSD and SMS. USSD is not store and forward and does not offer retries. This enables it to be simpler and faster than SMS. USSD does not offer guaranteed delivery, but any failures are reported back to the originator. USSD should achieve many times the speed of SMS due to its simplicity and reduced reliance on non-volatile storage. USSD offers a simple TCP/IP interface to external applications, which knows nothing of the SS7 network. Routing to applications is achieved via a simple service code, which is contained in the USSD message. The interpretation of the Service Code is achieved by configuration of the USSD Gateway and by the actions of the external application to which the service code relates. The external applications can be on any machine reachable by a TCP/IP network [35][36].

5.2.5 Security of USSD

USSD cannot be viewed as a trusted or secure channel from a financial point of view. To understand this concept, we need to look at the structure of a USSD message. The formatting of a USSD messages can be summarized as follows [36]:

- An asterisk is used to separate each of the parameters
- A service code of 2 or 3 digits is entered
- Supplementary information can then be entered. This may be of variable length. As an example, a PIN may be used as a measure of security.
- The # key terminates a request.

This message string is then transmitted across the network to the subscribers HLR. We know that the transmission is not encrypted or secured with an integrity check on the GSM backbone. This makes these USSD messages highly vulnerable to attack. An attacker can thus easily delete, alter, or even fabricate false messages on the network. For service alteration requests this does not pose a problem. If however we are initialing financial transactions on this channel, the picture changes rapidly.

Let's look at a possible attack on a financial application in order to demonstrate the risk associated with the USSD channel.

A client of Bank A is resident in South Africa, but is currently roaming in Europe. He has registered with Bank A as an m-Banking user via their normal client registration guidelines. Bank A provides him a payment application, based on USSD, whereby he can pay monies from his Credit Card to any other, Card Association endorsed, Credit Card. Client A just conducted a business deal and needs to pay Client B, based at Bank B, the amount specified by the transaction. Client A decides to make use of his mobile facility to effect the transaction and retrieves the relevant information from the intended recipient.

Client A now initiates the payment instruction via USSD to Bank A as depicted in Figure 5.1. Client A has memorized the instruction, and also needs to enter his PIN for the transaction to be affected. Client A enters the following string into his GSM based mobile phone: *184*1234*1*50000*5120123412341234*+411234567890#, where:

*184 = the service code

*1234 = PIN of Client A

*1 = the account indicator from which the transaction should take place

*50000 = the amount to be paid in Client A's local currency

*5120123412341234 = the account number that the payment should be made too.

*+411234567890 = the mobile number where the payment confirmation should be sent.

# = request terminator.

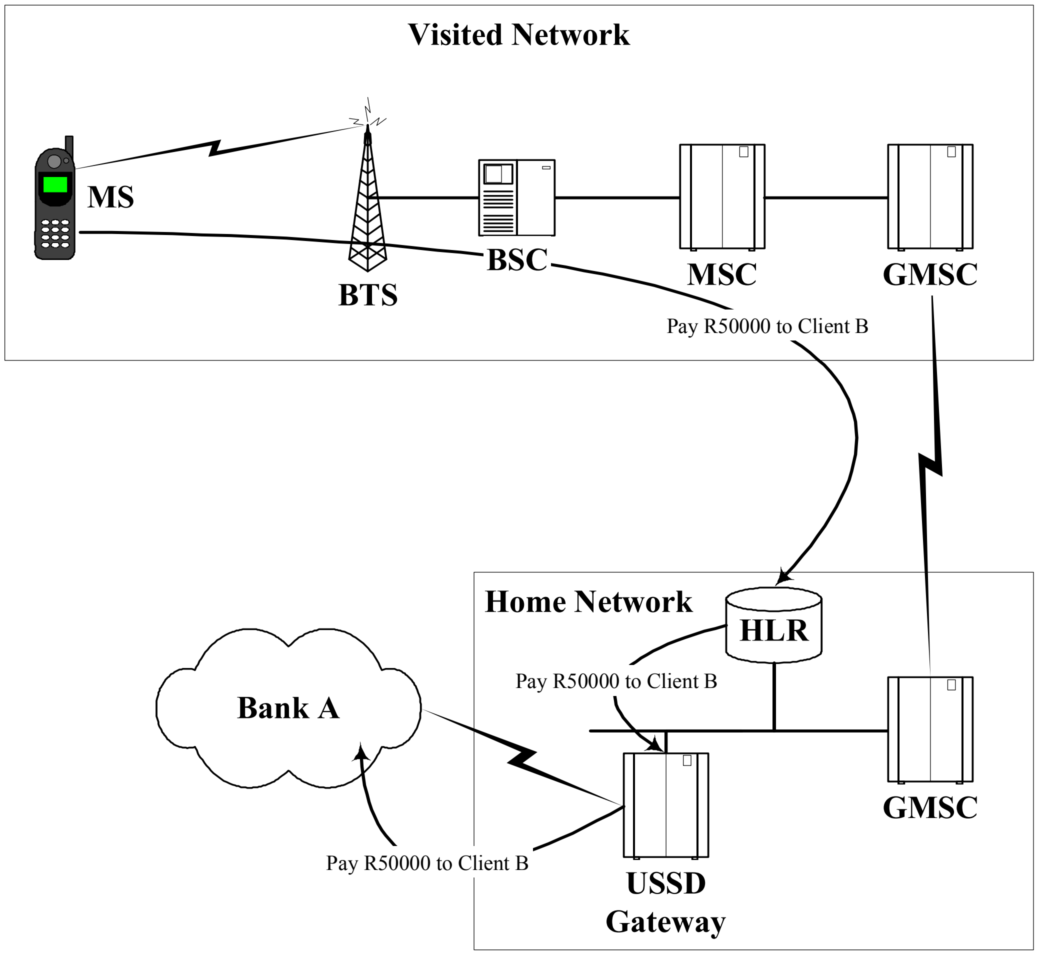


Figure 5.1: Normal USSD Transaction Flow

Client B however, has high gambling debt with a loan shark. Client B also has some friends in a fraud syndicate and tells them that Client A will be paying the agreed amount via his m-Banking application within the near future, and that he would like them to steal as much money as possible in order to pay for his gambling debt.

The syndicate in turn has access to the required technology to intercept the transaction and alter it. Due to the fact that there is no encryption or integrity checking on the message during transmission through the GSM backbone, this attack is possible. The attackers can easily alter the amount to be paid, or even the account to which the money is to be paid. Figure 5.2 depicts this attack.

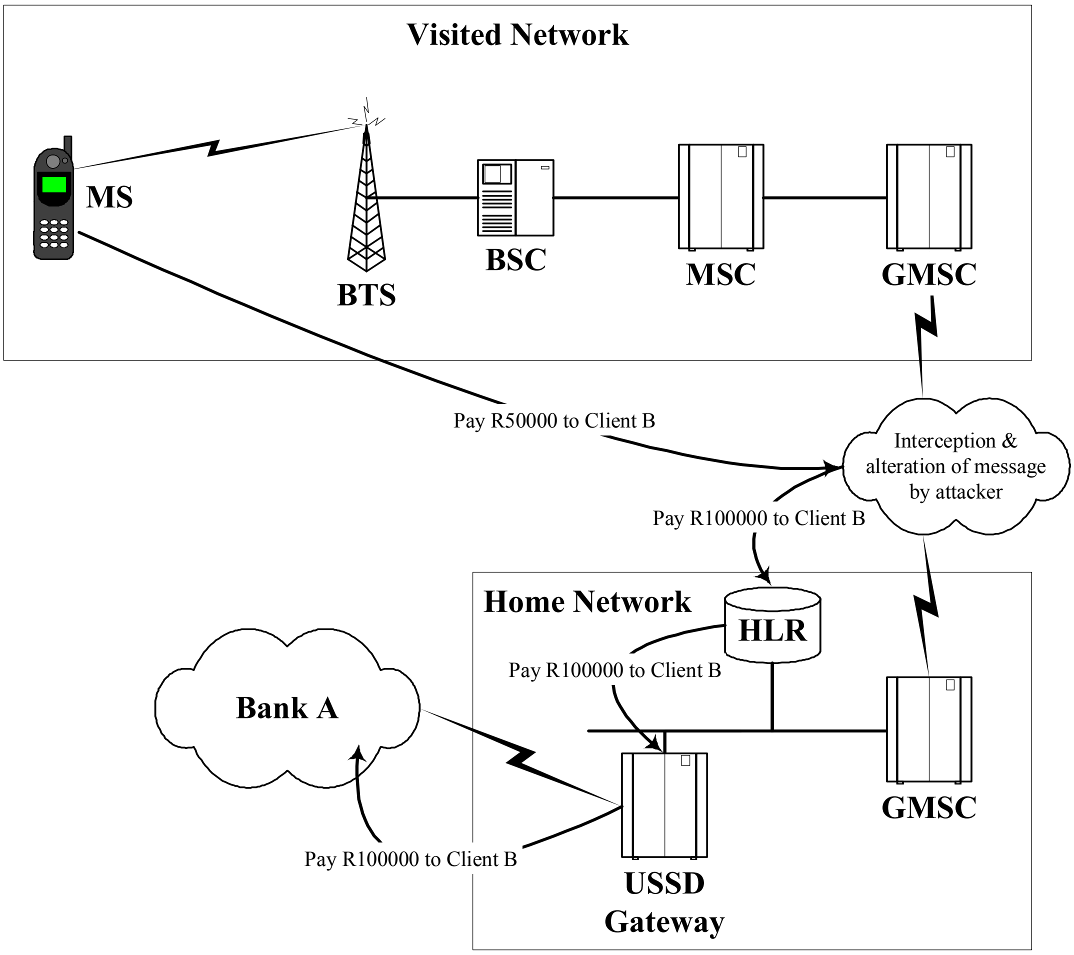


Figure 5.2: Interception and Alteration of USSD Message

From the above scenario it is evident that some measure of encryption or message integrity checking is required in order to provide a reasonably secure USSD based payment application. USSD cannot provide this service on its own. Another application or technology is required in order to secure this channel.

5.3 SHORT MESSAGE SERVICE (SMS)

5.3.1 Introduction to SMS

SMS appeared on the wireless scene in 1991 in Europe and the GSM standards included short messaging services from the outset. SMS provides a mechanism for transmitting short messages to and from wireless handsets. The service makes use of a short message service center (SMSC), which acts as a store-and-forward system for short messages. The wireless network provides for the transport of short messages between the SMSCs and wireless handsets. In contrast to existing text message transmission services such as alphanumeric paging, the service elements are designed to provide guaranteed delivery of text messages to the destination [37].

A distinguishing characteristic of the service is that an active mobile handset is able to receive or submit a short message at any time, independent of whether or not a voice or data call is in progress. SMS also guarantees delivery of the short message by the network. Temporary failures are identified, and the short message is stored in the network until the destination becomes available [38].

SMS is characterized by out-of-band packet delivery and low-bandwidth message transfer. Initial applications of SMS focused on eliminating alphanumeric pagers by permitting two-way general-purpose messaging and notification services, primarily for voice mail. As technology and networks matured, a variety of services were introduced, including electronic mail and fax integration, paging integration, interactive banking, and information services such as stock quotes [38].

5.3.2 SMS Architecture

Figure 5.3 graphically illustrates the SMS architecture as described in the GSM specifications. A brief description of each element in the SMS architecture follows.

5.3.2.1 Short Messaging Entities

Short messaging entity (SME) is an entity that may receive or send short messages. The SME may be located in the fixed network, a mobile station, or another service center [38].

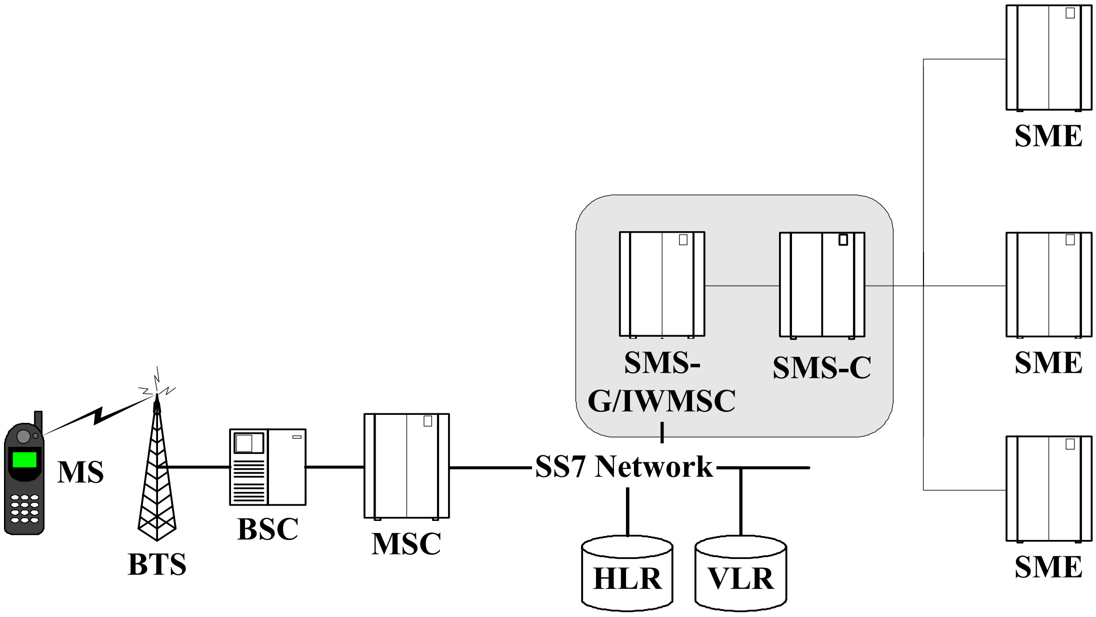


Figure 5.3: The SMS Architecture

5.3.2.2 Short Message Service Center

Short Message Service Center (SMS-C) is responsible for relaying, storing and forwarding of a short message between an SME and mobile station [38].

5.3.2.3 SMS-Gateway/ Interworking Mobile Switching Center

The SMS Gateway Mobile Switching Center (SMS–GMSC) is an MSC capable of receiving a short message from an SMSC, interrogating a HLR for routing information, and delivering the short message to the visited MSC of the recipient mobile station. The SMS Interworking MSC (SMS–IWMSC) is an MSC capable of receiving a short message from the mobile network and submitting it to the appropriate SMSC. The SMS–GMSC/ SMS–IWMSC are typically integrated with the SMS-C [38].

5.3.2.4 Home Location Register

Upon interrogation by the SMS-C, the HLR provides the routing information for the indicated subscriber. The HLR also keeps previously initiated unsuccessful short message delivery attempts to a specific mobile station. If a previously unreachable mobile station is now recognized by the mobile network to be accessible, that HLR informs the SMS-C of the fact so that the SMS-C can retry the delivery of the undelivered SMSs [38].

5.3.3 SMS Operation

5.3.3.1 SMS Signalling

The Mobile Application Part (MAP) layer defines the operations necessary to support SMS [39]. Both American and international standards bodies have defined a MAP layer using the services of the SS7 transaction capabilities part. The following basic MAP operations are necessary to provide the end-to-end short message service:

- Routing information request [37]:

  Before attempting short message delivery, the SMS-C must retrieve routing information to determine the serving MSC for the mobile station at the time of the delivery attempt. This is done by way of an interrogation of the HLR, which is accomplished via the use of the sendRoutingInfoForShortMsg mechanism.

- Point-to-point short message delivery [39]:

  The mechanism provides a means for the SMS-C to transfer a short message to the MSC that serves the addressed mobile station and attempts to deliver a message to an MS whenever the MS is registered, even when the MS is engaged in a voice or data call. The short message delivery operation provides a confirmed delivery service. The operation works in tandem with the base-station subsystem while the message is being forwarded from the MSC to the MS. The outcome of the operation thus comprises either successful delivery to the mobile, or failure caused by one of several possible reasons. The point-to-point short message delivery is accomplished via the use of the forwardShortMessage mechanism.

- Short message waiting indication [39]:

  The operation is activated when a short message delivery attempt by the SMS-C fails due to a temporary failure and provides a means for the SMS-C to request the HLR to add an SMS-C address to the list of SMS-Cs to be informed when the indicated mobile station becomes accessible. This short-message waiting indication is realized via the use of the set message waiting data mechanism.

- Service center alert [39][40]:

  The operation provides a means for the HLR to inform the SMS-C, which has previously initiated unsuccessful short message delivery attempts to a specific mobile station, that the mobile station is now recognized by the mobile network to be accessible. This service-center alert is accomplished via the use of the alert service-center mechanism.

5.3.3.2 SMS Message flow

SMS comprises two basic point-to-point services [38][39].

- Mobile Originated - Short Message (MO–SM)
- Mobile Terminated - Short Message (MT–SM)

MO–SMs are transported from the handset to the SMS-C and can be destined to other mobile subscribers or for subscribers on fixed networks such as paging networks or electronic mail networks. MT–SMs are transported from the SMS-C to the handset and can be submitted to the SMS-C by other mobile subscribers via MO–SM or other sources such as voice-mail systems, paging networks, or operators [38][39].

For MT–SM, a report is always returned to the SMS-C either confirming the short message delivery to the handset or informing the SMS-C of the short message delivery failure and identifying the reason for failure. For MO–SM a report is always returned to the handset either confirming the short message delivery to the SMS-C or informing the handset of the encountered failure and identifying the reason [38][39].

Figure 5.4 describes the signaling information and data flow for a Mobile Terminated SMS. The message flow is described by [38][39]:

1. The short message is submitted from the SME to the SMSC.
2. After completing its internal processing, the SMS-C interrogates the HLR and receives the routing information for the mobile subscriber.

3. The SMS-C sends the short message to the MSC using the forwardShortMessage operation.
4. The MSC retrieves the subscriber information from the VLR. This operation may include an authentication procedure.
5. The MSC transfers the short message to the MS.
6. The MSC returns to the SMS-C the outcome of the forwardShortMessage operation.
7. If requested by the SME, the SMS-C returns a status report indicating delivery of the short message.

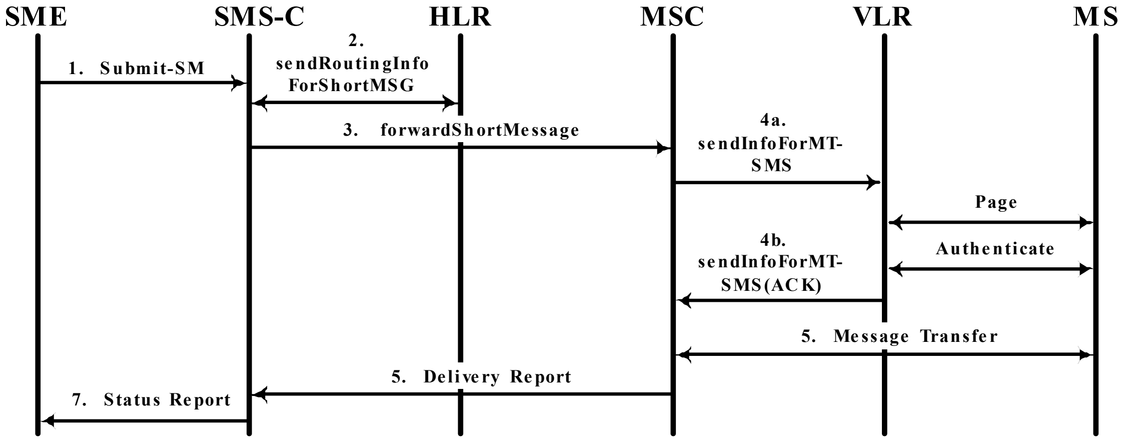


Figure 5.4: Mobile Terminated Short Message

Figure 5.5 describes the signaling information and data flow for a Mobile Originated SMS. The message flow is described by [38][39]:

1. The MS transfers the SM to the MSC.
2. The MSC interrogates the VLR to verify that the message transfer does not violate the supplementary services invoked or the restrictions imposed.
3. The MSC sends the short message to the SMSC using the forwardShortMessage operation.
4. The SMSC delivers the short message to the SME.
5. The SMSC acknowledges to the MSC the successful outcome of the forwardShortMessage operation.
6. The MSC returns to the MS the outcome of the MO–SM operation.

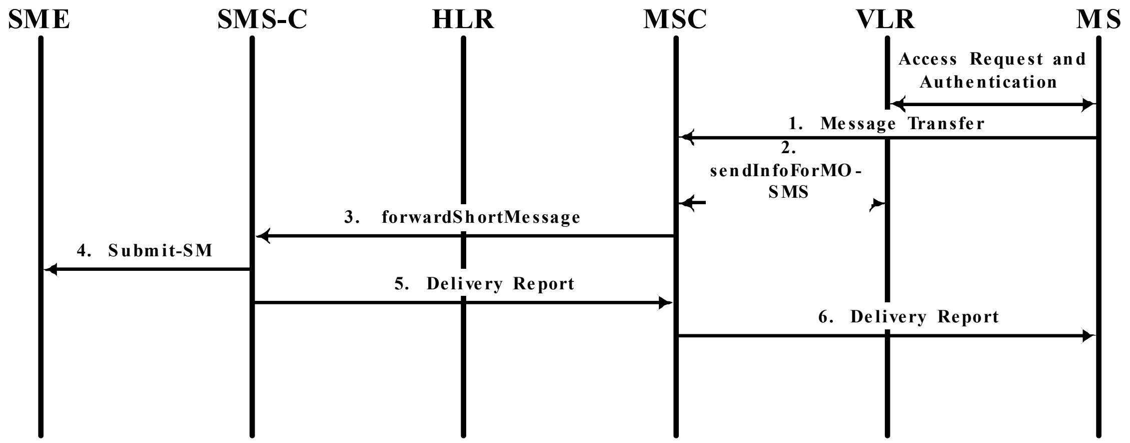


Figure 5.5: Mobile Originated SMS

5.3.4 SMS Security

In the previous sections we described the Short Message Service characteristics. SMS is a useful technology for transmitting information to large numbers of recipients in a cost effective way. However, SMS is not secure enough for carrying financial transactions. The same reasoning applies to SMS that applied to the USSD channel described in Section 5.2. There is no form of encryption or message integrity checking on the SMS message whilst traversing the GSM backbone. The situation is worsened by the fact that SMS is a store and forward application and that SMS messages are stored on the SMS-C in clear. If an attacker gains access to the SMS-C, he can alter any SMS message. This means that an attacker can locate a SMS with a payment instruction and alter any part of the message with no one being the wiser.

Standard SMS technology can be used in other value-adding applications in the banking and financial arena to reduce associated risk. When used as a medium to provide real-time transaction history on a clients account, a financial institution can reduce its fraud risk. A bank might employ a SMS service to send SMS messages to its Credit Card holders each time a transaction is conducted on their Credit Card account. The cardholders will then instantaneously, or in a relative short period of time know that his Credit Card is being used for purchases not affected by him.

This enables the cardholder to let the issuing bank know to cease all payment on his card, as he is not effecting the transactions. This reduces the attacker's window of opportunity for conducting fraud, and thus reduces the risk to the issuing bank as well as the cardholder.

5.4 THE SIM APPLICATION TOOLKIT (SAT)

5.4.1 Introduction

The SIM Application Toolkit (SAT), also referred to as SIM Toolkit (STK), is today mainly used as a tool that enables an operator-controlled menu for SMS and voice services. This allows operators to create specific applications resident on the subscribers SIM. It is also used for more advanced services that require high security, where the SIM plays a natural role as a Smart Card. The SIM Toolkit is, just as SMS, a well-proven GSM standard that has been out on the market since 1995. It has by now been incorporated into all major mobile telecommunications standards. Just like SMS it experienced a slow up-take in the beginning, partly as the market has been awaiting newer, more 'hyped' technologies.

SIM Toolkit technologies enhance the ease of deployment of mobile services. In practice, as the Toolkit is supported by all phase 2+ compliant handsets on the market, it is reasonable to believe that we will see a massive increase in services utilizing SIM Toolkit functionality.

The fact that the operator controls the SIM makes it an ideal platform for operator-provided services. The major drivers for implementing services using SIM Toolkit are the combination of its maturity, and its network technology independence as it is now incorporated in 3G and TDMA standards (GAIT). An application developed using SIM Toolkit will work in the 3G networks as well as when roaming into a foreign 2G network.

SIM Toolkit should be used for user-oriented services based on short transactions of the 'request – response' type and for implementing advanced SIM based functionality as a complement to a service developed using another browsing channel, e.g., the Web. SIM Toolkit is also ideal for server-initiated transactions, as the main data bearer is SMS. This makes it perfect for Internet services where the handset is only one of the devices, e.g., using the handset and the SIM for authentication of users and confirmation of payments while using a PC as the browsing device.

SAT services have had moderate take-off due to interoperability issues between different SIM vendors. However perhaps even more important is that there hasn't been any standardized application language developed for the communication between the SIM and the Server component. A dedicated Client and Server component had to be developed for each service, thus making it the traditional Client/Server technology. As such the SAT has the same drawbacks as the Client/Server technology that was popular in the eighties before the introduction of HTTP and HTML. It generates a good business for the companies offering professional services to build the solutions while the customers, the operators and the content providers, get stuck with a static and proprietary solution.

5.4.2 SAT Operation

Figure 5.1 shows the entities involved in sending a SAT message from a sending application to a receiving application. Figure 5.1 also shows the entities where the security of the message should be applied. If conforming to the standard as specified in [12], the communication channel will look like depicted in Figure 5.1 with a secure communication "tunnel" between the sending and receiving entities.

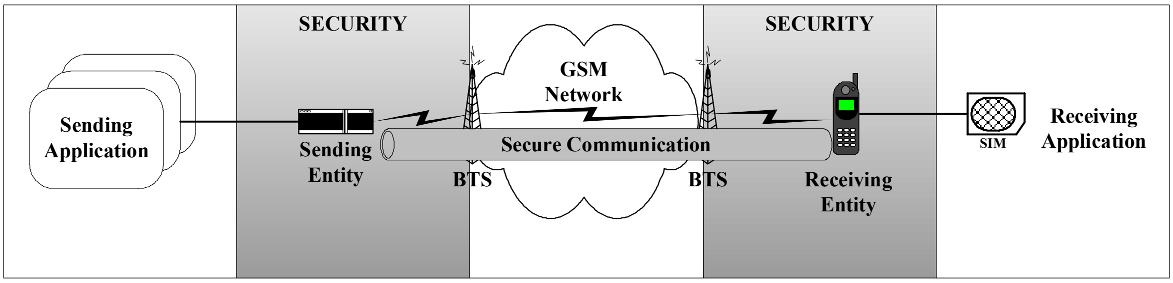


Figure 5.6: End-to-end System Overview of the SAT

The specification in [12] describes the flow of information between the sending and receiving entities and the security associated with that in the following manner and is depicted in Figure 5.7:

1. The Sending Application prepares an Application Message and forwards it to the Sending Entity, with an indication of the security to be applied to the message.
2. The Sending Entity prepends a Security Header (the Command Header) to the Application Message. It then applies the requested security to part of the Command Header and all of the Application Message, including any padding octets. The resulting structure is here referred to as the (Secured) Command Packet. Under normal circumstances the Receiving Entity receives the Command Packet and unpacks it according to the security parameters indicated in the Command Header.
3. The Receiving Entity subsequently forwards the Application Message to the Receiving Application indicating to the Receiving Application the security that was applied.
4. If so indicated in the Command Header, the Receiving Entity shall create a (Secured) Response Packet. The Response Packet consists of a Security Header (the Response Header) and optionally, application specific data supplied by the Receiving Application. Both the Response Header and the application specific data are secured using the security mechanisms indicated in the received Command Packet.
5. The Response Packet will be returned to the Sending Entity, subject to constraints in the transport layer, (e.g. timing).

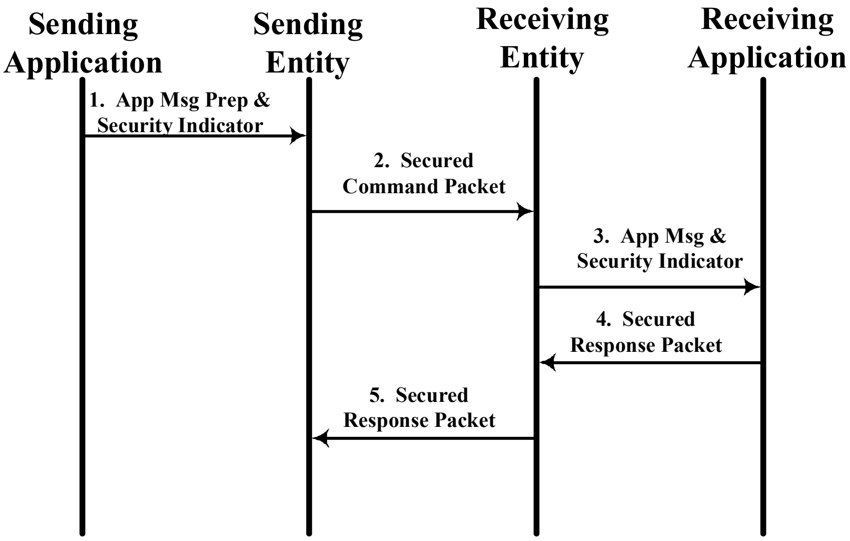

Sending Application
Sending Entity
Receiving Entity
Receiving Application
1. App Msg Prep & Security Indicator
2. Secured Command Packet
3. App Msg & Security Indicator
4. Secured Response Packet
5. Secured Response Packet

Figure 5.7: SAT Message flow. [13]

In some circumstances a security related error might be detected at the Receiving Entity. In such circumstances the Receiving Entity shall react according to the following rules [13]:

- Nothing shall be forwarded to the Receiving Application. i.e. no part of the Application Message, and no indication of the error.
- If the Sending Entity does not request a response (in the Command Header) the Receiving Entity discards the Command Packet and no further action is taken.
- If the Sending Entity does request a response and the Receiving Entity can unambiguously determine what has caused the error, the Receiving Entity shall create a Response Packet indicating the error cause. This Response Packet shall be secured according to the security indicated in the received Command Packet.
- If the Sending Entity does request a response and the Receiving Entity cannot determine what has caused the error, the Receiving Entity shall send a Response Packet indicating that an unidentified error has been detected. This Response Packet is sent without any security being applied.
- If the Receiving Entity receives an unrecognizable Command Header (e.g. an inconsistency in the Command Header), the Command Packet shall be discarded and no further action taken.

Table 5.1 shows the structure of the SAT Command Packet that is sent from the Sending Entity to the Receiving Entity.

Element	Length	Comment
Command Packet Identifier (CPI)	1 octet	Identifies that this data block is the secured Command Packet
Command Packet Length (CPL)	variable	This shall indicate the number of octets from and including the Command Header Identifier to the end of the Secured Data,including
Command Header Identifier (CHI)	1 octet	Identifies the Command Header.
Command Header Length (CHL)	variable	This shall indicate the number of octets from and including the SPI to the end of the RC/CC/DS.
Security Parameter Indicator (SPI)	2 octets	see detailed coding in section 5.1.1.
Ciphering Key Identifier (KIc)	1 octet	Key and algorithm Identifier for ciphering.
Key Identifier (KID)	1 octet	Key and algorithm Identifier for RC/CC/DS.
Toolkit Application Reference (TAR)	3 octets	Coding is application dependent.
Counter (CNTR)	5 octets	Replay detection and Sequence Integrity counter.
Padding counter (PCNTR)	1 octet	This indicates the number of padding octets at the end of the secured data.
Redundancy Check (RC), Cryptographic Checksum (CC) or Digital Signature (DS)	variable	Length depends on the algorithm. A typical value is 8 octets if used, and for a DS could be 48 or more octets, the minimum should be 4 octets.
Secured Data	variable	Contains the Secured Application Message and possibly padding octets.

Table 5-1: Structure of the SAT Command Packet [13]

5.4.3 SAT Security

In [12], which is the international specification for SAT security, the authors state that there is a definite need to secure SAT related communication over the GSM network with a level of security chosen by the network operator or the application provider. No mandate is stated as to the level of encryption required, if any. Further to this, an assumption is made that the sending and receiving entities in the communication exchange are in secure environments.

This is basically equivalent to stating that there is no need to secure traffic traversing the Internet, for if we take the same point of view as the authors of [12], then when Bank A sends a file detailing client information and Credit Card details to Bank B over the Internet, we need not be concerned as both Bank A and Bank B are housed in secure environments.

According to [12], certain security requirements are to be satisfied when transferring Application Messages from the Sending Entity to the Receiving Entity. They are listed below with a brief description of why the service is needed and the mechanisms used to provide this service.

- Authentication. Needed to authenticate the subscriber. This function is performed by the subscribers unique SIM.
- Message integrity. Needed to ensure that no corruption of the content of the message has occurred during transmission, be it accidental or intentional. This can be achieved by adding a Redundancy Check, Cryptographic Checksum, or Digital Signature to the security header.
- Replay detection and sequence integrity. Needed to protect the Receiving Entity against replay attack and Secured Packet Duplication. This is implemented by adding a counter in the calculation of the cryptographic checksum in the Security Header.
- Proof of receipt and proof of execution. Proves to the Sending Entity that the Receiving Entity has correctly received a secured packet, has performed the necessary security checks, and forwarded the contents of the secured packet to the receiving application. Proof of receipt is returned from the Receiving Entity in an acknowledgement to a secured packet transmitted by the Sending Entity. The acknowledgement takes the form of a Status Code in a response message.

- Message confidentiality. Needed to provide proof that the information contained in the messages exchanged is not disclosed to an unauthorised individual, entity, or process. This is achieved by encrypting the message with a block cipher.

They also specified the following assumptions that govern the mechanisms to satisfy the above requirements [12]:

- Security should be provided for each Secured Packet transmitted. An Application Message may be broken into several Secured Packets, each of which shall have identical security mechanisms applied to it.
- There should be the ability to turn mechanisms on and off on a "per Application Message" basis, with an indication of the status transmitted with the message.
- Security related information used should be independent of that used with existing GSM network keys.
- Third party applications may have access to the Sending Entity, however this is considered to be an internal network security issue and therefore outside of the scope of this specification.

[12] Specifies the following cryptographic mechanisms in order to achieve the requirements stipulated above:

- Cryptographic Checksum of the Application Data, denoted by B1
- Digital Signature of the Application Data, denoted by B2
- Cryptographic Checksum of the Application Data and Security Header, denoted by D1
- Digital Signature of the Application Data and Security Header, denoted by D2
- Acknowledgement as Cryptographic Checksum, denoted by F1
- Acknowledgement as Digital Signature, denoted by F2
- Encryption of the Application Data and possibly a part of the Security Header, denoted by G

Table 5.2 summarizes how these cryptographic mechanisms are achieved.

Requirements	Mechanisms		
	Cryptographic Checksum	Digital Signature	Encryption
Authentication	B1	B2	
Message Integrity	B1	B2	
Replay detection and sequence integrity	D1	D2	
Proof of receipt	F1	F2	
Confidentiality			G

Table 5-2: Overview of Cryptographic Mechanisms for the SAT. [13]

The GSM specification allows for the following encryption algorithms to be used [13]:

- DES in CBD-mode.
- DES in ECB-mode.
- Triple DES in outer-CBC-mode using two different keys.
- Triple DES in outer-CBC-mode using three different keys.
- Some proprietary algorithms that needs to be known to both entities.

As can clearly be seen from the previous sections, much development has gone into the security specification for the SAT. There are however some problems, and they are not technology related. It has come to the author's attention whilst dealing with network operators in Southern Africa that many of them have not implemented the requirement as stipulated in [12] and [13]. The main reason for this is due to the costs involved in implementing these solutions.

Many of these operators had networks up and running prior to the introduction of the SIM Toolkit and its security requirements. This means that they had to implement some network changes in order to cater for these security requirements. Many of them just implemented the basic changes needed to comply with the specification and the security issues were cast by the wayside. As a result many operators cannot handle any message integrity checks on these SAT messages, which are pure SMS messages in essence, sent between the Sending and Receiving Entities.

For any financial institution this poses a serious problem. The same attack that was described in Section 5.2 and 5.3 still applies. Without a cryptographic message integrity check, there is no way to be sure that the message reaching the financial institution is actually the message their client intended to send. Even if some operators offer the encryption of the user's PIN for the m-Banking application, the attack is still possible. In order for us to explain this, let us once again look at an example:

Figure 5.8 shows a linear representation of the SAT Command Packet structure as described in Table 5.1. This is the makeup of the SMS send to the Receiving Entity during communication.

CPI	CPL	CHI	CHL	SPI	KIc	KID	TAR	CNTR	PCNTR	RC/CC/ DS	Secured DATA

Figure 5.8: The SAT Command Packet Structure. [13]

The structure of the data section in a typical payment application is very easy to decode and would most probably look something like this:

085123456711 2AB5120123412341234500000857654321, where

- 0851234567 = the mobile number from which the transaction originated.
- 1 = Account Indicator.
- 12AB = encrypted PIN in Hex.
- 5120123412341234 = the account number the payment should go to.
- 50000 = the amount of the transaction.
- 0857654321 = the mobile number the confirmation message should be sent too.

Without message integrity checking, an attacker could easily alter the message to look something like this (the field descriptions stay unchanged):

0851234567112AB5120111111111111500000852222222, with no one being any wiser.

In conclusion, it is very important to note that one cannot implicitly trust a network operator to implement the security required to utilise a channel for m-Commerce applications.

5.5 WIRELESS INTERNET GATEWAY (WIG) TECHNOLOGY

5.5.1 Introduction

The Wireless Internet Browser (WIB) technology was introduced to ease the development of SIM-based services using a generic SIM supplier independent interpreter on the card for any kind of application. The WIB optimizes the utilization of SIM memory in addition to offering a true interoperable execution environment on the SIM. It also solves the client/server problem since it uses standard web technologies. The WIB is contained within a 32k SIM card resident in a GSM phase 2+ compliant handsets.

The WIB operates together with a network component called Wireless Internet Gateway (WIG). The WIG opens up a channel to the WIB on the SIM. The WIG enables the usage of an easy to use application language, called the Wireless Markup Language (WML), for implementing SAT based mobile services. These messages are carried over the USSD or SMS channels. The channel of choice is however the SMS channel.

WIB is a specification for a SIM Toolkit interpreter, or browser, that is licensed free of charge to companies developing software for SIM cards. It has been in commercial use since the beginning of 1999 in various wireless applications, mostly in the m-Commerce area. In July 2001 there were approximately 25 million SIM cards on the market with the WIB enabled. Since mid 2000 there was a standardization initiative, the USAT Interpreter, within 3GPP to standardize the concept.

With the WIB/WIG it is possible to implement ease-of-use services to the operator's installed base of mobile phones, and still be compliant with future technologies. With the WIB as a standard application in ROM from all SIM vendors all previous problems with SAT can be circumvented. SIM vendors have previously pushed for proprietary implementations with proprietary and difficult interfaces for service implementations. The

WIB together with its corresponding delivery platform, WIG, drastically changes this as it provides one generic Internet based interface for service creation, independent of SIM suppliers.

5.5.2 WIG Architecture

The WIG Architecture in its simplest form is very basic in nature. A subscriber making use of the WIB sends messages via SMS to the WIG. The WIG converts these WML messages to a web based protocol like HTTP, or HTTPS if security is required, and forwards them to the content provider. The WIG Architecture is depicted in Figure 5.9.

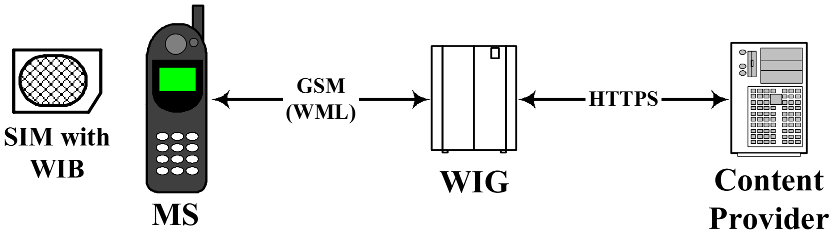


Figure 5.9: The WIG Architecture

5.5.3 Operation of WIG

WIG operates in conjunction with the SAT. The user browses a SAT menu and supplies the relevant information that is required to complete the transaction. Figure 5.10 illustrates a typical user interface and transaction flow on a mobile phone.

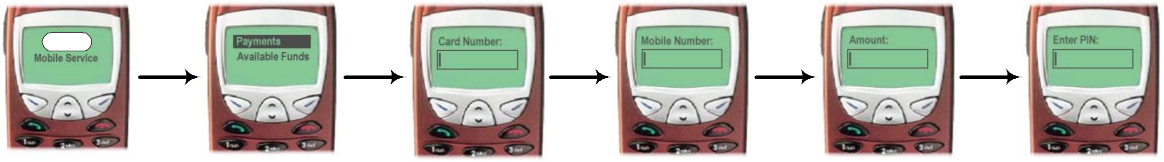


Figure 5.10: Example of a WIB payment menu on a GSM phone

Once the SAT application has all the information it requires to complete a transaction, it forwards the relevant information for processing to the WIG server. Figure 5.11 illustrates the transaction flow, followed by a brief description of each transaction.

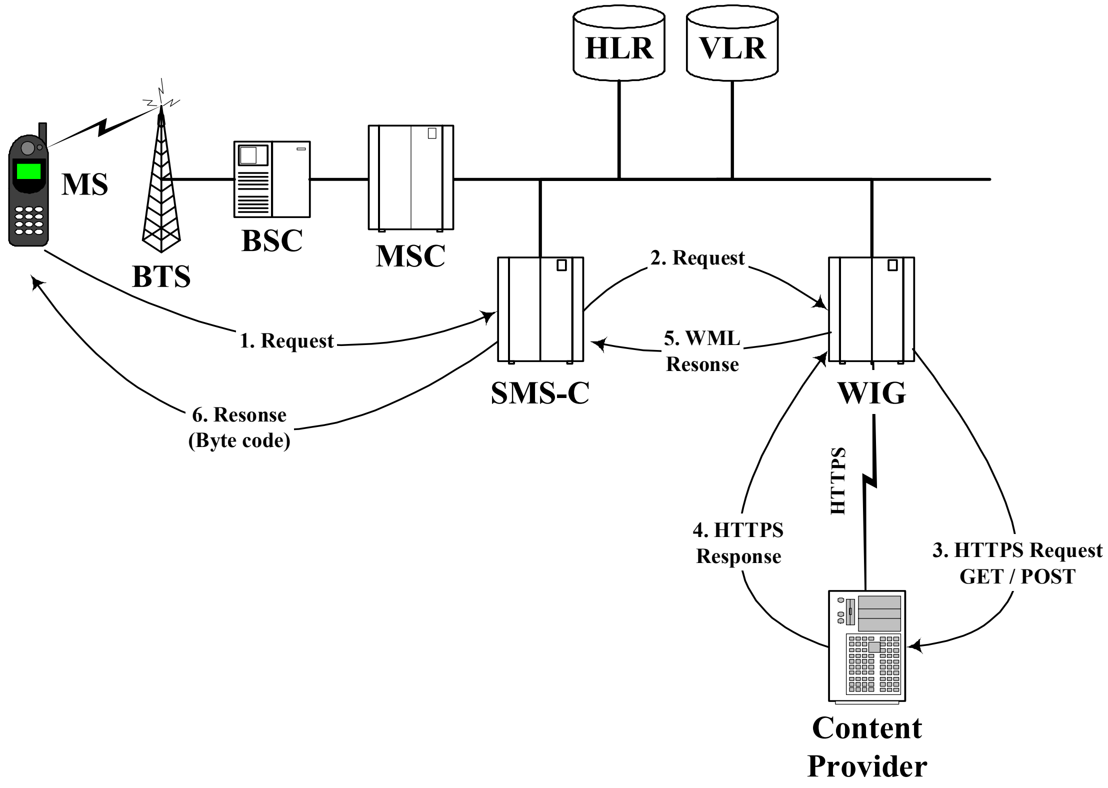


Figure 5.11: WIG Operation

The sequence of events during WIG operation is (Figure 5.11):

1. The Wireless Internet Browser (WIB) makes an URL request and sends it to the SMS-C via SMS.
2. The WIG Server receives the request from the SMS-C.
3. The WIG Server translates it into an HTTP GET or POST request and sends it to the Content Provider.
4. The Content Provider processes the request and sends an HTTP Response back to the WIG Server.
5. The WIG server passes the HTTP response and compresses the WML document into byte code and forwards the response to the SMS-C.
6. The SMS-C forwards the response to the handset. The WIB receives the sequence of commands in byte code from the WIG server and runs these commands. The WIB will use SIM Application Toolkit for user interactive commands.

5.5.4 Security of WIG

WIG in itself does not offer any additional security features apart from those offered by the technologies discussed in preceding sections. WIG is merely a standard that allows any handset with a SAT application loaded, to interact with a content provider.

CHAPTER 6: SECURE M-COMMERCE BASED ON WIG

6.1 INTRODUCTION

In the previous chapters various technologies that can provide the tools to power m-Commerce applications were discussed. In this section these elements will be combined to present a model that will provide a secure m-Commerce application that relies on WIG technology. This model is a culmination of practical industrial experience gained by the author of this dissertation; technology advances in the GSM space and basic Information Security principles.

It is of cardinal importance for any financial institution wishing to utilise new technologies to afford their customers with an enhanced capability, to ensure some basic security concepts. Some of these concepts include:

1. Authentication of the client and the Financial Institution. Making sure that the person initiating the transaction is actually authorised to do so, and that he is communicating with the institution he intends to.
2. Confidentiality of client information. Ensuring that a client's Credit Card details or home address does not get exposed to unauthorised persons, organisations or applications.
3. Integrity of payment instructions. Ensuring that the payment instruction received from a client is actually the instruction he wishes to have processed, and not an instruction altered during transmission.
4. Non-repudiation. Ensuring that the client cannot deny ever sending the instruction to the institution.

In this section we will look at the practical implementations of transmission security in the financial sector, with specific reference to the way organisations like VISA and MasterCard maintain integrity and confidentiality of transmissions. A model is then proposed that caters for these requirements, either by means of cryptography, or by some other means like business processes. The author makes the assumption that no entity in the end-to-end scope of the solution, except the financial institution itself, can be trusted to implement security on their behalf. This does not imply that the institution should re-invent the wheel, but rather that it should understand what the technology can guarantee and cannot be guaranteed. Where the technology cannot provide an acceptable solution, a creative alternative should be sought.

6.2 MESSAGE CONFIDENTIALITY IN THE FINANCIAL SECTOR

The following section defines the process for ensuring message integrity and confidentiality used by all major banks and Credit Card providers throughout the world.

Financial institutions make use of symmetric key encryption to secure Personal Identification Numbers (PIN), or any other information deemed sensitive enough, traversing the information networks. The scheme they employ enables an Acquiring bank to communicate with any Issuing bank in the world, making use of well defined processes for symmetric key exchanges between banks and 3rd parties. This scheme is illustrated in Figure 6.1.

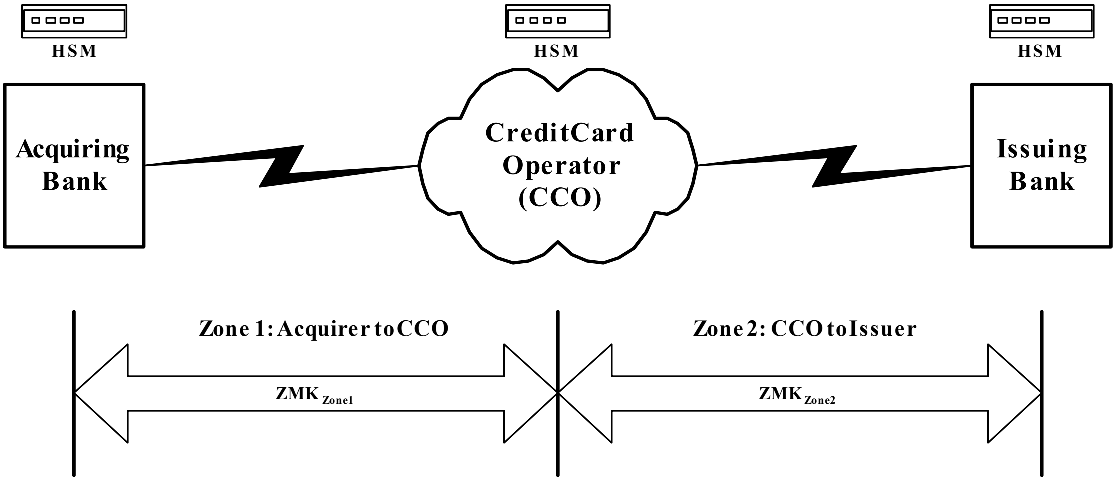


Figure 6.1: Key exchange architecture

The Acquiring and Issuing institutions exchange a Triple DES Zone Master Key (ZMK) or Key Encrypting Key (KEK), also called a Zone Control Master Key (ZCMK), with the Credit Card Operator (CCO), like VISA or MasterCard. This key exchange is done via a paper based method, with three different key components of the key being couriered to three different individuals within the CCO. This ZMK is then stored in an encrypted format via a machine specifically designed for banking encryption. This machine is called a Host Security Module, or a Hardware Security Module (HSM). These machines comply with the strictest security standards and are Federal Information Processing Standard (FIPS) 140-1 Level 3 certified. The ZMK is used to encrypt all subsequent key exchanges between organisations.

As an example, these organisations will also exchange a key called the Zone PIN Key (ZPK) which is used to encrypt all PINs for transmission between organisations. The ZPK is generated by either the financial institution or the CCO. It is then encrypted under the 3DES ZMK and sent electronically to the other party. It is important to note that the financial institutions do not exchange keys with each other, but merely with the CCO. The CCO keeps a database of all the banks that makes use of their services, and all the keys that the bank holds. All these key are stored encrypted under that specific bank's ZMK. They are not stored in clear text. In order for the bank and the Credit Card institutions to communicate with each other, all communicating parties have to make use of the same encryption standards and technologies. For this reason all the banks also by default posses HSM processors.

The HSM makes use of hardware encryption to translate a message from being encrypted under one key (the Acquirer), to being encrypted under another key (the Issuer). These translations take place in the hardware and under no circumstances is any part of the decrypted message visible to anyone apart from the hardware. The actual functioning of these devices falls outside of the scope of this article (The Thales e-Security RACAL module is the de-facto standard for such HSM devices. Visit http://www.thales-esecurity.com/productsservices/HSM7000.shtml for more information)

Figure 6.2 depicts this translation process graphically. (VISA is only used as an example)

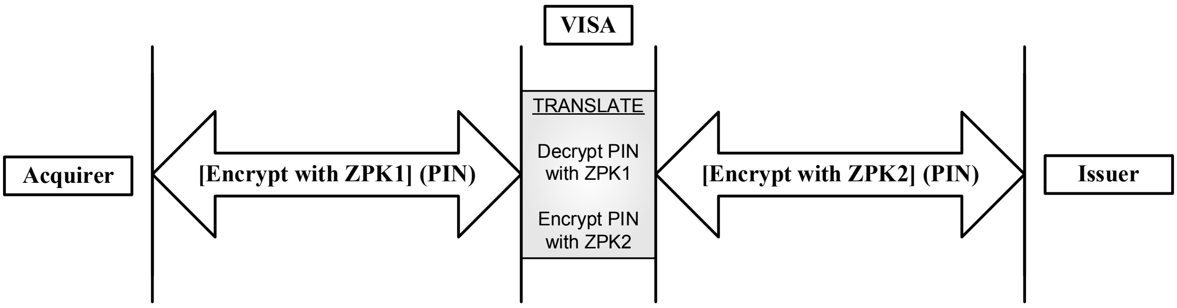


Figure 6.2: PIN Translation

However, there is one problem that has been identified. If only a PIN is encrypted, problems could arise with the use of cryptographic padding of the PIN before encryption, as PIN lengths differ. To overcome this problem, banks make use of either one of two modes of operation, defined in the international standard for PIN block generation. The ISO 9564-1 standard defines modes of PIN block generation. The most widely used is either the Format 0 PIN Block, or the Format 1 PIN Block. For the purpose of this dissertation, and the proposed solution, we will make use of the Format 1 PIN Block, although Format 0 could just as easily be used.

As an example, the ISO 9564-1 Format 1 PIN block is generated in the following manner:

For the ISO 9564-1 Format 1 PIN block, the PIN is formatted as follows:

141234RRRRRRRRRR, where:

- 1 = ISO Format indicator
- 4 = PIN Length
- 1234 = PIN
- RRRRRRRRRR = Padding with random Hex values

For 5 digit PINs it will look like:

1412345RRRRRRRRR

and for 6 digit PINs it will look like:

14123456RRRRRRRR

For transportation, the PIN Block now needs to be encrypted with the ZPK shared between the Acquirer and the CCO.

The CCO then translates the received encrypted PIN Block to being encrypted under the ZPK shared with the Issuer of the card.

By making use of this process, the banking fraternity has solved the problem of encryption of PINs between Acquiring and Issuing banks. As this process has been proven to function admirably in the real world, the authors chose to apply this architecture to the proposed m-Commerce model.

6.3 CLIENT REGISTRATION

Client registration plays an integral part in the solution. The more information that can be obtained from the client during registration, the less information about the client needs to be transmitted across the network. Ideally this should be a face-to-face meeting with the client in order to obtain all the relevant information from the client, or to insure that current client data is in fact still correct. If this meeting cannot be conducted face-to-face, the client can register via a secured web site, where after the information should be confirmed with the client via some other channel, (e.g. e-mail, telephone call, etc.)

Assuming the client is already an account holder with the institution, additional information that should be collected includes:

- The client's mobile number or any other mobile number he wishes to have access to this service.
- The account number(s) of the account(s) he wishes to use during his transactions.
- The payment limits he wishes to implement. This could be in the form a specified maximum amount to any other account, any amount payable to only certain account(s), or a combination of both. These limits might also depend on the business rules imposed by the financial institution. If they decide to only allow payments to pre-defined accounts whilst using this service, then the service is sold as such.
- Any other information the institution might feel is pertinent. Some institutions might want the client to sign an agreement that specifies that the client carry the risk associated with using the service.

Once the client is registered, his information should be stored in a database that can easily be accessed by the content server. (The security of this database should be comparable to the security of databases used in industry standard e-Commerce applications. The analysis of the security of these databases falls outside of the scope of this dissertation.) In this database, the client's mobile number should be linked to his account number, so that either can easily identify him. If the client wishes to use more that one account for these payments, each account should be assigned an identifier. This identifier can take any form, i.e. an integer number, or an alias. This will ensure that the client's account details are never transmitted across the network.

6.4 CONFIDENTIALITY OF THE CUSTOMER PIN

Each 32K SIM card distributed by the Mobile Networks in South Africa has a set of unique double-length DES keys. These keys are diversified and unique per device. In order for the financial institution to verify a PIN encrypted with one of these keys, they either need access to the key used for encrypting the PIN, or the PIN needs to be translated to a key that the financial institution knows. Usage of these keys is limited by the network operators and they are very reluctant to give one access to these keys. Despite this fact, they have a system in place for encrypting certain sensitive information from the MS to the HSM on the backend using one of these keys.

The end-to-end confidentiality of the PIN is ensured by a combination of standard financial processing and encryption by the network operator. This process is explained in Figure 3.

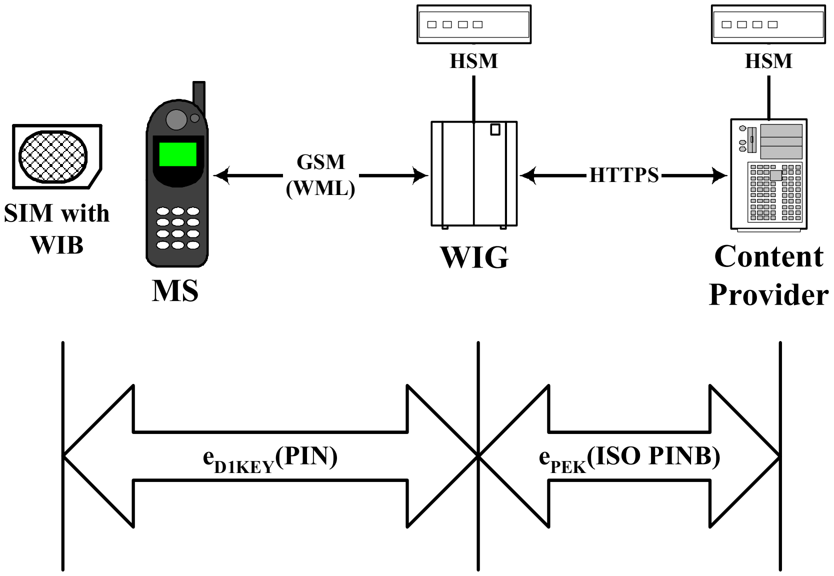


Figure 6.3: Confidentiality of the PIN

Once the client has completed the entering his PIN, the PIN is encrypted using 3DES with one of the unique keys on the SIM and sent, along with the rest of the message, to the WIG server.

Once the PIN arrives at the WIG, the HSM is called to decrypt the PIN, reformat it into an ISO 9564-1 Format 0 PIN Block and re-encrypt it under a 3DES Pin Encryption Key (PEK). All the cryptographic functions should be performed in a tamper evident hardware security module (HSM). This PEK is a key that is securely exchanged between the network operator and the financial institution.

The reformatted encrypted PIN block is then sent through to the financial institution where the PIN is verified against an encrypted PIN stored on the financial institution backend, using a one-to-one compare method. This function should also be performed within a HSM. This process should ensure the end-to-end security on the clients PIN.

6.5 AUTOHORIZATION OF THE FINANCIAL INSTITUTION

The question arises: "How does the client know that he is actually communicating with his own bank?" The answer is quite simple. By making use of a Personal Assurance Message (PAM) the client can have full comfort that he is actually communicating with his desired bank. The PAM is a phrase recorded by the financial institution upon client registration. It is a shared secret that only the bank and the client know. It can take any form, like "My dog is spot", or any other phrase that falls within the criteria specified by the financial institution. This PAM is presented on the client's mobile phone once the communication keys are passed back to him. This message which only the bank and the client knows, then assures the client that he is in fact communicating with his bank.

6.6 END-TO-END MESSAGE INTEGRITY

There are two distinct possibilities in solving the problem of message integrity in mobile payments solutions by using the current technology available to us.

The first option makes use of the same architecture as used in securing the confidentiality of the client's PIN. In this scenario, one of the derived keys resident on the SIM card is used to do a 3DES CBC MAC of the whole message, excluding the encrypted PIN. This message is then packed into the SMS and sent to the WIG. Once the SMS arrives at the WIG, the application strips the PIN from the SMS and verifies the MAC on the remainder of the message. If the MAC verified correctly, the WIG application translates the PIN into the ISO format PIN Block. The WIG application then recalculates the MAC on the entire message, including the encrypted PIN Block, using the Message Authentication Key

(MAK) that was securely exchanged with the financial institution. This new message is then sent to the financial institution, which verifies the MAC and the client PIN before acting on the payment instruction contained within the SMS. This architecture is depicted in Figure 4.

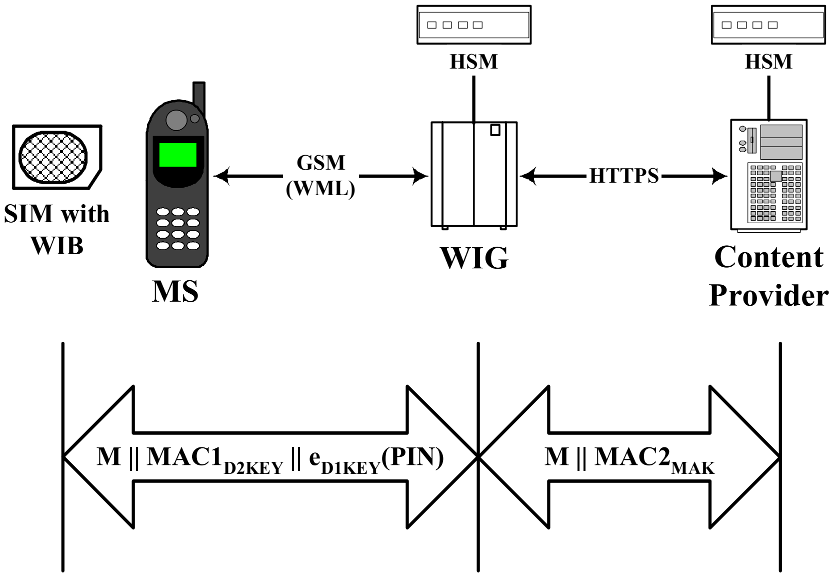


Figure 6.4: Message Integrity with MAC translate

Although this solution is possibly the easiest, it is not always possible to implement it. Some network operators have proprietary HSM modules, which cannot do this kind of translation without development work. As this is highly specialised equipment, any development work required is very costly. Network operators are reluctant to incur such costs. Some network operators do not even feel that the need to validate the integrity of the message from end-to-end is justified, as they are of the opinion that their networks are secure. Another creative alternative must therefore be sought.

The second option entails the use of a unique message authentication key per transaction. In this model, the financial institution and the network operator establish a secure Zone Master Key, also called a Key Encryption Key, between the two HSM modules. The application resident on the client's handset now has to cater for an online "registration" process before the actual payment application can take place.

The client application sends a SMS to the WIG server indicating that the client is ready to perform a payment transaction. The WIG then send the request to the financial institution's application server which in turns requests the financial institution's HSM to generate a random session message authentication key (SMAK) encrypted under the secure ZMK exchanged previously. The financial application server forwards the encrypted key to the WIG who asks its HSM to translate the encrypted SMAK to a key resident on the subscribers SIM card. The translated key is then sent to the SIM, where it is used to generate the MAC on the message, without the encrypted PIN, before it is packaged into the SMS.

Upon receiving the SMS, the WIG server translates the encrypted PIN into the ISO format PIN Block before sending the payment instruction to the financial institution. The network operator does not have to translate the MAC to a different key, as the financial institution has the SMAK, and can therefore verify the MAC on the payment instruction by merely removing the encrypted PIN block from the message. This provides true end-to-end message integrity verification. Figure 5 depicts this.

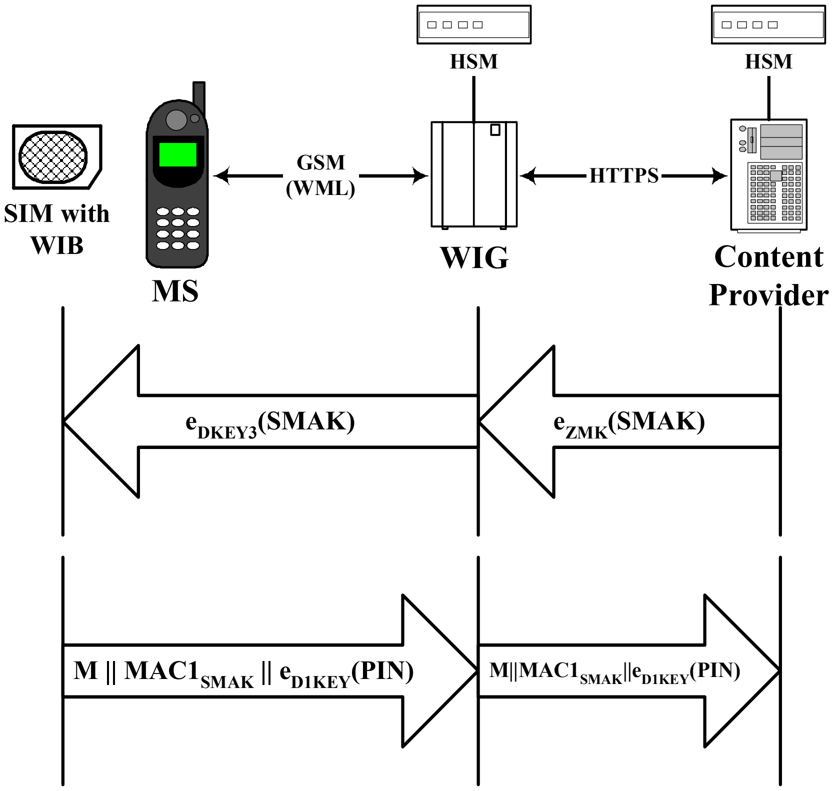


Figure 6.5: End-to-End Message Integrity

6.7 THE ARCHITECTURE OF THE PROPOSED SOLUTION

The proposed solution will have a very similar architecture to WIG, as this is the technology used. Figure 5.12 illustrates this. The only difference here is that the Content Provider is indicated as a cloud.

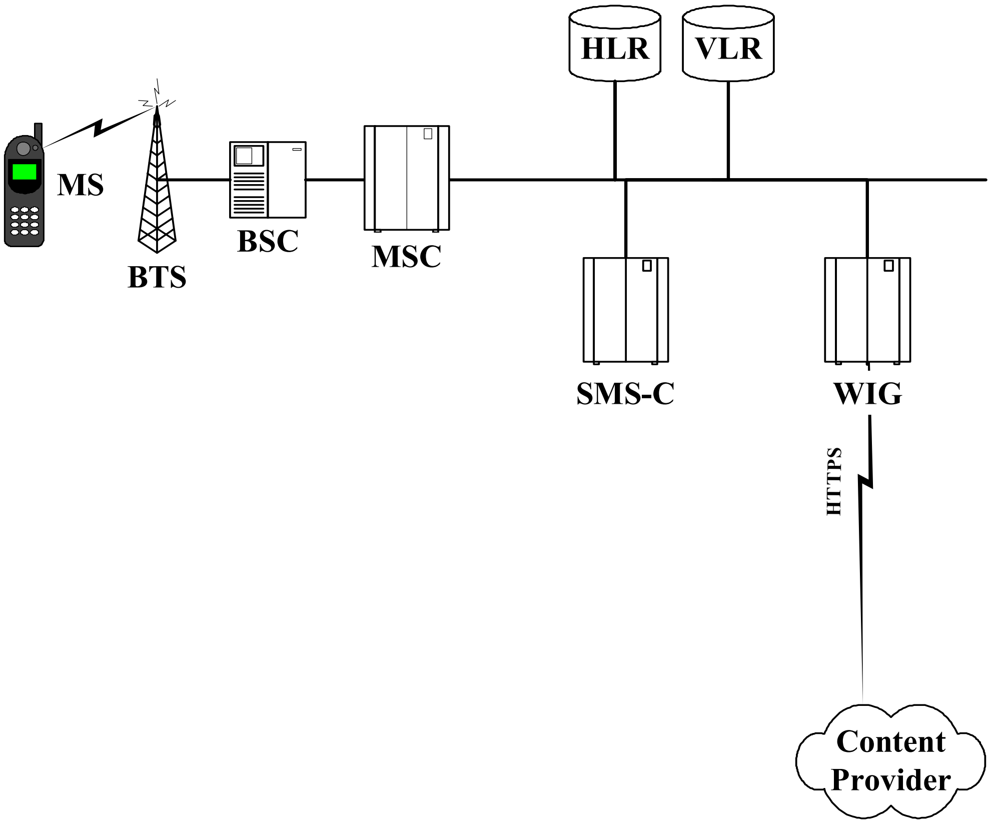


Figure 6.6: Architecture of the Proposed Solution

6.8 TRANSACTIONAL FLOW

The solution is best described by an end-to-end transactional flow. Figure 5.13 shows the flow. Each step is discussed in sequential order.

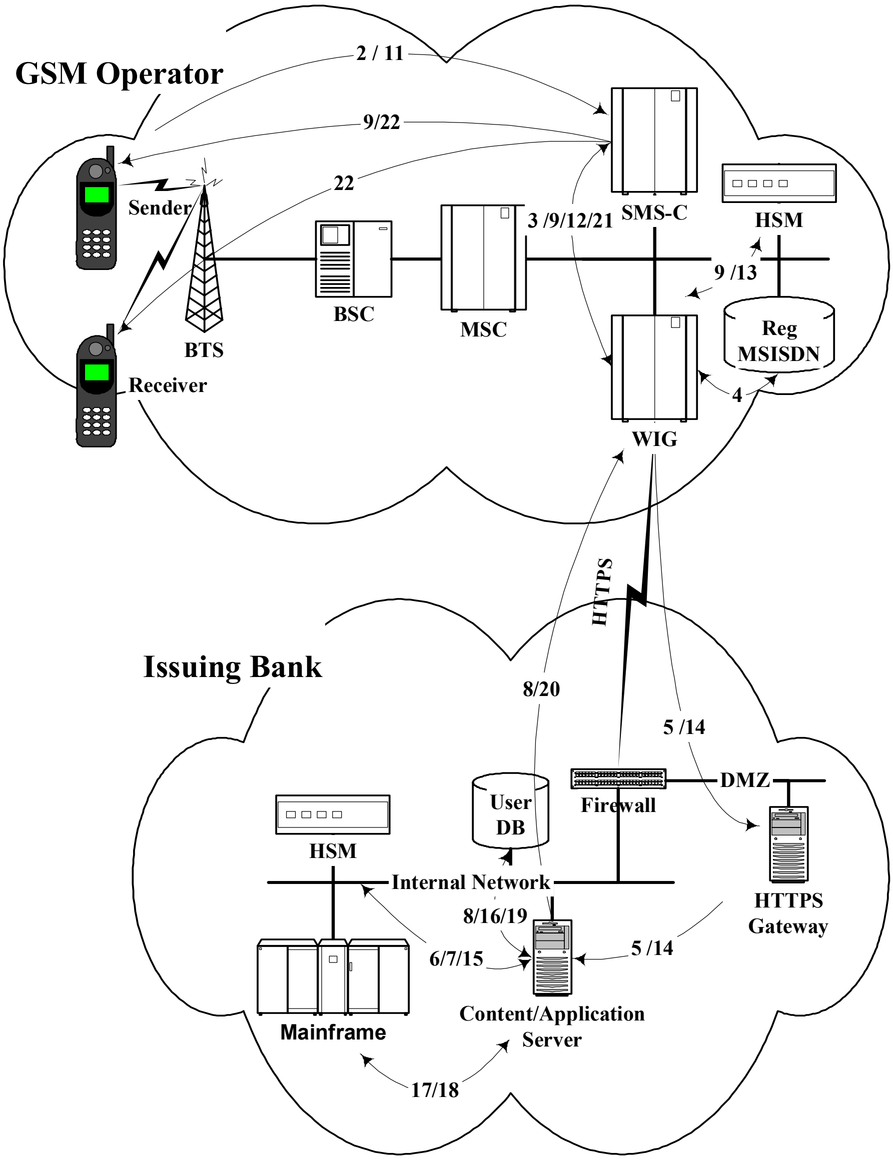


Figure 6.7: Solution Transaction Flow

The flow of a typical transaction in the proposed model would be:

1. Client selects the WIG banking application on the phone menu and selects the initialize application.
2. The message is sent via SMS to the SMS-C.
3. The SMS-C forwards the message to the WIG server.

4. On receiving the message the WIG gateway verifies the MSISDN is registered for the application.
5. If the client is registered, the WIG server sends the MSISDN of the user to the Application Server at the bank via the HTTPS gateway, and requests a random Message Authentication Key (MAK) from the Application Server which will be used to compute a MAC on the transaction message.
6. The Application Server makes a call to its HSM requesting a MAK.
7. The HSM responds to the Application Server with the MAK encrypted under the ZMK.
8. The Application Server retrieves the user's PAM and forwards it and the encrypted MAK to the WIG server via HTTPS.
9. The WIG server decrypts the MAK and sends it and the PAM to the handset via an instruction message.
10. Upon receipt of the instruction message, the SAT application launches the WIB and displays the PAM to the client. The client is prompted to verify the PAM. Upon successful verification of the PAM, the SAT prompts the user to enter the required information, i.e. from account indicator, to account number, amount, and the mobile number where the confirmation message should be sent. After displaying a confirmation message, the user is required to enter his PIN. The PIN gets 3DES encrypted with the unique derived keys resident on the SIM card. The SIM constructs the message to be sent, appends a timestamp and performs a DES MAC on the whole message, excluding the encrypted PIN and the timestamp.
11. The handset forwards the message to the SMS-C
12. The SMS-C forwards the message to the WIG server.
13. The WIG server, making use its the HSM, translate the encrypted PIN to an ISO 9564-1 Format 0 PIN block.
14. The WIG appends the PIN block and the users MSISDN to the original message, excluding the 3DES encrypted PIN, and MACs the message with the ZAK shared with the bank. The WIG server sends the message to the Application Server via the SSL link.
15. Upon receipt of the message, the Application Server verifies the MAC on the message from the WIG. It also verifies the client's PIN using the PIN block, and verifies the MAC of the original SMS.
16. If all verifications succeed, the Application Server retrieves the client's account number from the user database, and formulates the payment instruction to be sent to the mainframe.

17. The Application Server sends the payment instruction to the mainframe.
18. The mainframe processes the transaction and sends a response code to the application server.
19. The Application Server MACs the response code by using the ZAK.
20. The application server sends the MAC as a confirmation code, along with details of the transaction to the WIG indication the MSISDN numbers that should receive the confirmation message.
21. The WIG forwards the confirmation message to the SMS-C.
22. The SMS-C forwards the confirmation message, as a normal SMS to the intended recipients.

In this propose model, data integrity is kept by means of a MAC from the handset to the issuing bank. This kind of end-to-end integrity checking is required in order to secure this channel for m-Commerce applications. The confidentiality of data in this instance is not so crucial, as very little information about the client is transmitted across the network. This is why client registration plays a very important role in the proposal.

CHAPTER 7: ANALYSIS OF THE PROPOSED SOLUTION

7.1 INTRODUCTION

In this chapter we will analyse the security of the proposed solution end-to-end against the following international recognised information security criteria (as described in Section 2.2):

1. Interruption
2. Interception
3. Modification
4. Fabrication
5. Confidentiality
6. Message Integrity
7. Authentication
8. Replay
9. Non-Repudiation

We will look at different attack taxonomies on a transaction flow before and after application of the proposed solution.

7.2 INTERRUPTION OF THE MESSAGE

The transaction message can be interrupted either via an attacker or a breakdown of the communication channel. This can result in unavailability of the service.

If this is intentional disruption, we refer to it as a Denial-of-Service attack. This attack is the most common attack found on the Internet. At the time of writing this dissertation, no software or hardware exists that is truly capable of preventing a Denial-of-Service attack initiated by a savvy hacker. Microsoft, Yahoo and Amazon.com are some of the better-known parties that recently suffered from just such a crippling attack.

Due to the nature of the End-to-End solution, the easiest and most likely point of such an attack would be on the interfaces between parties that are connected to the Internet. To overcome these vulnerabilities, it is suggested that normal procedures be followed in securing the IP Network between said parties. The security of the IP Network is beyond the scope of this dissertation.

If it is an unintentional interruption, then the service is plainly unavailable until technicians can restore it. This only provides discomfort to the users of the service and can result in bad publicity with the public.

Before application	**After Application**
Vulnerable to Interruption	*Vulnerable to Interruption*

7.3 INTERCEPTION OF THE MESSAGE

The message can be intercepted by an attacker and analysed for content. This could provide the attacker with valuable information about transactions taking place from a specified client. Such information could be used for extortion or industrial espionage purposes.

The most likely point of such an attack would be on the GSM network due to the simple reason that the traffic flow between the WIG server and the banking backed is encrypted via SSL. Although someone can still intercept the message here, they will not be much wiser for doing so, unless they have a lot of time to decrypt the messages via a cryptographic attack.

On the GSM network however, the messages run in the clear until they hit the WIG server. This would be my choice for a first interception attempt. The saving grace in this case is that in order to intercept these messages at this point, you either need access to the GSM network infrastructure and to be a specialist in the protocols of the GSM network, or have access to someone that does.

Before application	**After Application**
Vulnerable to Interception	*Vulnerable to Interception*

7.4 MODIFICATION OF THE MESSAGE

Before application of the proposed solution, the message is vulnerable to modification during transit. An attacker can easily alter the message without any party in the communication being aware of the change. The client assumes that the payment instruction he sent is the one the bank receives.

With application of the DES MAC on the message, the client and bank is assured against this kind of attack. The attacker will need intercept the message and MAC. Whenever the attacker modifies the message, the MAC will not verify correctly. In order for the attacker to spoof the MAC, he has to have access to the double length DES key. It would be of no value for him to try a cryptographic attack on the MAC, as these transactions are of a transactional nature and thus has a short lifetime.

Before application	**After Application**
Vulnerable to Modification	***NOT** vulnerable to Modification*

7.5 FABRICATION OF THE MESSAGE

As with Modification, the messages that are sent before the introduction of the proposed solution will be vulnerable to the Fabrication attack, as there is no form of message verification performed on it.

With application of the proposed solution however, the attacker cannot fabricate the message, as he does not have access to the double length DES key in order to perform the MAC on the message. The attacker would also require the user's PIN code in order to use the system fraudulently.

Before application	After Application
Vulnerable to Fabrication	***NOT** vulnerable to Fabrication*

7.6 CONFIDENTIALITY OF THE MESSAGE

Both the normal and proposed solutions offer some form of confidentiality over certain zones through which the message travels. These include the Over-the-Air and the IP interface. There is however no confidentiality provided while the message travels over the GSM network and the banking backbone.

Before application	After Application
Some Confidentiality provided	*Some Confidentiality provided*

7.7 MESSAGE INTEGRITY

No message integrity checking is available without applying the changes as proposed. This is due to the fact that before application of the proposed changes, there is no MAC calculated on the message before transmission. This means an attacker can alter the message without either the sending or receiving party being the wiser.

With the proposed solution we add a 3 DES MAC to the message, which gives us a full measure of message integrity checking. The attacker can no longer alter the message without someone noticing the change.

Before application	After Application
***Cannot** verify Message Integrity*	***Can** verify Message Integrity*

7.8 AUTHENTICATION OF THE MESSAGE

Before application of the proposed solution there is nothing that one can use to authenticate the message origin apart from the users MSISDN. This MSISDN can however easily be spoofed and an attacker can masquerade as a legitimate user.

After application of the proposed solution one can use the MAC to verify the message origin. This is due to the fact that only the intended originator of the message is in possession of the key used to generate the MAC. If the bank can verify the MAC as being correct, they can accept that the message originated from the intended client.

Before application	After Application
MSISDN alone to Authenticate	*MSISDN and MAC to Authenticate*

7.9 REPLAY OF THE MESSAGE

Implementing a timestamp from the handset when sending the payment instruction, can safeguard the client and the bank that the message has not been replayed. Without such a timestamp, an attacker can replay the payment instruction without the client being aware that such an attack has taken place.

We further introduced a client confirmation step in which the client has to verify the transaction and accept the payment instruction by selecting the "Correct" option displayed on his handset, and entering his PIN to authenticate that the real client actually authenticated the transaction. This ensures that the client is aware of all transactions that take place on his account via the WIG facility.

Neither of these two functions is in a normal WIG banking application, which makes them vulnerable to this kind of attack.

Before application	After Application
Vulnerable to Replay	*Not vulnerable to Replay due to Timestamp*

7.10 NON-REPUDIATION OF THE MESSAGE

Non-repudiation means that the client uses the "I did not do it-" excuse. Before applying the MAC on the message, this kind of excuse might have been a viable one, as one can alter the payment instruction during transmission of the message.

With application of the MAC on the message in the proposed solution, a client cannot repute a message sent from his handset, as his identity is verified via his Personal Identification Number (PIN), and it is insured that the message is not altered during transmission. This enables the bank to prove that the no one except the client could have sent the message, unless he gave his PIN to someone else.

Before application	**After Application**
*Client **can** repute the message sent*	*Client **cannot** repute the message sent*

7.11 SUMMARY

Below is a table that summarises the vulnerabilities to specified attacks before and after application of the proposed solution:

Attack	**Before application**	**After application**
Interruption	Y	Y
Interception	Y	Y
Modification	Y	X
Fabrication	Y	X
Confidentiality	Y	Y
Message Integrity	Y	X
Authentication	Y	X
Replay	Y	X
Non-Repudiation	Y	X

Where:	
Y =	Vulnerable to attack
X =	Not Vulnerable to attack

Table 7-1: Summary of attack vulnerability

7.12 POSSIBLE SOLUTION VULNERABILITY

7.12.1 The WAP Gap

Although Wireless Transport Layer Security (WTLS) provides us with Wireless Application Protocol (WAP) security over the wireless network, much the same as SSL does in the wired medium; a huge flaw exists in some implementations of WAP. This is commonly referred to as the WAP gap [23]. The WAP gap comes from the manner in which the WAP Gateway is implemented. Figure 7.1 shows the WAP gap.

The WAP gap highlights the issue of control over the WAP Gateway. In certain implementations of the WAP model the WAP Gateway is not under the control of the financial institution, etc. In essence, not having control over the physical and logical security of the gateway renders it an untrusted element in the end-to-end security of the transaction data.

Due to the WAP stack functionality it is not feasible to do away with the gateway. It is therefore necessary to establish the extent of the risk and alternatives to of address these risks.

Some might argue that the WAP gateway is not a security risk because the gateway vendors are aware of the issue and have taken steps to ensure that the process of decrypting from WTLS and re-encrypting into TLS cannot easily be compromised. Typical steps taken will be to ensure that the decryption and re-encryption takes place in memory, that keys and unencrypted data are never saved to disk, and that all memory used as part of the encryption and decryption process is cleared before being handed back to the operating system.

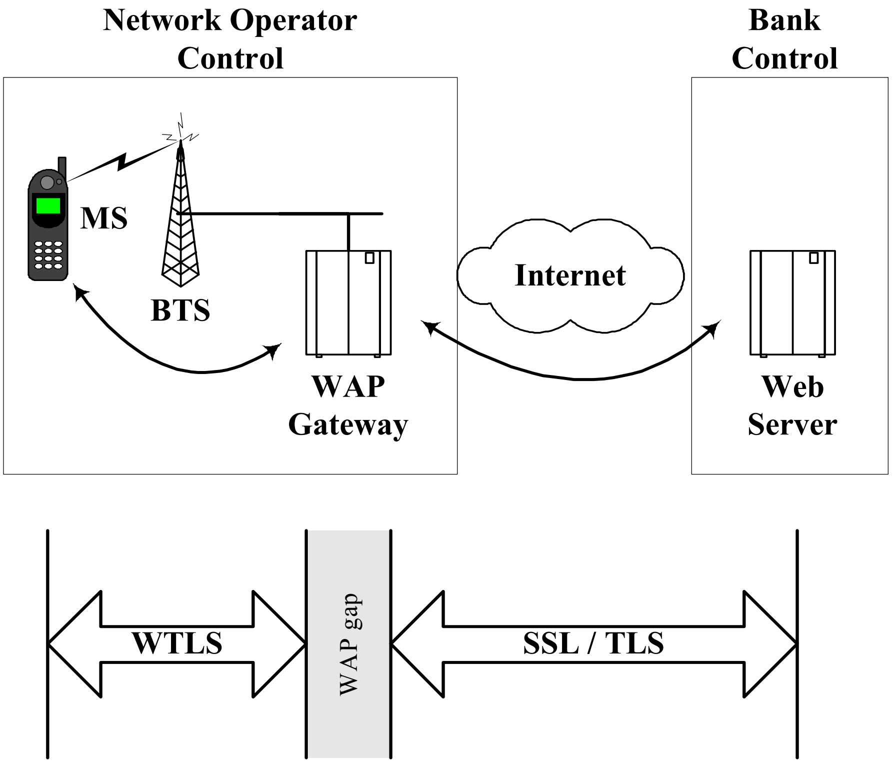


Figure 7.1: The WAP Gap

The problem with this is that there are no standards or guarantees about these precautions. There exists no way of ascertaining how robust a vendor's implementation actually is, and in the case of a gateway that is hosted by a network operator you may not even be able to tell who implemented it. The possibility exists that a vendor that can implement a WAP gateway on a very secure operating system in a thoroughly secured environment under the control of extremely competent administrators could provide a reasonably secure implementation. Even so, there is still an exposure around the gateway and at some stage, when it can become financially feasible, it will become a target for hackers [41].

With a minor change in the network architecture of Figure 6.7, we can secure WAP to such an extent that the risk becomes either manageable or acceptable to the organisation. By moving the gateway “in-house” to the financial institution, the WAP gap becomes manageable. Figure 7.2 shows this architecture [41].

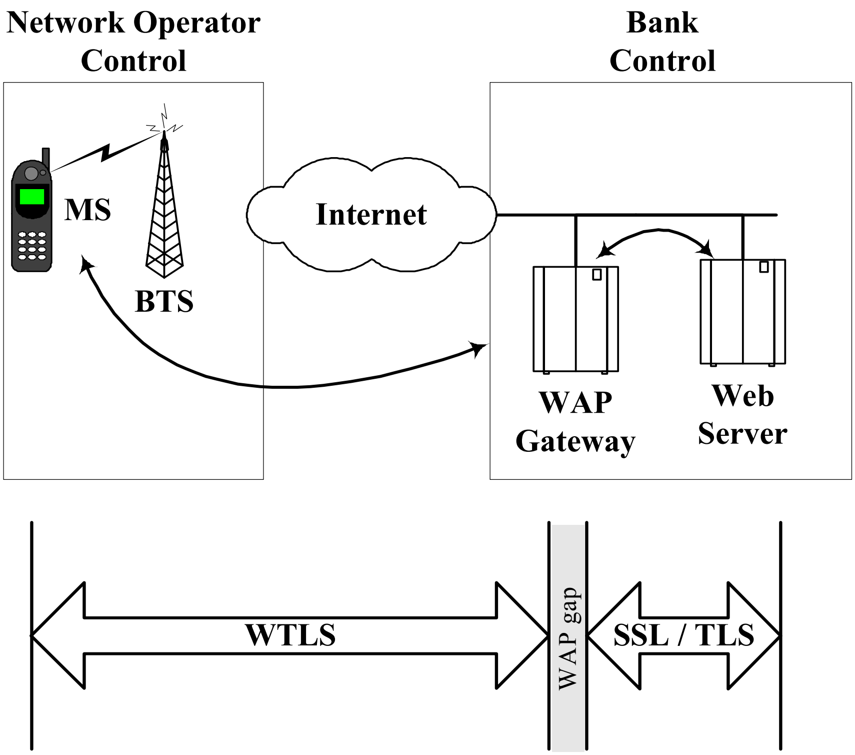


Figure 7.2: Closing the WAP Gap

7.12.2 The WAP GAP applied to WIG

In analysis of the proposed solution, the question arises: "Does the WAP Gap principal apply to the proposal?" In a WTLS secured WAP application, the WAP server has to convert the WTLS instructions received from the handset to SSL instructions in order for the Web server to understand them. This implies that the WAP server has to "translate" the WTLS encryption into SSL encryption. It would therefore seem that the same vulnerability could exist in the proposed WIG solution due to the fact that the WIG server has to translate the MAC from encryption under one key to the next. Does this not then constitute a WIG Gap? Figure 7.3 shows this graphically.

There is a difference that needs to be noted. In the WAP scenario, the "translation" takes place in the WAP server's memory, and the local resources of the WAP server is used to compute the translations in encryption schemes. In the WIG scenario, the translation takes place in an industry certified hardware encryption device as discussed in section 6.2.

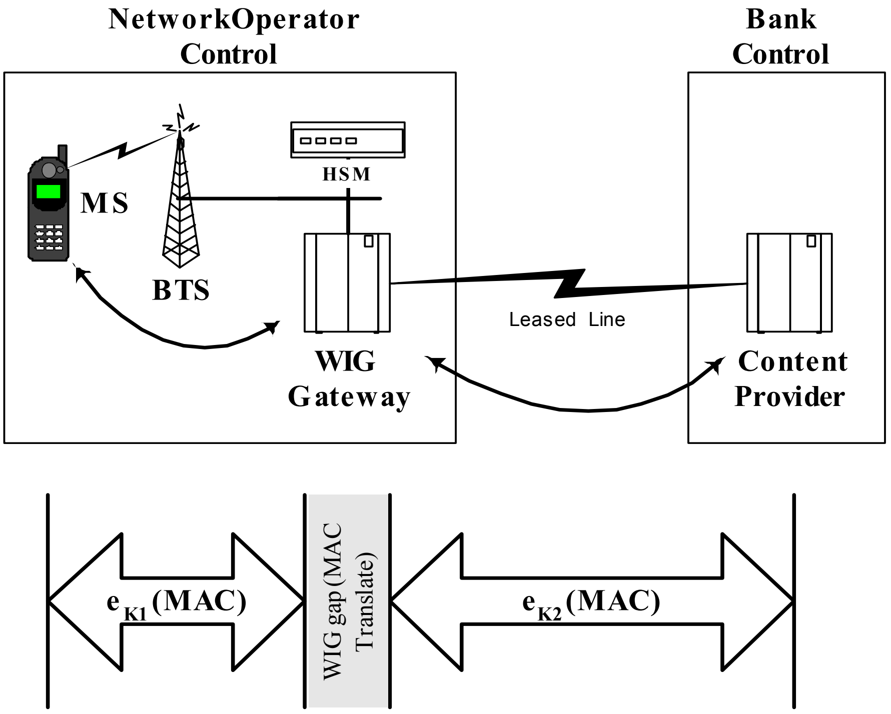


Figure 7.3: The WIG Gap

These hardware encryption devices are FIPS 140-1 level 3 certified, which by implications means they comply with a certain standard. This standard is accepted by all financial institutions in the world, including the four biggest Credit Card Organisations, i.e. VISA, MasterCard, American Express and Diners Club. It can therefore be assumed that, given the architecture of the proposed solution, the use of a HSM in the design greatly reduces, or even negates, the WIG Gap vulnerability as described above. The only problem the author has with this scenario is that the HSM is not under control of a trusted entity. It is one thing to trust that a Credit Card Organisation has implemented their HSMs correctly, and another matter entirely to trust a Network operator to implement the HSMs in the specified manner. The author suggests that this can be achieved by certifying the Network operator for m-Commerce applications. In doing so, the implementation of the WIG server and its corresponding HSM can be audited by independent individuals in order to insure that the devices are configured as specified.

Another option would be to move the WIG server into the trusted entity's area of influence and control. This would imply that as in the WAP Gap, the WIG server gets moved into the Network of the bank, as displayed in Figure 7.4.

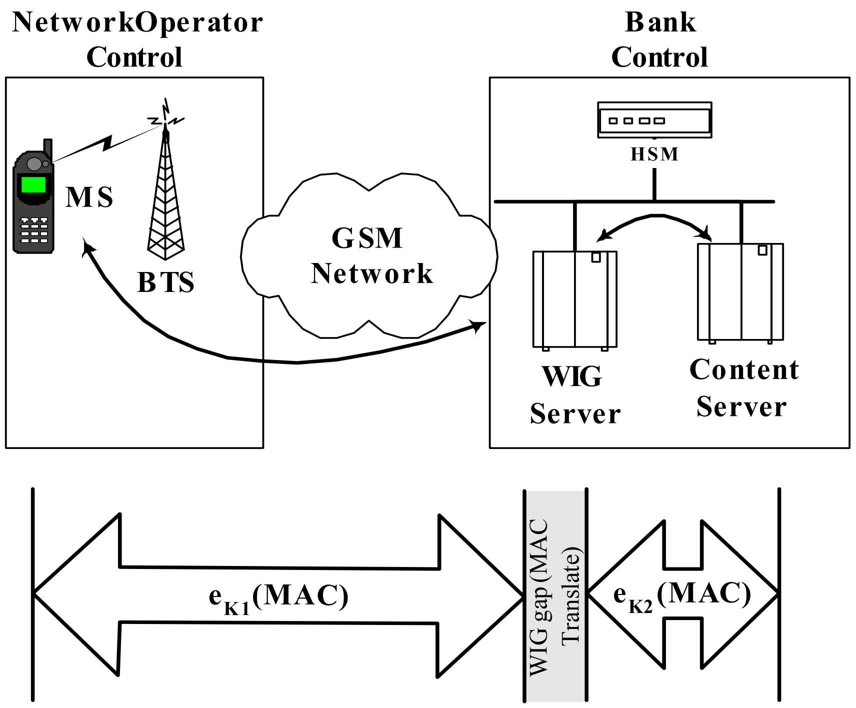


Figure 7.4: Closing the WIG Gap

This change in architecture effectively negates the WIG Gap vulnerability. During investigations of this option however, the author has found that this option is extremely expensive, due to the nature of the WIG server, and the volumes such a server should be able handle. Another factor that adds to the cost is that a bank wishing to deploy this solution would require at least one WIG server per Network operator. In South Africa this would imply that each bank that wants such a mobile payments solution would require three WIG servers in order to cater for the clients of the three Network operators resident in South Africa.

In the opinion of the author, the added security benefit derived from deploying the WIG server internal to the bank's network does not warrant such a huge expense.

CHAPTER 8: EXPERIMENTAL SETUP AND RESULTS

8.1 INTRODUCTION

In this chapter we will look at the design and architecture of an experimental simulator setup. The simulator was built in order for the author to verify that the solution he proposed in Chapter 5 can in actual fact deliver the required result. The Simulator was constructed from a commercially available phone simulator, and the applications to handle the transactions and encryption were programmed by the author and an assistant. Valuable information was derived from the simulator setup, and it conclusively proves that the proposed solution answers the need for message integrity in WIG m-Commerce transactions over GSM.

8.2 SIMULATOR ARCHITECTURE

The simulator consists of three functional entities as depicted in Figure 7.1:

1. The Phone Simulator application. This application is the SmartTrust WIG Application Creator 2.0. The phone simulator functions as the user interface to the m-Commerce application.
2. The HTTP logging application. This application captures and displays the HTTP traffic flowing between the phone simulator and the WIG simulator. This function is necessary in order to verify that the information sent and received does in actual fact represent the expected result.
3. The WIG Simulator application. The WIG simulator acts as the gateway between the phone simulator and the phantom content provider. It receives messages from the phone simulator, interprets and reformats them before forwarding the information to the phantom content provider for processing.

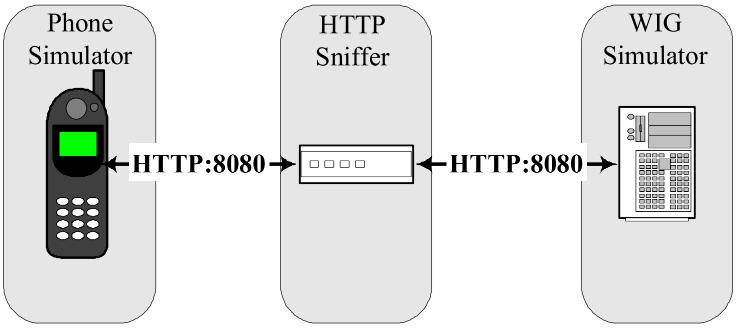


Figure 8.1: Simulator Architecture

8.3 SIMULATOR OPERATION

The Secure m-Commerce Simulator operates in the following way (Figure 7-2):

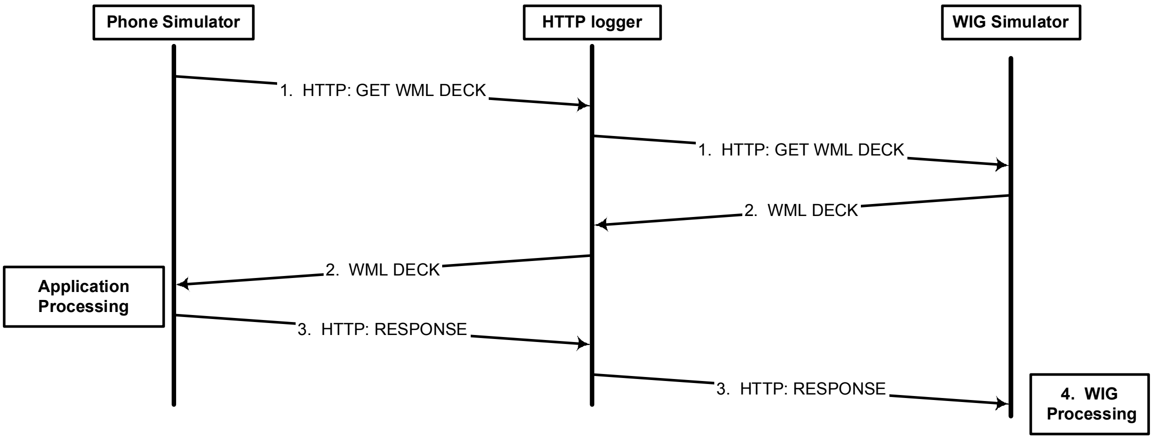


Figure 8.2: Simulator operation

1. The phone simulator requests the WML page from the WIG simulator via a HTTP GET request. In this request the phone simulator specifies the address where it believes the WML page to be as well as the originating MSISDN. The HTTP logger recorded the following expression as shown in Table 7.1. In all transactions the HTTP logger just displays the messages passing through it to the screen, and then forwards the message to the intended recipient.

```
>> == Sending request =========
14:06:46.093 GET /testserv/servlet1?MSISDN=0827849333 HTTP/1.1
14:06:46.093 Host: 127.0.0.1:8080
14:06:46.093 Accept: text/*; q=0.5, text/vnd.wap.wml
14:06:46.093 Accept-Charset: iso-8859-1, UTF-8
14:06:46.093 Accept-Encoding: identity
14:06:46.093 User-Agent: WIG Browser/1.2
14:06:46.093 Connection: close
14:06:46.453
```

Table 8-1: HTTP log of possible GET request

2. The WIG simulator receives the request, and replies with the WML page to the phone. The received page is a set of instructions written in WML that the phone interprets in order for the application to work. Please refer to Appendix A for the full WML source code.

```
<< == response ===============
14:06:46.453 HTTP/1.1 200 OK
14:06:46.453 ETag: W/"655-1061906352921"
14:06:46.453 Last-Modified: Tue, 26 Aug 2003 13:59:12 GMT
14:06:46.453 Content-Type: text/vnd.wap.wml
14:06:46.453 Content-Length: 655
14:06:46.453 Date: Thu, 04 Sep 2003 12:06:46 GMT
14:06:46.453 Server: Apache Coyote/1.0
14:06:46.453 Connection: close
14:06:46.453 <wml>
</wml>
```

Table 8-2: Possible HTTP response from WIG

3. Upon receipt of the WML page, the phone simulator displays the onscreen prompts for the user to enter the required information. Once the information is received, the phone simulator performs the following functions:

   a. Computes the encrypted PIN by making use of the triple DES key stored on the phone simulator and shared with the WIG simulator. The phone simulator makes use of the embedded WIB plug-in function named ENCR on the phone simulator. This is the same function that can be found on WIB compliant GSM phase 2+ phones.

b. Takes the rest of the provided information, i.e. the Amount, Credit Card number and banking PIN and performs a triple DES CBC MAC on the data using the triple DES key shared between the phone simulator and the WIG server.

c. Constructs the HTTP response that is to be sent to the WIG simulator. The response will then look as in Table 7-3.

```
http://127.0.0.1:8080/testserv/servlet1?pin=%00%E8%01z%9F%B1%1F%40%BF
      &Amount=112233&CreditCard=1234567890123456&Sig=%BA%08Qi
```

Table 8-3: Possible HTTP string from phone to WIG

Where:

Message extract	Description
127.0.0.1	The WIG Simulator IP address
8080	The TCP port to send the message on
/testserv/servlet1	The location of the WIG server application
pin=%00%E8%01z%9F%B1%1F%40%BF	The 3DES encrypted PIN
Amount=112233	The purchase amount
CreditCard=1234567890123456	The Credit Card Number
Sig=%BA%08Qi	The 3DES CBC MAC of the message, excluding the encrypted PIN.

Table 8-4: Possible Message from Phone to WIG

4. Upon receipt of the HTTP response message, the WIG Simulator performs the following functions:

   a. Verifies the MAC on the received message

   b. Translates the encrypted PIN to an ISO 9564-1 Format 0 PIN Block.

c. Triple DES encrypts the PIN Block with the triple DES key shared between the WIG and the content provider

d. Calculates the MAC on the message with all data excluding the encrypted PIN Block using the triple DES key shared between the WIG and the content provider.

e. Sends the information to the content provider.

Table 7-5 shows the information received by the WIG server. The MAC and encrypted PIN fields are displayed in HEX. Screen dumps of the received information can be seen in Appendix A.

Parameter	**Value received by WIG**
CreditCard #	1234567890123450
Amount	112233
MAC	BA085169
MSISDN	0827849333
Encrypted pin	E8017A9FB11F40BF
Clear Pin	1234

MAC Verification	
Mac Received	BA085169
MAC Calculated by WIG	BA085169

Message to Content Provider	
Clear Pin Block	0412719876FEDCBA
Encrypted Pin Block	FC8BCA50F59481AE
MAC to Content Provider	EBC8CE88

Table 8-5: Possible Parameter information from phone to WIG

Table 7-6 shows the Triple DES keys used between the different zones.

Zone	**Key**
Phone to WIG	3031323334353637 3839414243444546 3031323334353637
Wig to Content Provider	3736353433323130 4645444342413938 3736353433323130

Table 8-6: Cryptographic keys used in Simulator

8.4 CONCLUSION

As can be seen from the examples above, the simulator now provides message authentication capabilities by means of a 3DES MAC to the sensitive data. This will give the communicating parties the assurance that the transaction messages were not tampered with during transmission.

Another important conclusion that can be drawn from the simulator is that we now have true end-to-end confidentiality of the PIN transmitted in the message. This is due to the fact that the WIG server only translated the PIN from the encryption used between the phone and the WIG into a format that can be interpreted by the financial institution. The financial institution is therefore the only entity that verifies the PIN entered by the client on the mobile station. The PIN is secured from the phone up until the HSM module of the financial institution.

We therefore have conclusively proved that by adding a MAC to the transmitted SMS financial transaction we have significantly improved the security of the WIG financial solution by providing end-to-end message integrity of the SMS.

CHAPTER 9: CONCLUSION

9.1 DISCUSSION

Just as the Internet has changed the face of commerce forever, so GSM has changed the face of communication and information delivery forever. Although originally only intended for voice traffic, GSM networks have continued to develop, as have the GSM-associated services, or Value Added Services.

Today, a myriad of value added services are associated with GSM mostly because of the adoption of international standards, which guarantees interoperability. Some of these services include the Short Message Service (SMS), Wireless Internet Gateway (WIG) and the SIM Application Toolkit (SAT). Each of these services brings with them a host of possible applications and opportunities. One such opportunity is m-Commerce. By using mobile phones over a GSM network, users of these applications can effect a financial transaction from anywhere in the world where there is GSM network coverage.

However, with these location independent opportunities comes risk. In order to ensure the integrity of payment instructions, these m-Commerce messages must be secured, by some means, while traversing the global networks. In essence, the security required for these transactions are very similar to the security of more traditional Internet based e-Commerce transactions.

Although GSM provides ample security on the air interface, the GSM backbone is an entirely different matter. Data is unencrypted and unprotected on the backbone network of the service provider. This fact is further explored in Chapter 3. In the case of normal GSM voice and message applications, transmission across these networks does not pose a huge risk. Adding financial value to the equation changes the picture considerably. In this scenario, an attacker could potentially reap huge financial benefit from altering, or even fabricating messages. The possible attacks on the GSM network are explored in Chapter 4.

With this in mind, Chapters 6 and 7 drills down into the WIG application order to see what security is offered by the application, and what is not. Several areas of vulnerability are highlighted and suggestions of solutions for these areas are proposed.

9.2 RESULTS

Throughout this dissertation the author took the viewpoint of no trust. This resulted in some interesting observations. Solutions that appear concrete at one point can be dashed suddenly when other factors are uncovered.

The cryptographic capabilities of the different handsets are the first limiting factor, as this directly limits available functionality on the handset. Owing to the fact that the South African cellular market is predominantly pre-paid (pay-as-you-go), users must purchase their own handsets upfront. This initial expenditure for a GSM handset usually limits the accessibility of top-end cellular handset technology due to the high cost of such handsets. For this reason the proposed solution made use of the WIG m-Commerce model and not one of the more advances GSM technologies

The next factor is in the way the network operators actually implemented their networks. Some operators still do not conform to the specifications mandated by GSM. This implied that the solution had to be network independent. The proposed solution therefore made use of standard SMS functionality with the assistance of the SIM Application Toolkit.

The issue of end-to-end message integrity still had to be addressed. The banking PIN was encrypted by making use of the SIM card resident on the mobile station, which ensure confidentiality of the PIN. All other data, except the PIN was then fed through the 3DES algorithm and a MAC was derived from it. This MAC accompanied the SMS messages from the mobile station to the financial institution, providing end-to-end message integrity.

As a result of this, financial transactions may be conducted over the Wireless Internet Gateway with confidence.

9.3 FUTURE WORK

The words of a well-known Carpenters' song, "*We've only just begun…*" best describe the position of m-Commerce at this point in time. The technologies providing these services have not yet matured and already new technologies are introduced. GPRS, for instance, was only launched in South Africa in mid 2002. With each new technology comes a new security challenge. This might imply that WIG will use GPRS networks in future, and therefore we need to evaluate the security provided to us by GPRS and then assess what is lacking, and what can be done to improve the security for financial transactions.

The ever increasing enhancements in Smart Card technology means that we will shortly see a 64k SIM card available which can handle a new type of m-Commerce technology. Again this new system needs to be evaluated for its security capabilities.

With the coming of 3rd Generation networks in the next couple of years the game is going to change a bit. With 3G networks the best, and the worst, of the Internet and mobile networks are thrown together into one lot. This is a challenge I am personally looking forward to.

BIBLIOGRAPHY

[1] The ARC Group. *Mobile Financial Services: From Concept to revenues*. September 2000.

[2] Senn, J. *The Emergence of m-Commerce,* @ IEEE Computer Magazine. December 2001. Pages 148-150.

[3] Kekki, B. *m-Commerce Security*. Helsinki School of Economics. 2001. www.mli.hkkk.fi/users/mba4/benedek/Navigation/Education/Coursework/Mobile_Security_Paper.htm

[4] MÜller-Veerse, F. *Mobile Commerce Report*. 2000. Durlacher Research Ltd. London. United Kingdom.

[5] Varshney, U., Vetter, R., Kalakota, R. *Mobile Commerce: A New Frontier,* @ IEEE Computer Magazine. October 2001.

[6] Tsalgatidou, A., Veijalainen, J., Pitoura, E. *Challenges in Mobile Electronic Commerce*. Proceedings of IeC 2000. 3rd International Conference on Innovation through e-Commerce. Manchester, U.K. November 2000.

[7] Mynttinen, J. E*nd-to-end security of mobile data in GSM*. Helsinki University of Technology. Helsinki. Finland. November 2001.

[8] Hage, D. *Secure E-Commerce – Assignment 2. WAP Security in M-Commerce.* University of South Australia. Division of Information Technology, Engineering and the Environment. School of Computer and Information Science. Australia. 2001.

[9] Vyas, A., O'Grady, P. *A Review of Mobile Commerce Technologies. Internet Lab Technical Report TR 2001-06*. Department of Industrial Engineering. University of Iowa. Iowa. May 2001.

[10] Duraiappan, C., Zheng, Y. *Enhancing Security in GSM.* University of Wollongong. Melbourne. Australia. 1999.

[11] Scmidt, M. *Consistent M-Commerce Security on Top of GSM-based Data Protocols – A Security Analysis*. University of Siegen, Institute for Data Communication Systems. Siegen. Germany. 2001.

[12] European Telecommunications Standards Institute. *Digital cellular telecommunications system (Phase 2+); Security mechanisms for the SIM Application Toolkit; Stage1*. GSM 02.48 version 6.0.0 Release 97. ETSI. April 1998.

[13] European Telecommunications Standards Institute. *Digital cellular telecommunications system (Phase 2+); Security mechanisms for the SIM Application Toolkit; Stage2*. GSM 03.48 version 6.0.0 Release 97. ETSI. April 1998.

[14] Stallings, W. *Cryptography and Network Security: Principles and Practice*. Second Edition. Prentice Hall. Upper Saddle River, New Jersey. 1999

[15] Pfleeger, C. *Security in Computing*. Second Edition. Prentice Hall Inc. Upper Saddle River, New Jersey. 1997

[16] Menezes, A. van Oorschot, P. Vanstone, S. *Handbook of Applied Cryptography*. CRC Press. 1996.

[17] Du Toit, J., van der Merwe, P. *NISA. The Nedcor Information Security Architecture*. Nedcor Bank Limited. October 2001.

[18] Margrave, D. *GSM Security and Encryption*. George Mason University. http://spyhard.narod.ru/phreak/gsm-secur.html

[19] Scourias, J. *A Brief overview of GSM*. University of Waterloo. 1994.

[20] van der Merwe, P. *Advanced Communication and Network Technology Assignment 1*. University of Pretoria. 2001.

[21] Bettstetter, C., Vogel, H., Eberspacher, J. *General Packet Radio Service (GPRS): Architecture, Protocols and Air Interface*. www.zzz.com.ru/art61.html.

[22] van der Merwe, P. *Advanced Computer Networks Exam Assignment: A Study of Smart Cards*. University of Pretoria. 2001.

[23] Gadaix, E. *GSM and 3G Security*. Presentation presented at Black Hats conference in Hong Kong and Singapore. April 2001.

[24] European Telecommunications Standards Institute. *Digital cellular telecommunications system (Phase 2+); Security aspects*. GSM 02.09 version 7.0.1. ETSI. January 2000.

[25] European Telecommunications Standards Institute. *Digital cellular telecommunications system (Phase 2+); Security related network functions*. GSM 03.20 version 8.0.0. ETSI. October 2000.

[26] RACAL Research LTD. *Technical Information: GSM System Security Study*. Reading. England. June 1988.

[27] Pesonen, L. *GSM Interception*. Department of Computer Science and Engineering. Helsinki University of Technology. November 1999.

[28] Scourias, J. *Overview of GSM: The Global System for Mobile Communications*. University of Waterloo. March 1996.

[29] Popov, M. *Security Aspects of the Cellular Communications*. Information and Security. Volume 4. 2000

[30] Rao, J. Rothatgi, P. Scherzer, H. et al. *Partitioning Attacks: Or How to Rapidly clone Some GSM Cards*. 2002 IEEE Symposium on Security and Privacy. Oakland. May 2002.

[31] Biryukov, A., Shamir, A., Wagner, D. *Real Time Cryptanalysis of A5/1 on a PC*. The Weizmann Institute, Rehovot, Israel. 2000.

[32] Business Wire. *GSM Alliance Clarifies False & Misleading Reports of Digital Phone Cloning*. Business Wire Magazine, 20 April 1998. http://jya.com/gsm042098.txt

[33] Lorenz, G., Moore, T., Manes, G., Hale, J., Shenoi, S. *Securing SS7 Telecommunications Networks*. Proceedings of the IEEE Workshop on Information Assurance and Security. United States Military Academy. West Point. New York State. June 2001.

[34] Schmidt, M. *Consistent m-Commerce Security on Top of GSM-based Data Protocols – A Security Analysis*. University of Siegen. Institute for Data Communication Systems. Germany.

[35] Sema. *Unstructured Supplementary Services Data (USSD) - A Detailed Overview*. www.sema.com

[36] European Telecommunications Standards Institute. *Digital cellular telecommunications system (Phase 2); Unstructured Supplementary Service Data (USSD); Stage 1. GSM 02.90 version 4.1.1.* ETSI. September 1997.

[37] Buckingham, S. *Success 4 SMS Whitepaper*. Mobile Streams. February 2001. www.mobilestreams.com

[38] The International Engineering Consortium. *Wireless Short Message Service (SMS).* Web ProForm Tutorials. www.iec.org

[39] European Telecommunications Standards Institute. *Point-to-Point Short Message Service Support on Mobile Radio Interface*. GSM 04.11. ETSI. January 1993.

[40] European Telecommunications Standards Institute. *Digital cellular telecommunications system (Phase 2); Technical realization of the Short Message Service (SMS) Point-to-Point (PP)*. GSM 03.40. ETSI. October 1996.

[41] Howell, R. *WAP Security*. Concise Group Ltd.

APPENDIX A: SECURE M-COMMERCE WIG SIMULATOR

A.1 INTRODUCTION

This Appendix describes the operation of the Secure M-Commerce WIG Simulator in more detail. Specific attention is paid to the WIG server, as the phone simulator and the HTTP sniffer is commercially available products. The WIG server simulation is however a custom built application written in Java.

A.2 SIMULATOR ARCHITECTURE

The WIG Simulator consists of three components. The Phone Simulator, an HTTP sniffer, and the WIG simulator, as depicted in Figure A-1:

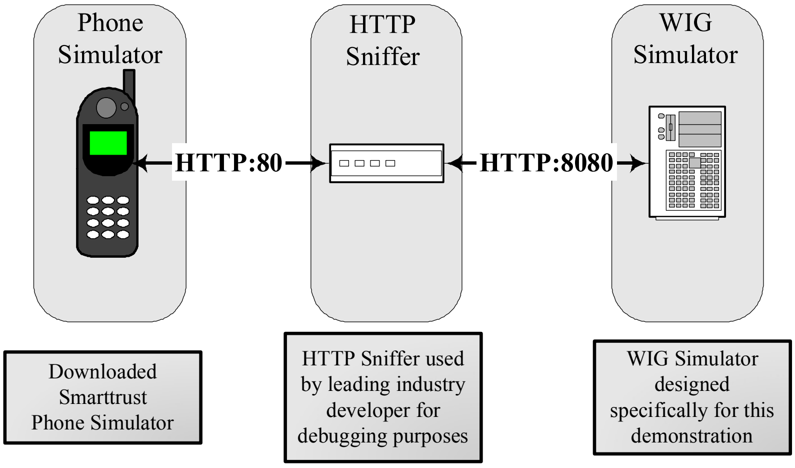

Phone Simulator
HTTP Sniffer
WIG Simulator
HTTP:80
HTTP:8080
Downloaded Smarttrust Phone Simulator
HTTP Sniffer used by leading industry developer for debugging purposes
WIG Simulator designed specifically for this demonstration

Figure A-0.1: Simulator Architecture

The individual components of the architecture were sources from different places:

- The Phone Simulator was downloaded from the SmartTrust website on the Internet.
- An industry-leading expert in the field of HTTP and WML applications wrote the HTTP sniffer, and he graciously provided permission to the author to make use of his application.
- The WIG simulator was specifically designed by the Author and a Java Application Developer to perform the required functions in order to provide proof of the proposed secure m-Commerce interface using WIG.

A.3 SIMULATOR OPERATION

The WIG simulator operates in the following way:

1. The phone simulator sends a WML POST request to the HTTP sniffer on HTTP port 80.
2. The sniffer will log the request to screen, and forward the request to the WIG simulator on HTTP port 8080.
3. Once the WIG simulator receives the request it passes the WML bytecode back to the phone.
4. The phone will display the menu structure and user prompts as described in the WML bytecode.
5. The user goes through the menu prompts and enters the required information in each section.
6. Once all the required fields have been harvested, the Phone simulator constructs the return message according the instructions received from the WML bytecode.
7. The return message is then sent to the HTTP sniffer and from there forwarded to the WIG gateway.

8. Once the wig gateway receives the message, it does the following:

   a. Verifies the MAC on the received message,

   b. Translates the received encrypted PIN into an ISO 9664-1 Format 0 PIN Block.

   c. Calculates a new MAC on the message

   d. Forwards the newly constructed message to the Content Provider.

A.4 MESSAGE FORMAT

The Phone Simulator formats the message from user input according to the instructions it received in the WML bytecode from the WIG. The WML bytecode gives the following instruction to the phone: (The complete WML bytecode is listed in Appendix E)

```
<go href="http://127.0.0.1:8080/testserv/servlet1?pin=$(CipherText)&amp;A
mount=$(amt)&amp;CreditCard=$(ccard)&amp;Sig=$(MAC)"/>
```

Once the user input is captured, the formatted message looks something like this:

http://127.0.0.1:8080/testserv/servlet1?pin=%00%E8%01z%9F%B1%1F%40%BF&Amount=1234&CreditCard=1234567890123456&Sig=%B3%7FK%03

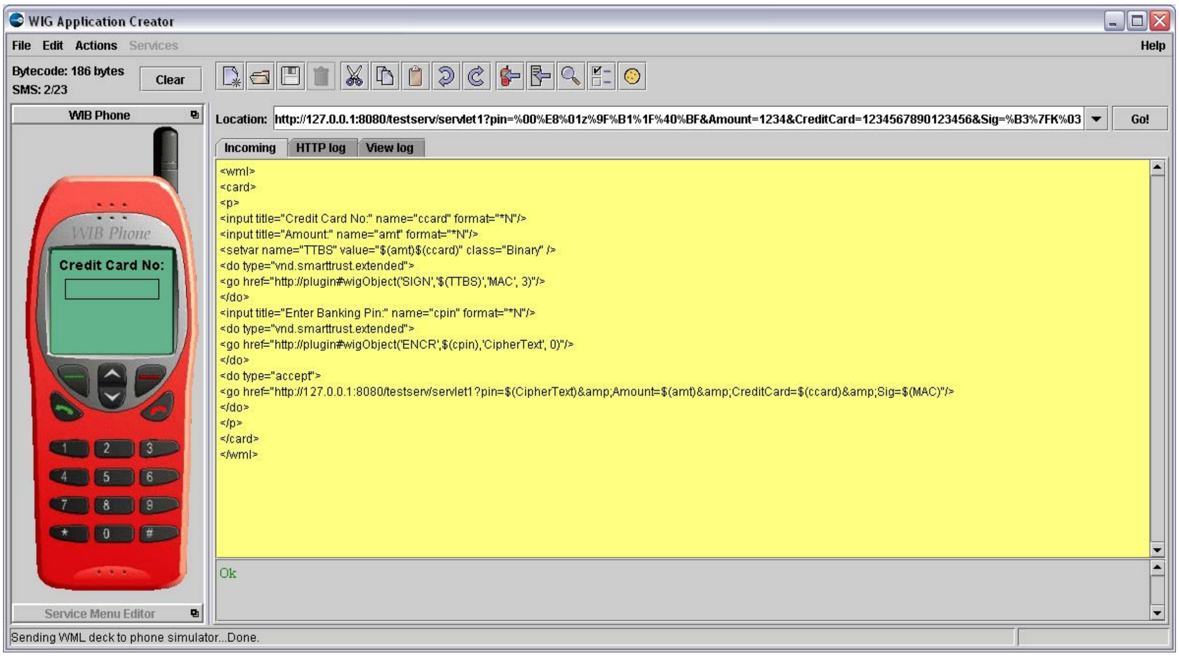
The elements in the message constructed by the phone in explained in Table A-1:

Message extract	Description
127.0.0.1	The HTTP sniffer IP address
8080	The TCP port to send the message on
/testserv/servlet1	The location of the WIG server application
pin=%00%E8%01z%9F%B1%1F%40%BF	The 3DES encrypted PIN
Amount=1234	The purchase amount
CreditCard=1234567890123456	The Credit Card Number
Sig=%B3%7FK%03	The 3DES CBC MAC of the message

Table A-1: Simulator Message from Phone to WIG

As can clearly be seen, only the Banking PIN is encrypted in this message. The MAC is calculated on all other data, except the encrypted PIN. This is due to the fact that the encrypted PIN gets translated to an ISO PIN Block at the WIG server for forwarding to the Content Provider. This ensures end-to-end message integrity with specific reference to the amount and the account number fields. These fields cannot be altered without the knowledge of the Content Provider.

Figure A-2 shows how this message looks on the WIG Simulator. The message string can clearly be seen on the address bar named Location in the figure. The whole DEC received from the WIG server can be seen on the Incoming tab on the WIB Browser page.



Location: http://127.0.0.1:8080/testserv/servlet1?pin=%00%E8%01z%9F%B1%1F%40%BF&Amount=1234&CreditCard=1234567890123456&Sig=%B3%7FK%03

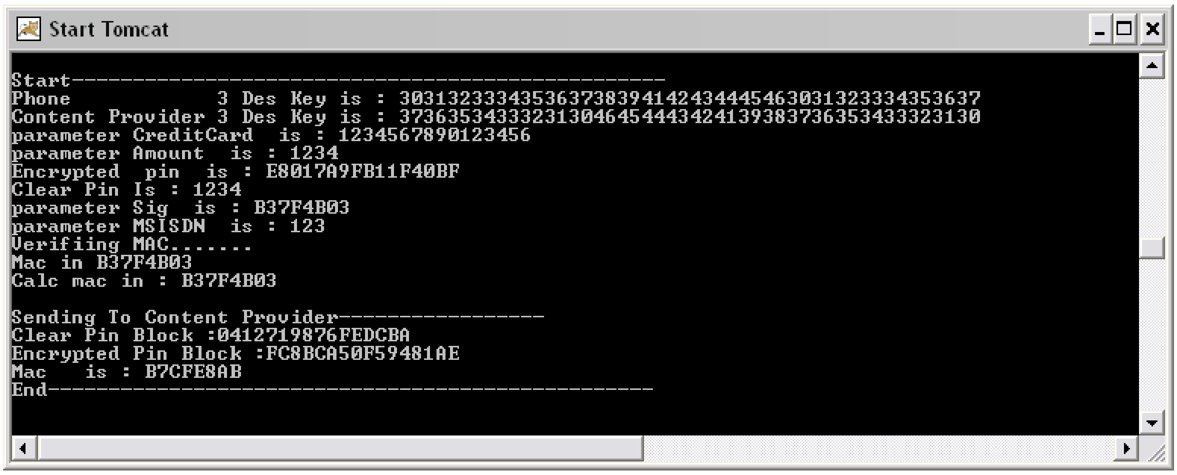
```
<wml>
<card>
<p>
<input title="Credit Card No:" name="ccard" format="*N"/>
<input title="Amount:" name="amt" format="*N"/>
<setvar name="TTBS" value="$(amt)$(ccard)" class="Binary" />
<do type="vnd.smarttrust.extended">
<go href="http://plugin#wigObject('SIGN','$(TTBS)','MAC', 3)"/>
</do>
<input title="Enter Banking Pin:" name="cpin" format="*N"/>
<do type="vnd.smarttrust.extended">
<go href="http://plugin#wigObject('ENCR',$(cpin),'CipherText', 0)"/>
</do>
<do type="accept">
<go href="http://127.0.0.1:8080/testserv/servlet1?pin=$(CipherText)&amp;Amount=$(amt)&amp;CreditCard=$(ccard)&amp;Sig=$(MAC)"/>
</do>
</p>
</card>
</wml>
```

```
Ok
```

```
Sending WML deck to phone simulator...Done.
```

Figure A-0.2: WIB Browser Simulator Page

Figure A-3 shows an actual screen dump of the information received by the WIG Simulator and sent to the Content Provider in the process of a transaction: (This information is displayed on the screen of the computer acting as the WIG server)



```
Start-------------------------------------------------
Phone              3 Des Key is : 303132333435363738394142434445463031323334353637
Content Provider 3 Des Key is : 373635343332313046454443424139383736353433323130
parameter CreditCard  is : 1234567890123456
parameter Amount  is : 1234
Encrypted  pin  is : E8017A9FB11F40BF
Clear Pin Is : 1234
parameter Sig  is : B37F4B03
parameter MSISDN  is : 123
Verifiing MAC.......
Mac in B37F4B03
Calc mac in : B37F4B03

Sending To Content Provider-----------------
Clear Pin Block :0412719876FEDCBA
Encrypted Pin Block :FC8BCA50F59481AE
Mac   is : B7CFE8AB
End--------------------------------------------------
```

Figure A-0.3: Screen dump of WIG Simulator messages

The elements of the information received by the WIG Simulator and sent to the Content Provider in the process of a transaction are described in Table A-2:

Field values	Description
Phone 3DES Key is : 303132333435363738394142434445463031323334353637	The 3DES key stored in HEX on the phone simulator
Content Provider 3DES Key is : 373635343332313046454443424139383736353433323130	The 3DES key shared between the WIG and the Content Provider. This key usually resides on the HSM connected to the WIG server.
Parameter CreditCard is : 1234567890123456	The Credit Card value received in the message
Parameter Amount is : 1234	The amount value received in the message
Encrypted pin is : E8017A9FB11F40BF	The PIN 3DES encrypted with the Phone's 3DES key
Clear Pin Is : 1234	The Clear text value of the PIN. (Only displayed for debugging purposes)
Parameter Sig is : B37F4B03	The value of the Sig parameter. This is the MAC of the message displayed in HEX
Parameter MSISDN is : 123	The value of the MSISDN Parameter, i.e. the phone number
Mac in B37F4B03	This is the received MAC
Calc mac in : B37F4B03	This is the MAC the WIG server calculated to verify the MAC received
Clear Pin Block :0412719876FEDCBA	This is the unencrypted PIN block assembled by the WIG server
Encrypted Pin Block :FC8BCA50F59481AE	This is the PIN block after encryption by the WIG server
Mac is : B7CFE8AB	This is the MAC on the message traveling to the Content Provider. The MAC was calculated using the shared key between the WIG and the Content Provider.

Table A-2: Information received and processed by the WIG

APPENDIX B: SOURCE CODE: WIG SIMULATOR

This appendix contains an implementation of the Secure WIG Server. The implementation is written in Java.

```
// This program was written to demonstrate the secure WIG server model
// as proposed by Pieter van der Merwe.
//
// Date: 11 August 2003
// Author: P.B. van der Merwe and D. Goosen
// Fileneme:  Servlet1.java


package testserv;

import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
import java.util.*;

public class Servlet1 extends HttpServlet {
  /**Initialize global variables*/
  public void init() throws ServletException { }
  /**Process the HTTP Get request*/
  public void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
    doAll(request,response);
  }
  public void doPost(HttpServletRequest request, HttpServletResponse
response)   throws ServletException, IOException {
    doAll(request,response);
  }

  public void doAll(HttpServletRequest req, HttpServletResponse resp)
      throws ServletException, IOException {
      // print out the rest of the parameters
    Enumeration e = req.getParameterNames();
    String amt = "";
    String cc = "";
    clearPinGlobal = null;

    if (req.getQueryString().indexOf("pin") >= 0 ) {

      if (TDesKey==null) {
        TDesKey =
HexConverter.byteArrayToHexString(TDesKeyAscii.getBytes());
        TDesKeyOut =
HexConverter.byteArrayToHexString(TDesKeyAsciiOut.getBytes());
      }

      System.out.println();
      System.out.println();
      System.out.println("Start--------------------------------------");
      System.out.println("Phone            3 Des Key is : " + TDesKey);
      System.out.println("Content Provider 3 Des Key is : " +
TDesKeyOut);
      while (e.hasMoreElements()) {
        String ss= (String)e.nextElement();
        try {
          if (!ss.equalsIgnoreCase("pin")) {
             if (!ss.equalsIgnoreCase("Sig")) {
               System.out.println("parameter "+ss+"  is : " +
req.getParameter(ss));
             }
             else {
                String sig =
HexConverter.byteArrayToHexString(req.getParameter(ss).getBytes("ISO-
8859-1"));
                System.out.println("parameter "+ss+"  is : " + sig);
             }
          }
          else {
            // pin decrypt
            String encHexPin =
HexConverter.byteArrayToHexString(req.getParameter(ss).getBytes("ISO-
8859-1"));
            encHexPin = encHexPin.substring(2,encHexPin.length());
            System.out.println("Encrypted  "+ss+"  is : " + encHexPin );
            doCrypto(HexConverter.hexStringToByteArray(encHexPin));
          }
        }catch(Exception ee) {
          throw new IOException(ee.getMessage());
        }
      }
      // do inmac verification
      try {
        System.out.println("Verifiing MAC.......");

        byte[] macData = ( req.getParameter("Amount") +
req.getParameter("CreditCard")).getBytes();
        int cntmac = ( Math.round (macData.length / 8) + 1) * 8;
        byte[] binMac = new byte[cntmac];
        System.arraycopy(macData,0,binMac,0,macData.length);

        String inMac =
HexConverter.byteArrayToHexString(req.getParameter("Sig").getBytes("ISO-
8859-1"));
        doMac(binMac,inMac);
        System.out.println();
        System.out.println("Sending To Content Provider-----------------
");
        doCryptoPinBlock(clearPinGlobal,req.getParameter("CreditCard"));
        doMacOut(macData);
      }catch(Exception ee) {
        throw new IOException(ee.getMessage());
      }
      System.out.println("End-----------------------------------------");
    }
    // get the next deck

this.getServletContext().getRequestDispatcher("/deck.wml").forward(req,re
sp);
  }

  private void doCrypto(byte[] pin) throws Exception{
    Crypto cipher = new Crypto();
    cipher.setDefaultCryptoParams();
    byte[] clearPin = cipher.setDataDecrypt(pin, TDesKey  );
    clearPinGlobal = new String(clearPin);
    clearPinGlobal =
clearPinGlobal.substring(0,clearPinGlobal.indexOf(0));
    System.out.println("Clear Pin Is : "+ clearPinGlobal);
  }

  private void doMac(byte[] clearMacData,String inMac) throws Exception{
    Crypto cipher = new Crypto();
    cipher.setCryptoParams("DESede","CBC","SHA1","NoPadding","SunJCE",new
byte[8]);
    byte[] macData = cipher.setDataEncrypt(clearMacData,TDesKey);
    System.out.println("Mac in "+ inMac);
    String calcMac = HexConverter.byteArrayToHexString(macData);
    System.out.println("Calc mac in : "+
calcMac.substring(calcMac.length()-16,calcMac.length()).substring(0,8));
  }

  private void doMacOut(byte[] clearMacData) throws Exception{
    int cntmac = ( Math.round (clearMacData.length / 8) + 1) * 8;
    byte [] padding = "FFFFFFFF".getBytes();
    byte [] binMac = new byte[cntmac];
    System.arraycopy(padding,0,binMac,binMac.length-8,8);
    System.arraycopy(clearMacData,0,binMac,0,clearMacData.length);

    Crypto cipher = new Crypto();
    cipher.setCryptoParams("DESede","CBC","SHA1","NoPadding","SunJCE",new
byte[8]);
    byte[] macData = cipher.setDataEncrypt(binMac,TDesKeyOut);
    String calcMac = HexConverter.byteArrayToHexString(macData);
    //System.out.println("Mac Data :" +calcMac);
    System.out.println("Mac   is : "+calcMac.substring(calcMac.length()-
16,calcMac.length()).substring(0,8));
  }


  private void doCryptoPinBlock(String clearPin,String acc) throws
Exception {
    acc = "0000" + acc.substring(acc.length() - 13,acc.length()-1);
    String pinBlockZero = "0" + String.valueOf(clearPin.length()) +
clearPin + "FFFFFFFFFFFFFFFF";
    pinBlockZero = pinBlockZero.substring(0,16);

    java.math.BigInteger bi1 = new java.math.BigInteger(acc,16);
    java.math.BigInteger bi2 = new java.math.BigInteger(pinBlockZero,16);
    pinBlockZero = "0"+bi1.xor(bi2).toString(16).toUpperCase();
    System.out.println("Clear Pin Block :" + pinBlockZero);

    Crypto cipher = new Crypto();
    cipher.setDefaultCryptoParams();

    byte[] encPinBLock =
cipher.setDataEncrypt(HexConverter.hexStringToByteArray(pinBlockZero),TDe
sKeyOut);
    System.out.println("Encrypted Pin Block :"
+HexConverter.byteArrayToHexString(encPinBLock));
  }


  public void destroy() { }

  private  String clearPinGlobal     = null;
  private  String TDesKey            = null;
  private static String TDesKeyAscii = "0123456789ABCDEF01234567";

  private  String TDesKeyOut            = null;
  private static String TDesKeyAsciiOut = "76543210FEDCBA9876543210";

}
```

APPENDIX C: SOURCE CODE: JAVA CRYPTOGRAPHY

This appendix contains an implementation of the Java Cryptography used in the Secure WIG application. Standard 3DES in CBC-mode was used to both encrypt, and derive the MAC for the messages. The implementation is written in Java.

```
// This program was written to develop the cryptographic tools to be used
//  in the Secure WIG implementation as proposed by Pieter van der Merwe.
//
// Date: 11 August 2003
// Author: P.B. van der Merwe and D. Goosen
// Fileneme:  Crypto.java

package testserv;

import javax.crypto.*;
import javax.crypto.spec.*;
import java.security.*;
import java.io.*;

public class Crypto {
       public void Crypto() {
         cipher = null;
       }

      public void setCryptoParams(String algo,String mode,String
hash,String padding,String provider,byte[] iv) throws Exception {
            cipher = null;
            this.algo = algo;
            this.provider = provider;
            this.mode = mode;
            this.padding = padding;
            this.iv = null;
            this.hash = hash;
            setIV(iv);
      }

      public void setDefaultCryptoParams() throws Exception {

      setCryptoParams("DESede","ECB","SHA1","NoPadding","SunJCE",null);
                  }
```

```
   public byte[] setDataDecrypt(byte[] encData,String key) throws
Exception{
      setKey(key);
      initDeCrypt();
      return process(encData);
   }

   public byte[] setDataEncrypt(byte[] clrData,String key) throws
Exception{
      setKey(key);
      initEnCrypt();
      return process(clrData);
   }

    public byte[] getDigest(String data)throws Exception {
       MessageDigest md = MessageDigest.getInstance(hash);
       return md.digest(data.getBytes());
    }


   // private ---------------------------------------------------

   private void setIV(byte[] iv)  throws Exception {
      if (iv != null) {
         byte[] encParam = new byte[iv.length + 2];
// 2 for ASN blowfish ??
         encParam[0] = ASN1_TAG;
         String hexVal = Integer.toHexString(iv.length);
         if ((hexVal.length() & 1) == 1)  hexVal = '0' + hexVal;
         encParam[1] =
HexConverter.hexStringToByteArray(hexVal)[0];
         System.arraycopy(iv,0,encParam,2,iv.length);
         this.iv = encParam;
      }
   }

   private void genKey(String clearKey) throws Exception {
      byte[] digest = getDigest(clearKey);
      byte[] keyData = null;
      if ("DESede".equalsIgnoreCase(algo)) {
         keyData = new byte[24];
```

```
                  System.arraycopy(digest,0,keyData,0,16);
                  System.arraycopy(digest,0,keyData,16,8);
                  key = new SecretKeySpec(keyData, algo);
            }
      }

        private void setKey(String clearKey) throws Exception {
                byte[]                      keyData                      =
HexConverter.hexStringToByteArray(clearKey);
                 key = new SecretKeySpec(keyData, algo);
        }

      private void initDeCrypt()  throws Exception {
            checkCipher();
            if (iv == null)
                  cipher.init(Cipher.DECRYPT_MODE,key);
            else {
                   AlgorithmParameters parm = null;
                   parm = AlgorithmParameters.getInstance(algo,provider);
                   parm.init(iv);
                   cipher.init(Cipher.DECRYPT_MODE,key,parm);
             }
       }

      private void initEnCrypt()  throws Exception {
            checkCipher();
            if (iv == null)
                  cipher.init(Cipher.ENCRYPT_MODE,key);
            else {
                   AlgorithmParameters parm = null;
                   parm = AlgorithmParameters.getInstance(algo,provider);
                   parm.init(iv);
                   cipher.init(Cipher.ENCRYPT_MODE,key,parm);
             }
       }

       private byte[] process(byte[] inData) throws Exception  {
             return  cipher.doFinal(inData);
       }
       private void checkCipher() throws Exception {
            if (cipher == null)
                  cipher                                               =
Cipher.getInstance(algo+"/"+mode+"/"+padding,provider);
       }

       private static void loadDefaultProvider() throws Exception{

java.security.Security.addProvider((java.security.Provider)Class.forName(
defaultProvider).newInstance());
       }

       private static byte ASN1_TAG             = 4;
       private    static    String    defaultProvider                 =
"com.sun.crypto.provider.SunJCE";

       static {
            try {
                    loadDefaultProvider();
             } catch (Exception e) {
                    System.err.println(e.getMessage());
             }
       }


       private byte[]  iv               = null;
       private Cipher  cipher           = null;
       private Key     key              = null;
       private String  algo             = null;
       private String  provider         = null;
       private String  mode             = null;
       private String  padding          = null;
       private String  hash             = null;


         public static void main(String args[]) throws Exception   {
           Crypto c = new Crypto();
           c.setDefaultCryptoParams();
          //
c.setCryptoParams("DESede","ECB","SHA1","NoPadding","SunJCE",null);
```

```
           byte[]                          ct                            =
c.setDataEncrypt("12345678".getBytes(),"123456789012345612345678901234561
234567890123456");

           System.out.println(HexConverter.byteArrayToHexString(ct));

           System.out.println(new
String(c.setDataDecrypt(ct,"12345678901234561234567890123456123456789O123
456")));

         }

}
```

APPENDIX D: SOURCE CODE: HEX CONVERTER

This appendix contains an implementation of a Hexadecimal to ASCII converter. The implementation is written in Java.

```
// This program was written as a tool for the conversion of ASCII values
// entered into Hexadecimal values for use by the cryptographic function
// in the Secure WIG server model as proposed by Pieter van der Merwe.
//
// Date: 11 August 2003
// Author: P.B. van der Merwe and D. Goosen
// Fileneme:  Hexconverter.java


package testserv;

import java.io.ByteArrayOutputStream;

public class HexConverter {

  public static byte[] hexStringToByteArray(String hexString)
  {
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    char c1;
    char c2;
    byte b = 0;
    if((hexString.length() & 1) == 1)
      throw new IllegalArgumentException("Invalid Hex String ODD Number
of Arguments");
    for(int i = 0;i < hexString.length();i += 2)
    {
      c1 = hexString.charAt(i);
      c2 = hexString.charAt(i + 1);
      b = 0;
      if((c1 >= '0') && (c1 <= '9'))
        b += ((c1 - '0') * 16);
      else
        if((c1 >= 'a') && (c1 <= 'f'))
          b += ((c1 - 'a' + 10) * 16);
      else
        if((c1 >= 'A') && (c1 <= 'F'))
          b += ((c1 - 'A' + 10) * 16);
      else
        throw new IllegalArgumentException("Invalid Hex String");
      if((c2 >= '0') && (c2 <= '9'))
        b += (c2 - '0');
      else
        if((c2 >= 'a') && (c2 <= 'f'))
          b += (c2 - 'a' + 10);
      else
        if((c2 >= 'A') && (c2 <= 'F'))
          b += (c2 - 'A' + 10);
      else
        throw new IllegalArgumentException("Invalid Hex String");
      baos.write(b);
    }
    return (baos.toByteArray());
  }
  public static String byteArrayToHexString(byte bytes[])
  {
    StringBuffer sb = new StringBuffer(bytes.length * 2);
    for(int i = 0;i < bytes.length;i++)
    {
      sb.append(convertDigit((int)(bytes[i] >> 4)));
      sb.append(convertDigit((int)(bytes[i] & 0x0f)));
    }
    return (sb.toString().toUpperCase());
  }
  private static char convertDigit(int value)
  {
    value &= 0x0f;
    if(value >= 10)
      return ((char)(value - 10 + 'a'));
    else
      return ((char)(value + '0'));
  }
}
```

APPENDIX E: SOURCE CODE: WML BYTECODE

This appendix contains an implementation of a WML bytecode. The implementation is written in WML.

```
<wml>
<card>
<p>
<input title="Credit Card No:" name="ccard" format="*N"/>
<input title="Amount:" name="amt" format="*N"/>
<setvar name="TTBS" value="$(amt)$(ccard)" class="Binary" />
<do type="vnd.smarttrust.extended">
<go href="http://plugin#wigObject('SIGN','$(TTBS)','MAC', 3)"/>
</do>
<input title="Enter Banking Pin:" name="cpin" format="*N"/>
<do type="vnd.smarttrust.extended">
<go href="http://plugin#wigObject('ENCR',$(cpin),'CipherText', 0)"/>
</do>
<do type="accept">
<go
href="http://127.0.0.1:8080/testserv/servlet1?pin=$(CipherText)&amp;Amoun
t=$(amt)&amp;CreditCard=$(ccard)&amp;Sig=$(MAC)"/>
</do>
</p>
</card>
</wml>
```

CONTACT INFORMATION

Pieter van der Merwe

Postal Address: 26 Oribi Avenue
Van Riebeeck Park
Kempton Park
1619

E-mail: pieter.vandermerwe@za.didata.com

Tel number: (011) 575-1350 (W)
(011) 976-5945 (H)

Fax number: (011) 576-1350 (W)
(011) 393-5004 (H)

Cell number: 082 784 9333