Mobile Banking in Developing Countries: Secure Framework for Delivery of SMS-banking Services

MASTER THESIS

Author: Abunyang Emmanuel
Student Number: s0535249

Radboud University Nijmegen.
The Netherlands
Security of Systems
Supervisor:
Prof. Dr. Bart Jacobs

August 2007

Abstract

The cost of delivering financial services in both developing and developed countries has always been an aspect of concern to financial institutions. Financial institutions incur exorbitant operating costs in the course of providing services to their clients. These costs definitely in the end trickle down to the bank client and translate into a draw back to the number of clients an institution attracts. On top of this the inconvenience to the client in terms of time delays and access is also a fundamental issue. While developed countries have a developed Internet infrastructure that has expedited development and pervasiveness of electronic banking services, developing countries have low access to the Internet. Take for example in Uganda internet connectivity is at level 1.5 given a scale of 0 to 4 according to Minges et al (2001) [18]. Thus with the diffusion of mobile telephony taking the case of Uganda cellular subscribers have increased from 3000 in 1996 to over 2.3 million by 2006. Today researchers are working at developing more low cost and secure mobile banking services to suit developing countries. This has led to the development of short messaging service (SMS) as a mobile banking conduit by banking institutions take for example the centenary Bank in Uganda. SMS is considered a globally accepted wireless service initially adopted and developed for use in the GSM system. It enables transmission of alphanumeric messages between mobile subscribers and external systems. However questions about data confidentiality, user authentication and data integrity arise. In this thesis we investigate , analyse and propose a prototype implementation that takes into account these security issues. Hence we present a secure model for SMS mobile banking services tailored to suit mobile cellular phone users. We give conclusions about application of SMS banking services in developing countries and future trends.

Acknowledgements

I would like to express my gratitude to all the people who helped me make this thesis possible. My special thanks go to my supervisor Professor Bart Jacobs of Radboud University Nijmegen for his excellent guidance and support throughout the project. I also extend my gratitude to Dr Martijn Oostdijk for his valuable ideas and suggestions during the project. In deed without them this thesis would not have been possible.

I also wish to register my appreciation to Professor Theo van der Weide, Nicole el Moustakim and all the staff of external relations for all the encouragement and timely assistance offered

I would also like to extend my gratitude to NUFFIC and the coordinators of ICT capacity building project in Uganda and the Netherlands for giving me the opportunity of studying this masters program in Radboud University Nijmegen.

Finally to my family, I dedicate this work to my wife Everline and kids (Martha, Ludwig and Mary) who have had to bear with my absence throughout these two years of study. To them I say thanks for being on my side.

Contents

List of Figures

Summary

This thesis explores the current technological and security aspects in mobile banking systems. We review a number of systems offering mobile banking services and highlight their technologies, services and security implementations. We use insights from these reviews to construct a secure frame work for delivery of SMS banking in developing countries taking Uganda as our domain of study.

In our research we focus on how to achieve security of banking information used in SMS banking transactions because of varying degree of threats and resource constraints on some components like memory in mobile cellular phones. We achieve this by seeking to answer a number of questions;

1. What are the enabling technologies for mobile banking using a cell phone?
2. What are the security concerns on the enabling technologies used?
3. What security measures are currently deployed with these technologies?
4. What is the appropriate model applicable for developing countries?
5. What are the observed requirements, limitations and challenges of this mobile banking application in the context of the developing countries?
6. What are the possible solutions?

We present an overview of GSM network and its security limitations. We further explore various authentication protocols relevant to our study and review them to come up with our proposed protocol. Based on security analysis tools we present a threat model using attack trees to give a security guarantee of our scheme. The current systems have focused on cost implications at the expense of security. Hence a number of issues like confidentiality and integrity of the message including authentication still need attention. We therefore present a prototype implementation that demonstrates how these security aspects can reliably be achieved in a SMS mobile banking system.

Our prototype provides a mechanism for authentication, encryption and decryption for purposes of confidentiality and processes message digest for integrity checks. We end up by giving a conclusion about secure SMS mobile banking in developing countries and future work.

Chapter 1

Introduction

The way we live today is so much influenced by computing technologies. Computers control the economy, transportation, banking and many other functions. This development has made information attractive to criminals because of the economic value of such information. The advent of the Internet and wireless communication is believed to particularly have opened an entire new area of crime. The European cyber crime treaty has drawn a criminal policy aimed at protecting society against cyber crime by deterring and prosecuting actions directed against the confidentiality, integrity and availability of computer systems, communication networks and computer data [1]. This indicates the extent to which authorities are getting prepared to fight cyber crime in society.

Internet and mobile technologies are increasingly being adopted and utilised in the banking industry; this has reshaped the consumption of financial services [2]. In this research we analyse the security of electronic banking services with an emphasis on mobile commerce transactions with a focus on mobile banking using mobile devices specifically cell phones. Electronic banking is considered a way of delivering banking services through the internet to the consumer at a reduced cost to the banking industry and improved convenience to the customer [4]. However there exists a low internet connectivity in the developing countries given the costs of connection especially in rural areas and yet banking services need to be brought closer to the population to enhance development [16]. A viable solution here is mobile banking. Here we are interested in what the implications are in the terms of security and also in the economic viability of these technologies in developing countries.

Mobile commerce shall be defined as commercial transaction activities carried out via communication networks that interface wireless or mobile devices. A mobile device is a device used to connect to a mobile service for example cell phones and Personal Digital Assistants (PDA). The high diffusion rate of mobile phones coupled with the stability of mobile communication technolo-

gies have greatly contributed to the enhancement of mobile banking solutions in the provision of financial services in the world [5]. Mobile banking is considered as a service that enables users to receive information regarding the status of their accounts, to transfer among bank accounts, to facilitate stock trading and direct payment confirmation using mobile devices.

A number of enabling technologies are being used in the delivery of mobile banking service applications. They include Interactive Voice Response (IVR), Short Messaging Service (SMS), Wireless Access Protocol (WAP) and stand alone Mobile Application Clients (MAC).The goal of this research project is to analyse these enabling technologies and applications that enhance mobile banking trustworthiness in order to investigate the security limitations and challenges and to propose possible solutions to mitigate them. We will specifically look at the SMS enabling technology because its the most cost effective service suitable for a developing country. However it has a number of security limitations for example when authorising a bill payment the format is Account Number, PIN and amount [6]. Because these messages are not encrypted, prone to human error and normally telecommunication companies keep a copy of these messages in their servers they are quite an easy target for criminals.

The intention of this research is to propose and implement some measure that can be used to offset these limitations given the computing restrictions of the ordinary cell phone which are predominantly used in the developing countries. In chapter one, we discuss the GSM system which is an important data transmission media used in delivery of SMS messages. Chapter three discusses the cryptographic theory used in realization of security features in this project. In chapter four we present our proposed protocol and analyse its security capabilities in chapter five. We finally present the prototype implementation in chapter six and conclude in chapter seven.

1.1 Motivation

In mobile banking data is electronically transmitted over wireless communication channels and the Internet. These processes raise issues of how users are authenticated, how integrity of data is maintained and importantly the confidentiality of this data. Considering the low extent of development of ICT in developing countries when compared to the developed countries electronic banking has not really been able to diffuse into society given the low rate of Internet access[15,16]. However the advent of mobile telephony has seen the widespread adoption of cell phone usage this makes mobile banking in the developing countries a very attractive service for the banking industry

for example in Uganda mobile cellular subscribers have increased from 3000 in 1996 to over 2.3 million in December 2006 [3].

1.2 Research Questions

Main question.
What are the security considerations and challenges for a secure mobile banking service framework and how can the risks be managed?

Sub-questions:

1. What are the enabling technologies for mobile banking using a cell phone?
2. What are the security concerns on the enabling technologies used?
3. What security measures are currently deployed with these technologies?
4. What is the appropriate model applicable for developing countries?
5. What are the observed requirements, limitations and challengers of this mobile banking application in the context of the developing countries?
6. What are the possible solutions?

Chapter 2

Background of Related Systems and Security Issues.

In this chapter we discuss the GSM functionality and its security deficiencies. We also describe the present enabling technologies for mobile banking and their inherent security limitations.The Global System for Mobile Communication network (GSM) is the data transmission media in the SMS protocol and the flaws in its architecture have lead to the security shortfalls in SMS-banking system. It is therefore important to provide an overview of its architecture. The GSM technical specification [7] elaborates the architecture of the GSM system. The specification was started in 1982 by the European conference of postal and telecommunications administrations (CEPT) [26]. The GSM system offers the user the ability to be mobile. The subscriber identity module (SIM) is used for purposes of authentication. It gives the network operator on whose behalf the SIM has been issued the complete control over all subscription and security issues.

2.1 GSM Architecture.

This section briefly describes the functionality of the various components illustrated in figure 2.1. The GSM comprises of various components in the figure the solid lines show communication between core components. The dotted lines show the internal connection for communication used during maintenance. In a typical communication operation the Mobile Station (MS) which is in effect a cellular handset initiates the communication. The communication signals are transmitted from the MS and received by the Base Transceiver Station (BTS). The function of the BTS is to receive and transmit radio signals to and from the MS. It is also responsible for translating the radio signals into digital format and transferring them to the Base Station Controller (BSC). The BSC forwards the received signals to the Mobile Switching Centre (MSC). The MSC interrogates the Home and Visitor Location Registers (HLR and VLR) this databases keep information about

location of the destination MS. In the event that the received signal is an SMS message then it is routed to the Short Massage Service Centre (SMSC) for delivery to the required destination. The SMSC keeps a copy of the sent SMS in its database after it has been sent. In case of an international connection the signal is routed through the International Switching Centre(ISC). In order to facilitate equipment verification and user authentication the Equipment Identity Register and Authentications Register database are used. The operation and management centre controls maintenance operations

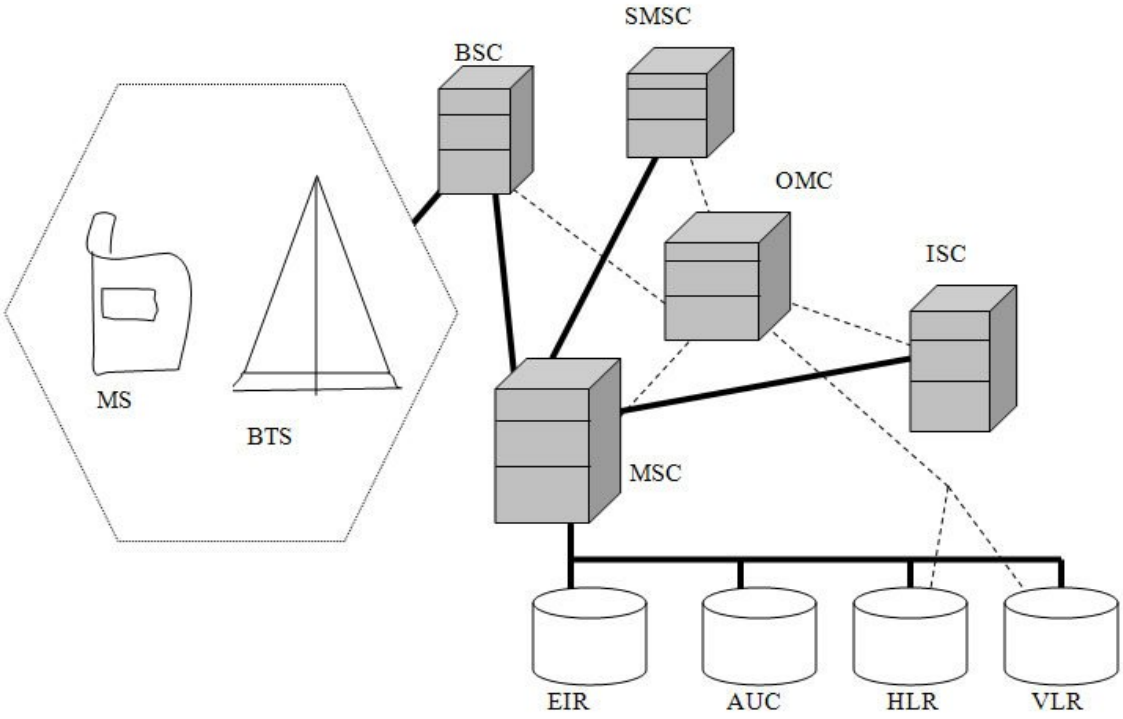

Figure 2.1: GSM Architecture

Key: Giving Full Meanings of GSM Architecture Component Abbreviations

MS – Mobile station	EIR–Equipment Identity Register
BTS – Base Transceiver Switch	AUC– Authentication centre
BSC – Base Station Controller	HLR – Home Location Registry
MSC– Mobile Switch Centre	SMSC–Short Message Service Centre
ISC – International Switching Centre	OMC– Operation Management Centre
VLR- Visitor Location Registry	

2.2 GSM Security.

In order to protect network operators and users the GSM system specification provides implementation of a number of security features. The following features are taken from the perspective of the user.

- Subscriber identity confidentiality
- Subscriber identity authentication
- User data confidentiality

2.2.1 Subscriber Identity Confidentiality

The GSM system uses temporary identities to deny an intruder the possibility of gaining information on the resources used by a subscriber for example preventing the tracing of users location and matching a user with the data transmitted [27]. The system uses the International Mobile Subscriber Identity (IMSI) number to uniquely identify subscribers. The identity is stored in the Subscriber Interface Module (SIM) card issued by the mobile network service provider. The mobile cellular phone operates only with a valid SIM. During subscriber verification it is desirable to keep the subscribers IMSI number secure in order to prevent the adversary from localising and tracking the users physical location. To achieve this instead of transmitting the IMSI in plaintext a temporary IMSI is used called TIMSI [27].

The TIMSI is valid per session of subscriber verification and it is unique within each location area where the user moves. The Location Area Identity (LAI) is always used in conjunction with the TIMSI. The TIMSI, IMSI and LAI are securely stored in the VLR database of the mobile network service provider. In order to establish subscriber identity the service provider uses the IMSI of the subscriber to interrogate the VLR database for the subscribers TIMSI. The service provider compares the TMSI with the mobile phones IMSI for authentication. The TIMSI is first encrypted before it is sent over the air. Following a successful authentication the server at the service provider sends the next TIMSI to the subscribers mobile set for the next authentication.

2.2.2 Subscriber Identity Authentication

Subscriber authentication is of major interest to each operator. It's purpose is to protect the network against unauthorised use thus preventing masquerading attacks [27].The protocol utilizes a challenge response authentication mechanism. Authentication of subscriber identity is achieved using

a subscribers authentication key K_i in addition to the IMSI. The key is installed in the SIM card and pre-stored in the Authentication Centre (AUC) by the service provider. The algorithm used for authentication is A3 and it requires both the subscribers mobile phone and service operator to have the same key K_i. In order to perform an authentication the mobile network provider generates a random number (RAND) that is used to calculate the signature response (SRES). This random number is sent to the mobile subscribers phone as well and it calculates the signature response (SRES) using this number and sends it back to the service provider. If the two SRES values are Identical the authentication is considered successful Figure 2.3 illustrates the basic flow diagram of subscriber authentication.

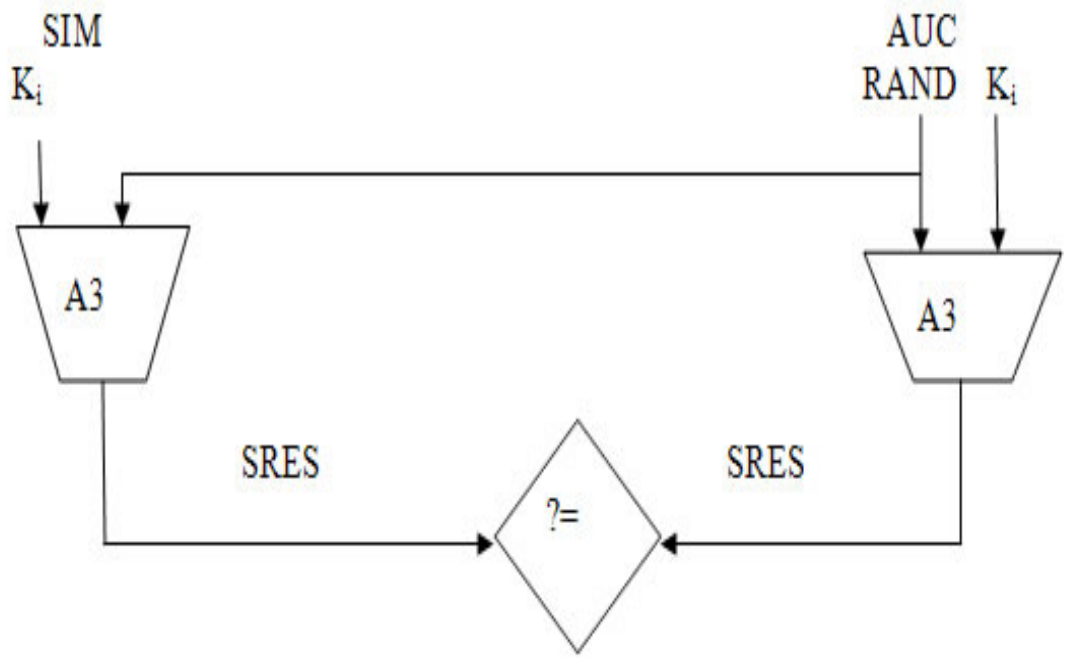

Figure 2.2: Authentication Procedure

2.2.3 User Data Confidentiality.

To ensure the privacy of user information carried in both traffic and signalling channels is upheld, the GSM system employs the A5 algorithm for encryption of transmitted data. The activation of this service is controlled by the mobile network service provider. It is initiated by the base station by sending a command to the subscribers mobile phone [27]. The A5 algorithm is a symmetric ciphering algorithm and uses a key K_c to generate a key

stream that is *XORed* with a block of plaintext to generate the cipher text. The key is derived in the SIM using the A8 algorithm generator specific to a network operator and also the RAND and K_i used in subscriber authentication procedure Figure 2.4. The ciphered text is converted to radio signals by the mobile phone and sent across the air to the base transceiver station. The figure below illustrates the flow diagram of cipher key generation and enciphering.

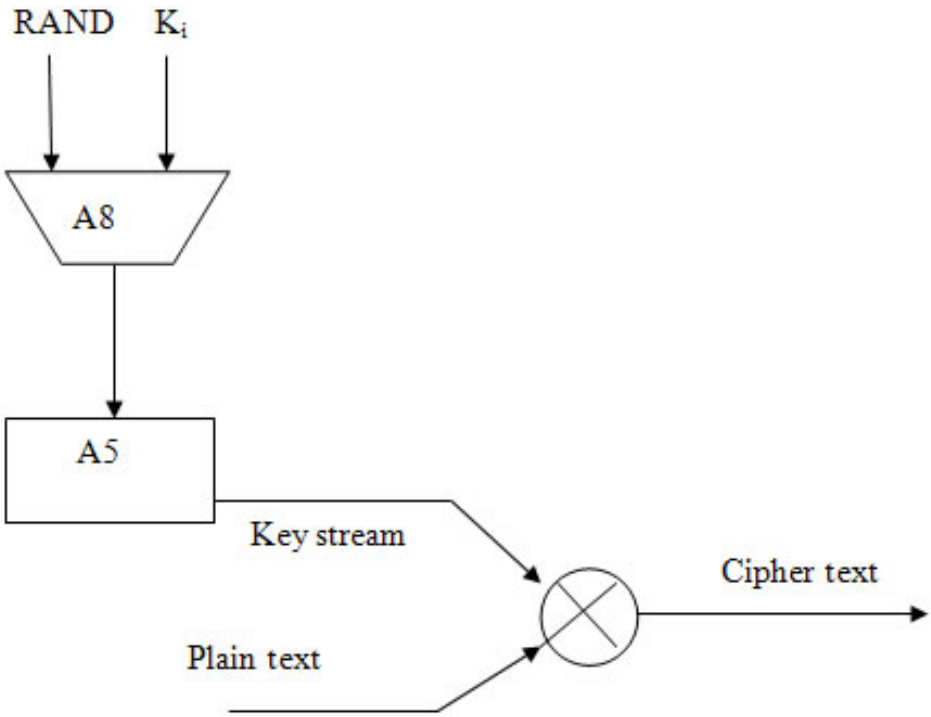

Figure 2.3: Cipher Key Generation and Enciphering

2.3 Security Deficiencies of GSM Architecture.

Much as GSM system strives to make a provision for security services as discussed in the previous sections it still has limitations in its security. Tasneem et al (1998) [17], point out the lack of data integrity in the GSM. On top of this the following cryptographic issues with regard to the authentication and encryption algorithms have been identified.

2.3.1 A5 Encryption Algorithm.

According to Steve (2003) [31], the commonly used A5 algorithm for encryption in the GSM system has already been reverse engineered. The A5 algorithm has two main variants A5/1 and A5/2. Biryukov et al [32], discovered three possible attacks on the A5/1 version that is commonly used in Europe. The attack could be achieved with a personal computer in a few seconds. The A5/2 variant was also cracked in less than a day [33]. This shows how the GSM system is vulnerable to cryptanalysis attacks.

2.3.2 A3/A8 Authentication Algorithm.

The A3/A8 algorithm is a commonly used authentication algorithm through out the world in GSM systems. However Wagner[1] et al have successfully shown that this algorithm can be broken after sporting several flaws in the algorithm. They were able to obtain the key K_i hence making SIM cloning feasible.

2.4 Enabling technologies for Mobile Banking.

In this section we discuss current channels through which mobile banking services are deployed. We point out the short comings of each in relation to the others. These banking services may include any of the following;

- Credit/Debit Alerts.
- Minimum Balance Alerts.
- Bill Payment Alerts.
- Bill Payment.
- Recent Transaction History Requests.
- Information Requests like Interest Rates/Exchange Rates.
- Account Balance Enquiry
- Account Statement Enquiries.
- Cheque Status Enquiry.
- Cheque Book Requests.
- Fund Transfer between Accounts.

[1] www.isaac.cs.berkeley.edu/isaac/gsm.html

2.4.1 Short Messaging Service (SMS)

SMS utilises the text messaging standard to enable mobile application based banking. It provides a mechanism for transmitting short messages to and from wireless devices [30]. The client requests information by sending an SMS containing a service command to a pre-specified number. The bank thereafter responds with a reply containing the specific requested information. An SMS service is hosted on an SMS gateway that connects to a mobile service providers SMS centre. The major shortcoming of SMS service bank transaction is that it has not taken root because of security concerns.Fortunately this is what this research is out to achieve. One attractive side of deploying mobile banking applications on SMS is that almost all mobile phones even those that are cheap are SMS enabled.

2.4.2 Wireless Application Protocol (WAP)

WAP is an open international standard that uses wireless communication to bring internet content and advanced data services to digital cellular phones [29]. It uses a concept similar to that used in internet banking. It requires banks to maintain WAP sites which clients access using WAP compatible browsers on their mobile devices. WAP sites are written in UML (Wireless Mark-up language) as opposed to traditional HTML, XML or HTML languages. The mobile application client, resident on the clients mobile device, accesses the banks site through the WAP gateway in order to carry out bank transactions.

2.4.3 Interactive Voice Communication (IVR)

Interactive voice response service requires clients to call pre-specified numbers in order to access a banking service [28]. The client makes a call to the IVR number and is usually greeted by a stored electronic message followed by a menu of different options. The service utilises mostly a text to speech program. IVR has its own short coming in that it is expensive when compared with other channels like SMS or data transfer since it involves making voice calls.

2.4.4 Standalone Mobile Application Clients (MAC)

Standalone mobile application clients are quite desirable for complex banking transactions like trading in securities. They are customised according to the user interface complexity that is supported by the phone. Mobile applications clients are downloaded into the clients mobile device and thus require the device to support development environments like J2ME [8]. The main short coming with standalone mobile application clients is that the application needs to be customised to each mobile phone on which it is to

be run. J2ME uses profiles to categorise functionalities of API for mobile phones. However the rapid development of mobile cellular phones supporting different functionalities has resulted into many profiles hence driving up development costs of mobile application client based systems.

2.5 Current SMS Banking Services in Uganda

Currently in Uganda the SMS banking service is being used by the Centenary Bank[2]. It offers clients the ability to access inquiry services like bank balance and mini statements using their mobile phones. The client sends a request in plaintext format to the bank server and the server responds with the required information.

2.5.1 Security Limitation with the Current SMS Approach

In this subsection we discuss the security short comings of the SMS banking service. During the conception of the GSM system the SMS service was designed for subscribers to send non-sensitive messages across the GSM network. Security considerations in terms of mutual authentication, data confidentiality, end to end security and non-repudiation were omitted with regard to the SMS service.

2.5.1.1 Message Spoofing

In this attack the adversary sends out SMS messages that appear to be from a legitimate sender by forging the originators address [31]. By altering the originators address field in the SMS message header to another alpha numeric string an adversary can perform masquerading attacks.

2.5.1.2 SMS Encryption

The default data format for SMS messages is plaintext. The GSM system offers encryption only between the mobile phone and the Base transmission station end to end security is currently not available. Besides the A5 algorithm used has been proved vulnerable [31].

2.5.1.3 SMS Service Centre Attack.

The storage of copies of SMS messages at the SMS centre server hosted by the mobile network service provider also provides a point of vulnerability to the SMS banking service. Since the message is in plaintext then any personnel

[2]http://www.centenarybank.co.ug/

who have access to the service providers SMS centre server can easily view sensitive details.

Chapter 3

Theory Concerning Cryptographic Security Mechanisms.

In this chapter we present the relevant theory used in providing required security in our project. In our prototype implementation the goal was to secure the value of information that is being transmitted in line with the elements of security that constitute secure communication [10]. These are; data integrity to ascertain that message contents are not altered, data confidentiality ensuring that transmitted data is kept confidential, authentication ensuring parties be able identify and verify each other and non-repudiation that is, the sender should not be able deny having sent a message. In the next sections we describe the theory against which this security functions are achieved.

3.1 Basic Cryptography.

In this section we give a formal description of a cryptosystem that includes the specification for its message, key , cipher text, encryption and decryption functions. There are two broad classes of cryptosystems symmetric (shared key) and asymmetric (public key) [19,11]. In the former a single key that should be kept secret is used for both encryption and decryption. And in the latter two different keys are used one for encryption and the other for decryption. However the encryption key can be made public while the decryption key must be kept secret.

3.1.1 Formal Description of Symmetric and Asymmetric Cryptosystem.

A crypto system is symmetric in the sense that both the message transmitter and receiver share a secret key for encryption and decryption. Whereas asymmetric the transmitter and receiver keys are not only different but it is also computationally infeasible to derive one key from the other. This formal description applies to both symmetric and asymmetric crypto system. Let M denote the message space and m a message in the space. The set of encryption and decryption key pairs be denoted by KE and KD. Where $[u]v$ denotes the encryption operation on message u if v is an encryption key and the decryption operation on u if v is a decryption key.

$$\forall\, m \,\epsilon\, M : \forall\, (k, k_{-1}) \,\epsilon\, KE \times KD : [[m]k]k_{-1} = m.....Equation1$$

Considering the symmetric cryptosystem which has identical encryption end decryption keys the operation is quite clear from the context of the formal expression.

The difficulty of distributing keys is the major limitation on the use of conventional symmetric cryptosystems. The transmitter and receiver must be prepared to wait for a key to be sent through a secure channel for example through registered mail. In large commercial networks the problem becomes enormous since the number of connection grows as $(n^2 - n)/2$ taking n as the number of users [20]. This implies that the cost of distributing the keys becomes extremely prohibitive. A widely used cryptosystem that have been used to solve this problem is the RSA (Rivest-Shamir-Adelman) asymmetric system [22]. The encryption and decryption of data in an RSA system satisfies the following additional property [20] given in the expression below.

$$\forall\, m \,\epsilon\, M : \forall\, (k, k_{-1}) \,\epsilon\, KE \times KD : [[m]k_{-1}]k = m.....Equation2$$

That is suppose k and k_{-1} are Bobs asymmetric keys then $[m]k_{-1}$ is used as Bobs signature on m since it can only be produced by Bob, the only principal that knows k_{-1}. Principals with knowledge of k can readily verify Bobs signature. Therefore (equation2) illustrates the commutative principle of asymmetric key crypto systems.

3.2 Authentication

Authentication involves identification and verification; it plays an important role in ensuring data secrecy [23]. Identification is a process by which an entity claims a certain identity and verification involves checking the claim of identity. The goal of any authentication is to verify identity of a principal. Principals are considered communicating entities in a distributed system.

3.2.1 Types of Authentication

Lam et al [23] point out three types of authentication this include;

Message content authentication where the content of a message received is verified to be the same as that one sent.

Message origin authentication that involves verifying that the sender of the received message is indeed the one recorded in the message field.

Finally we have identity authentication that involves verifying the principals identity as claimed. We employ all this three approaches in our protocol.

3.2.2 Approaches to Authentication.

Authentication procedures entail checking known information about a claimed identity against information supplied by the claimant during the identity verification procedure. This process can be classified in the following three approaches [23].

Proof by knowledge where the claimant demonstrates knowledge of some information regarding the claimed identity that can be known or produced by a principal with the claimed identity. Some good examples that demonstrate proof of knowledge are typing of a password or computing replies to challenges by a verifier.

Proof by possession where the claimant produces an item that can only be possessed by a principal with the claimed identity.

Proof by property. A good example of this approach is use of biometric techniques for authentication like the fingerprint, retina print etc. The verifier measures this claimant properties.

3.2.3 Freshness Assurance of Authentication.

Information transmitted over communication lines is vulnerable to security attacks such as eavesdropping and tampering. Use of cryptographic techniques helps protect against unauthorised information disclosure and detecting unauthorised modification of information. However an intruder can still record a message and then replay it after sometime. The concepts of time and challenge response operation are important in assuring freshness of messages in applications vulnerable to replay attacks [24]. Freshness implies a message is not replayed. The subsequent two subsections discuss these two concepts.

3.2.3.1 Clock Based Authentication

In order to achieve freshness in distributed systems timestamps are utilised. A principal accepts a message as fresh only if it contains a timestamp which is close enough to its knowledge of current time [24]. However one draw back about time stamp based approach is that it assumes the presence of a globally accessible clock that has to be reliable and available. This introduces need for synchronising clock protocols that in themselves cause extra delays and security vulnerabilities, thus leading to the recipient accepting messages that are not exactly the same as that of its local clock.

This time difference is refereed to as an acceptance window and its determined by system parameters such as message delivery delays and performance of clock synchronisation protocols. The acceptance window is assumed to be large enough to allow fresh messages to be accepted correctly by their recipients and yet small enough to allow for detection of replay attacks. However intruders can take up this opportunity to perform a replay attack [25]. Therefore the acceptance window must not be larger than the possible replay time which is often realistically difficult to attain. This therefore suggests that timestamp approach is suitable in less security sensitive environments.

3.2.3.2 Authentication by Challenge /Response

Lam et al [24] , proposed a major authentication approach that assures freshness of messages in the form of challenge/response operation. In this approach a principal say Bob expecting a fresh message from another principal say Alice sends a random number to her. This random number is required to appear in the subsequent response message received by Bob. The quality of the random number is an important attribute in this approach and its the responsibility of the principal to ensure quality. Acquiring truly random challenges can be done using random generators.

3.2.4 Secure Hash Function

The main role of a cryptographic hash function is in the provision of message integrity checks. A hash function (H) is a transformation that takes an input message m and returns a fixed size string which is called the hash value h (that is $h = H(m)$) [10].

William Stallings (2003) [10], points out important properties of hash functions as follows.

- Input can be of any length.
- The output has fixed length

- $H(x)$ is relatively easy to compute for a given x.
- $H(x)$ is one way.
- $H(x)$ is collision free.

A hash function is said to be one way if it is hard to invert. This implies that it is computationally infeasible to find some input y such that $H(y) = h$

Chapter 4

SMS Banking Proposed Secure Model.

In this chapter we discuss the proposed SMS banking secure model to mitigate inherent security limitations of the current existing SMS banking systems. We present the proposed solution with the authentication and message exchange protocols as defined in our system. Our focus is on creating a special Java program called a Mobile Information Device Application (MIDlet) that offers a secure message transmission.

4.1 The Solution.

The solution to offset the limitations mentioned in chapter one is to use SMS for encrypted and signed messages. This solution will specifically apply to Global system for Mobile Communications (GSM) networks to enable consumers securely transmit banking information. SMS was specifically intended for sending non sensitive information in a GSM network therefore in its implementation security considerations in terms of confidentiality, integrity, non-repudiation and authentication were not catered for [7]. In general end to end security measures are not implemented.The solution comprises of three components namely the mobile application, bank server and the backend database. This is illustrated in the figure 4.1.

4.1.1 Mobile Application Component

The major focus of the research was to develop a MIDlet application resident on a clients mobile device. The mobile device captures all the security information from the user. This information comprises of the users banking

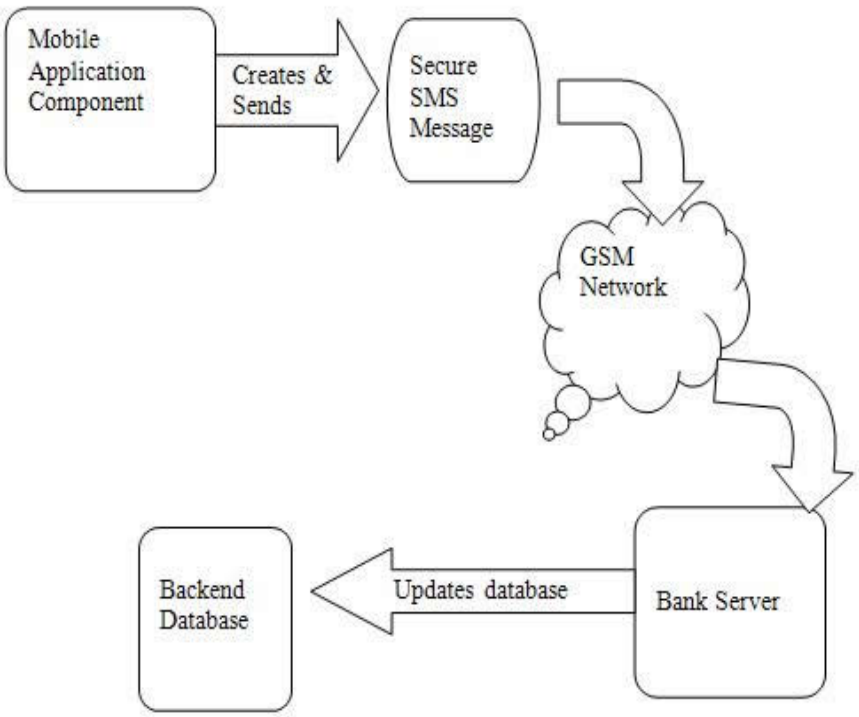

Figure 4.1: Overview of the Solution

details and is used in generating the secure SMS message that is sent to the server.

Message digest is used to ensure integrity of the message. To ensure confidentiality symmetric encryption is employed. On completing processing the secure message it will be sent to the server through the GSM network.

4.1.2 Bank Server

This component was developed to facilitate the testing of the MIDlet application. It is responsible for receiving and decoding the secure SMS message. The server will check to ascertain that the message is suitable for a secure SMS protocol. It will then proceed to check for the account identifier from the message and find out if the identifier exists in the server database. After the above check the server decrypts the message using the one time password. The password will be discarded when the decryption is successful.

To ascertain message integrity the server calculates the message digest from the decrypted message using the same algorithm used by the mobile application. It then compares the two digests for message integrity. When this checks are found to be positive by the server, it proceeds to retrieve the

PIN (account holders password) from the message and compares it to the account holders PIN from the servers database. The transaction will finally be performed when all the security checks have passed.

4.1.2.1 Back End Database

The backend database serves as a store for the users banking and security details. The communication between the bank server and the backend database will be independent from the one between the mobile device and the bank server. In this research we specifically concentrated on the security of communication between the bank server and the mobile application.

4.2 Message Format and Authentication Protocols

In this section we discuss the authentication and message protocols used in the secure SMS banking model. We also present the message format used in the protocols.

4.2.1 Message Structure.

In order to be able to explain the SMS secure protocol its important to specify the message format. The format of an SMS message packet provides for four bytes to specify meta data and the size of the payload. The maximum length of the payload is 160 characters at 8-bits per character adopted from (Clements 2003) [9]. The message structure is shown in figure 4.2.

Using the message format in figure 4.2 we derive the following secure SMS message structure. The secured message comprises of multiple fields for various security checks required by the protocol. The message structure will have a generic format as follows.
$< Version >< AccID >< Seq >< SecureMessage >$

- In order to ensure the bank server does not receive messages that are not intended for it a version number is used with a specified bytes pattern.
- The AccID comprises the account identifier of the user.
- The seq will specify a sequence number of the one time password.

The secure message field includes the following subfield formats.
$< EncryptedTextLength >< EncryptedBankingDetails >$
$< Digestlength >< Digest >$

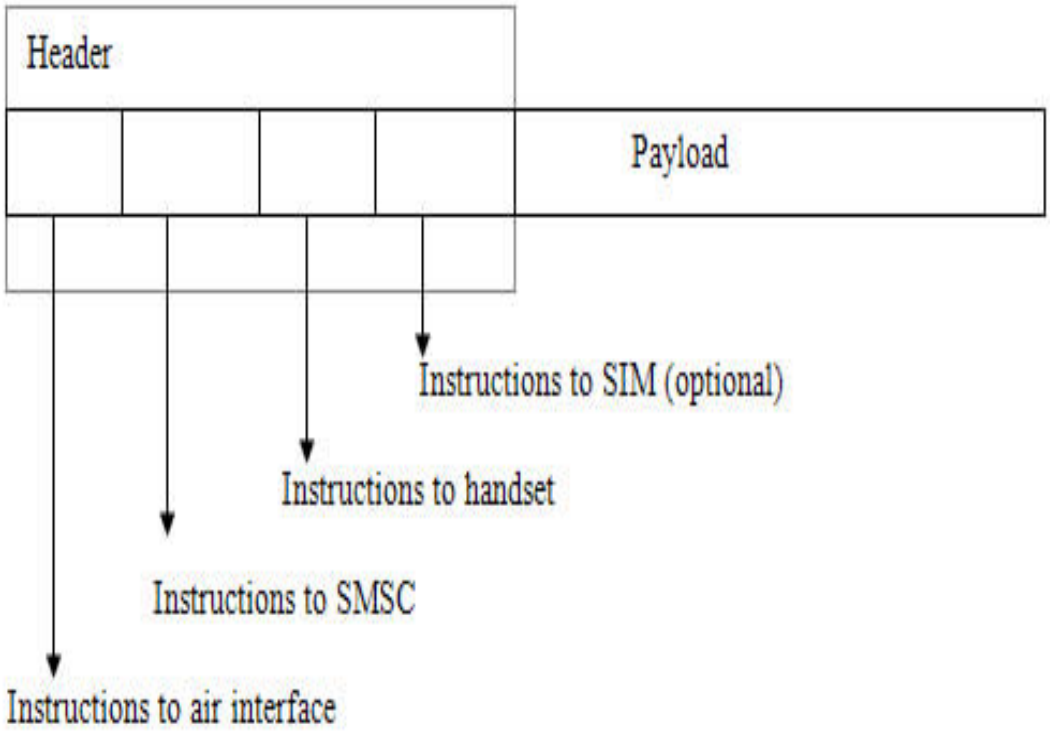

Figure 4.2: SMS Message Structure.

- The Encrypted text length gives fixed maximal length with padding of the ciphered message.
- Digest length also gives the number of next bytes that comprise the message digest.
- The digest is for the server to check for message integrity. In the protocol the digest will comprise the following fields *Password , PIN, Type of Transaction and Transaction payload.*

Further the Encrypted banking details field will comprise of the following subfields
$< PIN >< TypeofTransaction >< TransactionPayload >$
$< EncryptionPayload >$

- In order for the server to authenticate the user the PIN comprising of the users predefined password is used.
- In this protocol we specify two transaction types though more types could be added.The transaction types are check balance and transfer among accounts.The Type of Transaction field is used by the server application to identify the type of transaction to be performed.

- The Transaction Payload depends on type of transaction and comprises of extra data used for the transaction.For the case of the money transfer and check balance they will have the following fields respectively.
  $< DestinationAccount >< Amount >< RandomBytes >$
  $< Randombytes >$(No transaction payload required)

4.2.2 Review of Possible Handshake Protocols.

Ratshinanga et al [10] suggested a handshake protocol that uses the public and session symmetric key and authentication cryptography strategy.The following denotations are used in the explanation of the protocol.

- BS: - Denotes the Bank Server.
- MC: - Denotes the banks Mobile Client.
- BPK_{pub}: - Denotes the public key of the bank server.
- BPK_{pri}: - Denotes the private key of the bank server.
- R_{MC}: - Denotes the random challenge response generated by the mobile client.
- R_{BS}: - Denotes the random challenge response generated by the bank server.
- SK: - Denotes the symmetric session key shared by the bank server and mobile client.
- SQ: - Denotes the sequence number (nonce) generated by the mobile client starting at 1 on commencement of the protocol.
- Slt: - Denotes a salt value generated by MC its used by bank server and the mobile client to generate the session key (SK).

$||$: - Symbolises concatenation.
The handshake protocol is defined as follows;

$$M1 : C - S : E_{BPKpub}[AccID||Slt||SQ||R_{MC}]$$
$$M2 : S - C : E_{SK}[R_{MC}||R_{BS}||SQ] \text{ where } SQn > (SQn - 1 + 1)$$
$$M3 : C - S : E_{SK}[R_{MC}||R_{MC}||SQ] \text{ where } SQn > (SQn - 1 + 1)$$

In order to initiate the connection the mobile client sends its account identification and a salt number[1] to the bank server.The sent message is

[1]Salt number is a-bit random string used to obscure a password. It makes dictionary attacks on passwords less effective [11].

encrypted using the bank server public key. The bank server decrypts the message using its private key upon reception and retrieves the salt number and account number from the message. The bank server thereafter retrieves the users PIN from its database. This data is then used to generate the session key and a secure connection is established. The session key is generated individually by hashing the account identity, salt number and the shared PIN. The sequence number is used to guard against replay attacks. The sequence number is incremented each time a message reaches its destination. A message is discarded if the sequence is not followed. The purpose of the random challenge is to assure the freshness of authentication [12]. The R_{MC} random challenge used to ensure that a client is not able to perform an attack on the protocol by replaying previously used messages of a legitimate protocol.

Lam et al (2003) [13] also present a light weight security protocol for mobile commerce platforms discussed below. In the discussion the following denotations are used.

- S:-denotes the server side of the security protocol;
- C:-denotes the client side of the security protocol;
- E_{KS}: - denotes the public key of S. We assume that the public key of S is made known to C prior to the execution of the protocol;
- D_{KS} :-denotes the private key of S;
- PIN: - is the password of user C which is known only to C and is verifiable by S. The password typically has a length of eight to sixteen printable characters;
- R_a: - is a nonce identifier (a random number) generated by S;
- $(R_{b1} and R_{b2})$:- are nonce identifiers (random numbers) generated by C;
- SN: - is a sequence number generated by S for this protocol run;
- SK: - is the symmetric session key shared by S and C for protecting transaction messages in the session;

The protocol commences when a mobile client wishing to perform a commercial transaction sends an authentication request to a server. The following protocol between the server and the mobile client is then carried out to establish a shared secret key that is used for the subsequent secure communication for the session. The handshake protocol is defined as follows;

$M1 : C - S$: I am C
$M2 : S - C : R_a$
$M3 : C - S : E_{KS}[Ra, R_{b1}, PIN], E_{KS}[Ra, R_{b2}, SK]$
$M4 : S - C : E_{SK}[SN, R_{b2}]$

The description of the protocol is as follows; In message (M1) the cleint declares intentions to communicate to the server, message (M2) the server sends a random challenge R_a as an assurance for protocol freshness to the client. Message (M3) the client returns a two message response. The first contains the secret password (PIN) known only to the client and server. The PIN allows the server authenticate the client by verifying the correctness of the password. The secrecy of the PIN is enforced by encrypting the message using the public key of the server E_{KS}. The server checks the value of R_a to ensure that the message is not a replay of some previous protocol message. Since R_a is sent in clear form a nonce identifier R_{b1} is included to prevent brute force attack on the message to retrieve the PIN. In the second part the session key (SK) is generated by the client C. The nonce identifier R_{b2} is included to allow the client to confirm that the server knows SK because D_{KS} is only available to the server. In message (M4) by encrypting the nonce identifier R_{b2} with SK, the server proves to the client it knows the shared key. The communication thereafter proceeds using the shared key SK.

In both protocols presented above its quite clear the cost implications of the communication is a big concern. This is because the key exchange involves use of three and four messages per transaction respectively implying a huge cost to the customer. The protocols do not use the concept of the time stamp to ensure the protocol freshness because not all mobile phones will have their clocks coinciding with that on the server.

4.3 Proposed Authentication and Message Exchange Protocols

In this section we present our proposed authentication protocols and give insights on its key management aspects. Key management is a major issue when designing an authentication protocol. The method used for generating and storing potentially large numbers of client authentication keys and handling of authentication requests are of importance to proper running of a secure SMS banking system. In our proposed protocols we exclusively use symmetric key cryptosystem. The reason we have chosen the symmetric cryptosystem when compared to asymmetric that requires complex mathematical calculations is because of the need to optimise the cellular phone memory capacity. Given that the domain of this research is in a developing country most of the clients do not have access to high end modern phones

with improved memory capability. Therefore to ensure that our application is attractive to all clients irrespective of type of cellular device we employ symmetric cryptosystem. Key management is defined as the generation, storage, distribution deletion, archiving and application of keys [17]. We therefore describe key management aspects and assumptions employed in our protocols in the following subsections.

4.3.1 Key Generation

Brutch et al [17], point out two methods that may be used in key generation. These are by use of a random generator or by deriving them from the user related data with the help of an algorithm. In our protocols we utilise both methods.

The check balance and money transfer protocols involve use of a password generator. The password generator application is maintained and owned by the bank. The generator produces a list of random passwords with corresponding sequence numbers in advance. Sequence numbers protect against impersonation and replay attacks. It will be assumed that the list will be exchanged in a way that only the bank and the client have knowledge of them. The list will be written to a file and sent to the client using a secure communication media upon request.

In the check balance protocol the client encrypts the request with the randomly generated password. The reply from the bank is encrypted using a key derived from the concatenation of the clients PIN, salt number and account number. When the reply message is received by the client he or she is prompted to enter both the PIN and account number in order to access the massage content. The advantage here is that there is no need to store the derived key hence saving storage space.

In the event of a request to transfer money the client encrypts the request with the randomly generated password. The reply from the bank will follow the same procedure as explained in the check balance protocol.

4.3.2 Key Storage

The sequence and passwords generated by the password generator are stored in database owned and maintained by the bank. The password application generator is connected to the database that stores them as they are generated. Used sequence numbers and passwords are discarded from the

database after some defined time to minimise server space. On event that a client looses the list of passwords the client requests for a new list and the existing list in the database is discarded by re-setting and the new one stored.

4.3.3 Key management Assumptions

- A registered client of the bank is required to have a bank account and self select a personal identification number (PIN) only the client and bank should know this PIN.
- The user during time of registration receives the list of one time passwords and sequence numbers in a secure manner from the bank. This can be done by the client physically collecting them from the bank or the bank uses a secure delivery channel to send it to the client.

4.3.4 Application of Keys

In this subsection we present the message exchange and encryption protocols employed in our proposal. We derive separate protocols for the check balance and money transfer transactions.

4.3.4.1 Check Balance

In the check balance transaction a single message is sent to the server. The message comprises of an encrypted message m_1 using the one time password as discussed in section 4.3.1 and a message digest on message m_1 generated by a hash function. The protocol is presented below.

$$M1 : C - S : E_{sk}[m_1],\ H[m_1]$$
$$M2 : S - C : E_{AccP}[< balance >] \ or \ [< Errormessage >] \ \ inplaintext.$$

Where $m_1 = [AccID \,||\, Transaction\,Type \,||\, SQ \,||\, PIN]$, E_{sk} denotes encryption with the one-time password from the password generator and E_{AccP} denotes encryption using concatenation of the user account identification, salt number and PIN. H denotes a hashing function that generates a digest on message m_1. SQ denotes the sequence number of the password and the Transaction Type is a number specifying the transaction selected. AccID is the account identifier of the user and the PIN is the user predefined personal identification number. The reply message from the bank (M2) will automatically launch the mobile application requesting the user to input his/her account identification and PIN in order to access the message.

4.3.4.2 Money Transfer.

In the money transfer protocol a single message is as well sent to the bank server by the client. The message comprises of an encrypted message m_2 using the one time password as discussed in section 4.3.1 and a message digest on message m_2 generated by a hash function. The protocol is presented below.

$M1 : C - S : E_{sk}[m_2],\ H[m_2]$
$M2 : S - C : E_{AccP}[< confirmation >] or [< Errormessage >]$ in plaintext.

Where $m_2 = [AccID||DestinationAccID||payload||TransactionType||SQ\ ||\ PIN]$, E_{sk} denotes encryption with the onetime password from the password generator and E_{AccP} denotes encryption using concatenation of the user account identification salt number and PIN. H denotes a hashing function that generates a digest on message m_2. Payload is the extra data for the transaction, for example in this case it would contain the amount to be transferred. SQ, AccID, PIN and Transaction Type are as defined in the check balance protocol. Once a password has been used it will be considered as expired and the system will not recognize it if it is reused.

The confirmation can take the form of a plain text message as "Your transaction will take place within 5 minutes unless you send an authenticated cancel-SMS". The reply message from the bank (M2) will automatically launch the mobile application requesting the user to input his/her account identification and PIN in order to access the message.

4.3.5 Protocol Sequence.

We consider the protocol sequence in two parts. In the first scenario the mobile application generates the message and sends it to the server. The second scenario involves the security checks and message reception. The parts are elaborated in the following sections.

4.3.6 Secure Message Generation and Transmission.

Security information from the user is captured through the mobile application resident on the cellular phone. The information includes the account identification number, user PIN (personal identification number), sequence number and one time password from the user. This information is used to generate the secure SMS message that is sent to the server. In order to ensure message integrity on the receiving side message digest is used. The process of message digest occurs both at the user interface application and at the bank server. The digest comprises of the following fields $< Password ><$

$PIN >, < TypeofTransaction > and < Transactionpayload >$. To maintain message integrity the content used in calculating its integrity needs to be encrypted. This is to ensure that on event of message interception an attacker cannot generate another digest. The integrity validation check will fail if the original message is altered.

Further more in order for the server to be able to identify the account holders identity some identification details need not to be encrypted. The choice is dependent on the developer. Symmetric encryption algorithm is employed using one time password known to the user and server. The password is used to generate the key for encryption. On completion of the security processing of the contents the SMS message is sent to the server through the GSM network.

4.3.6.1 Secure SMS Message Reception and Decoding.

The server upon receiving the message through the GSM network decodes it according to the message structure described in section 4.2.1. The version bit pattern is checked and if found to be correct the message is considered suitable for the SMS secure protocol.

The server proceeds to read and check the account identification to find out if it exists in its data base. On event that the above checks are satisfactory the server retrieves the one time password from the data base. The password serves as a decryption key to decode the encrypted message.

When the decryption of the message is successfully accomplished the one time password is discarded and the server sequence counter incremented. The server reads the secured contents necessary to calculate the message digest. The server uses the same algorithm used by the mobile application to calculate the digest. The two digests are thereafter compared for message integrity by the server. The server also retrieves the account holder PIN from its database and compares it to that on the message. The requested transaction is performed if all the above security checks are satisfactory.

Chapter 5

Security Analysis of Proposed Protocol

In this chapter we perform security analysis of our proposed model. Current approaches of SMS banking systems use plaintext message for carrying out transactions. A highly motivated adversary can intercept this communication and gain access to modify important information. We observe that the goals of attacking an SMS banking system are to obtain the clients PIN and account number in order to fraudulently perform a banking transaction or simply read the balance or modify transfer information. In our system the adversary can be successful if only he or she can intercept the message and get to know the encryption key. In the following sections we identify stakeholders, describe security measures undertaken in our proposed protocol and discuss our threat models.

5.1 Stakeholders

It is important to identify stakeholders in our scheme in order to be able to carry out an effective security analysis. We observe that the bank is the main stakeholder in this scheme. Their interest is to maintain a satisfied clientele and protect the information and money in their care. The second group is the clients whose interests are system availability, ease and convenience of use, privacy of personal information and proper protection of their money.

5.2 Security in the Proposed Secure SMS Protocol.

This section describes the security considerations in the secure SMS protocol in conformity with the general security requirements as put forward in [10].

5.2.1 Integrity

The protocol employs a hashing algorithm to create a message digest of the message exchanged. The message digest is calculated both at the mobile phone application and at the bank server this is performed by integrity check algorithm incorporated in our application. The fields that constitute the digest are discussed in chapter 4. If the content is altered during transmission a mismatch digest will occur and the receiver will know that the message has been compromised.

5.2.2 Authentication.

For authentication purposes the protocol includes the clients account number and PIN. The PIN is selected by the user during registration for a bank account with the bank. The client enters his banking details that include the account and PIN in the mobile application and are used for authentication at the bank server side.

5.2.3 Non Repudiation

In our protocols non–repudiation is ensured through the use of the PIN, one time password and sequence number that are known only to the client and the bank. If the message is successfully decrypted by the bank using the same sequence–password pair and the PIN verified, then it indicates the corresponding client must have sent the message. The client cannot therefore deny having sent the message. Besides ensuring non repudiation the sequence number plays an important role in preventing replay attacks. We strongly assume here that the bank has concrete security policies on how it internally handles access to it's client details like PIN in its database.

According to Bond et al [34], banks have traditionally fought fraud from within and outside. In the case of insider fraud security audits and functional separation are notable methods used. However with increased complexity of banking systems banks now employ Hardware Security Modules (HSM's) that are used to protect PIN derivation keys from corrupt employees and physical attacks.

5.2.4 Confidentiality

This is achieved in both the check balance and money transfer protocols by encrypting the message using a onetime password. It is assumed that only the client and bank have knowledge of this one time password.

5.3 Threat Model.

Hypothetical Case 1;
To evaluate our model we use an example case study in which a client intends to perform a check balance or money transfer transaction. In this case the client enters his banking details into the banking application resident on his phone. This includes the PIN, account number and the encryption key. We recognize that the system can be vulnerable to both logical and physical attacks. While in transmission the message can be vulnerable to interception and also at the mobile network provider SMS centre server the message contents can easily be accessed given a copy of the message is retained at this server. In the next subsections we give assumptions used in our threat model.

5.3.1 SMS Centre Threats

In the server at the SMS centre copies of messages are held by the telecommunication company. They are not supposed to read or modify the content. We consider this a stable environment and give the assumptions below regarding this state.

- We assume the attacker has physical access to the server therefore it is easy for him or her to access the message content.
- We also assume the attacker could be a legal employee of the telecommunication company and has access to adequate computational resources to perform an attack.

5.3.2 Transmission Monitoring Threats

The mobility of clients poses a different challenge. We assume the following in this respect.

- Since messages are sent in a mobile environment an adversary has a low probability of monitoring the network during normal operation but can still if highly motivated have the opportunity of monitoring and intercepting the message as illustrated in figure 5.1.
- Unlike in the SMS centre in mobile situations the attacker has limited computational resources.

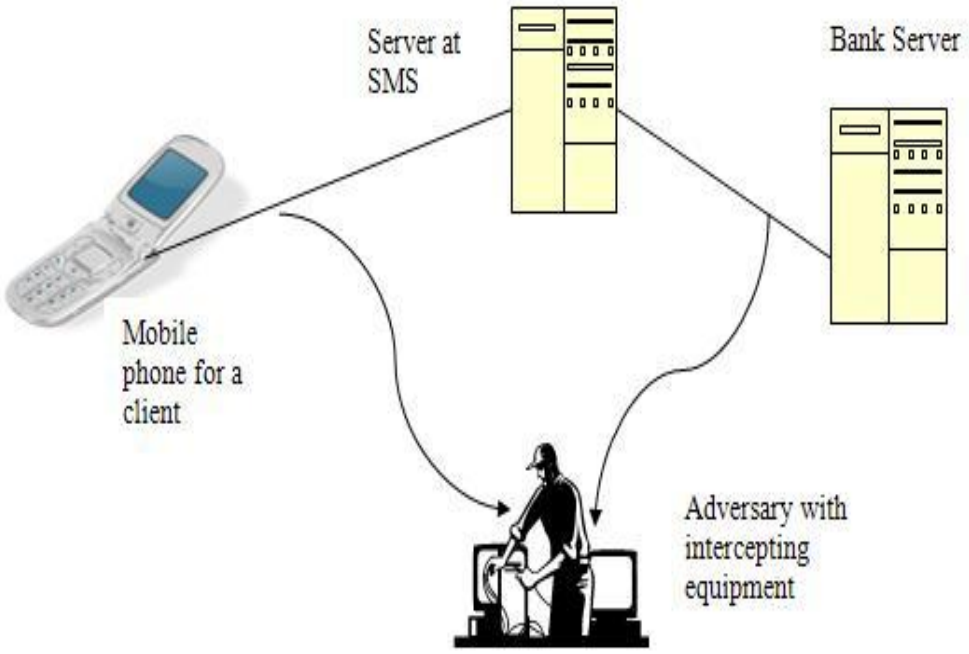

Figure 5.1: Attack Topology

5.3.3 Threat Model Discussion

We discuss the threat model of our proposed model using attack trees. This will enable us evaluate the extent to which our system is secure and to be able to identify areas for further improvement. Attack tree model is a formal method of describing security of systems [26]. Attacks against the system are presented in a tree structure form, with the goal as the root node and different ways of achieving that goal as leaf nodes.

Considering the stakeholders as listed in section (5.1) and our assumptions in sections (5.3.1) and (5.3.2) we present our security analysis attack tree model. We note that the adversarys major goals are to obtain the clients PIN number and account number. In the first case the adversary can achieve his or her goal by compromising the clients and banks physical security, intercepting the message during transmission or accessing it at the SMS centre server.

In order to be able to attack the system the adversary has to deploy surveillance equipment within the communication range of the telecommunication service provider and these devices have to remain in place for a sufficient period of time necessary to obtain enough material to perform a crypto anal-

ysis. We note that this attack is quite expensive but its imperative to take into account that the adversary will find sufficient motivation to mount an attack. Figure 5.2 below shows our attack tree model with a specific goal of obtaining the clients PIN and account number.

We present the criteria that can be applied by an adversary to achieve a given goal through representing the attack in terms of its cost μ, representing the possibility of an attack occurring by P and in case of an impossibility of occurrence by I (Figure 5.2). Based on the attackers capabilities and operational environment we derive the likelihood of occurrence of the leaf node attacks. The costs of the attacks are estimated by intuition on the time, knowledge, tools and effort an adversary invests in an attack. And also by considering the security measures established in the protocol. The costs of non leaf nodes depends on the values of the children nodes and if the node is an AND or an OR node. The cost is the sum of the children nodes if a particular node is an AND node and it is the minimum of the two children nodes if it is an OR node (Equation 5.1). The likelihood of an attack occurring is the summation of all the likelihoods of the children nodes. We formally present this as below.

Notation: Let w be a node in the tree (figure 5.2), also let $\delta(w)$ denote set of all children of (w),and let cost (w) denote cost of achieving the attack goal on node (w). We represent the costs of an attack on a node through our own evaluation by a factor μ.
Formally the cost (w) will be given by

$$cost\ (w) = \begin{cases} \mu & if\ w\ is\ a\ leaf\ node \\ \sum_{q\ \epsilon\ \delta(w)} cost(q) & if\ w\ is\ an\ AND\ node \\ Min_{q\ \epsilon\ \delta(w)} cost(q) & if\ w\ is\ an\ OR\ node. \end{cases} \tag{5.1}$$

We further define the likelihood of w occurring when w is an AND node by the following equation

$$Likelihood\ w = \begin{cases} P, & likelihood(\delta(w)) = P \\ I, & Otherwise \end{cases} \tag{5.2}$$

And also define the likelihood of w occurring when w is an OR node as follows

$$Likelihood\ w = \begin{cases} I, & likelihood(\delta(w)) = I \\ P, & Otherwise \end{cases} \tag{5.3}$$

Our security analysis in the attack tree model of (figure 5.2) shows our proposed system is quite strong save for the vulnerability through the use of a password and sequence list. It is therefore important that the password-sequence list be exchanged in a very secure way.

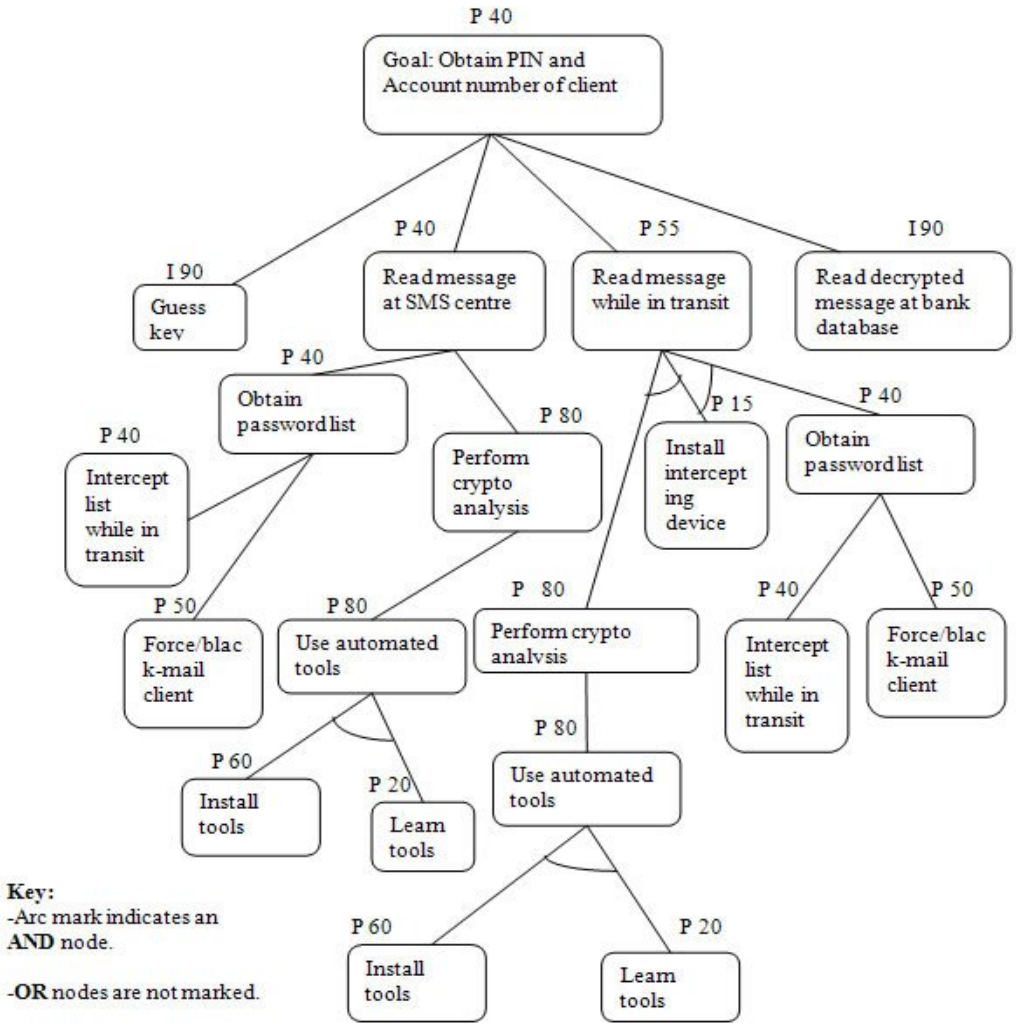

Figure 5.2: Attack Tree Threat Model

Chapter 6

The Prototype

In this chapter we present the implementation of our prototype. We start by describing the system development specifications and the development environment Then discuss the design, package implementation and testing.

6.1 System Development

The solution involved developing a MIDlet suit using the $Java^{TM}$ technology. The technology is a suitable development environment because presently many mobile phones in the market come with a standard built in Java Virtual Machine (JVM). The technology also has many libraries to assist in the mobile application development. The Java platform used is the Micro Edition (J2ME). It is specifically designed for consumer devices with limited memory, display capabilities and resources [8]. Within this platform there are two configurations that is the Connected Device Configuration (CDC) and the Connected Limited device configuration (CLDC). The implementation uses the (CLDC) configuration because the defined characteristics of the cellular handset fall under this category. The CLDC defines a specification for a JVM and a set of java classes (libraries). The minimum software and hardware requirements for CLDC are [8];

- 128 kilobytes of memory for running the JVM and CLDC libraries.
- 32 kilobytes of volatile memory for runtime memory allocation.
- The host Operating system capable of launching and selecting applications.
- And it should also have the ability to remove java applications from the device.

The majority of mobile handsets do have this minimum requirement which makes CDLC configuration a suitable tool for the development of the suggested solution.

6.1.1 Mobile Information Device Application, MIDlet

MIDlets are java programs loaded onto a mobile terminal for example a mobile phone. It consists of a JAD and a JAR file. We describe these two parts in the following subsections.

6.1.1.1 JAD Files

Java Application Descriptor (JAD) is a single text file containing information about the MIDlet. The information stored in this file depends on the version of Mobile Information Data Program (MIDP) in our project its version 2.0. It comprises the path where the JAR file can be downloaded [8].

6.1.1.2 JAR Files

The JAR files (Java Archive Files) hold the Manifest file that contains some of the attributes of the JAD file and also contains all classes of the MIDlet and the resources it needs. The reason for storing the same information in the JAD and JAR files is for verification of the vender of the MIDlet application [8].

6.2 Development Environment.

The development environment used was the netbeans Integrated Development Environment (IDE) that is written entirely in java using the netbeans platform. Netbeans provides all the tools a software developer requires to create cross platform java desktop enterprise and web applications. It is an open source program and runs on various operating systems i.e. Windows, Linux as well as Solaris.The version of netbeans used in this project was version 5.5 it builds upon the previous version 5.0 which introduced comprehensive support for developing IDE modules and rich client applications. All the applications in our project prototype implementation were developed under this environment. The client User Interface MIDlet application was developed using the netbeans mobility pack with an integrated mobile toolkit emulator. The server and password generator were developed as J2SE applications in netbeans. The database used in the project is the MySQL database and the connection is realised through the Java Database Connectivity (JDBC) using the MySQL $Java^{TM}$ library.

6.3 System Design

The major reason for design is to create a layout of how classes are connected. This was achieved using the Universal Modelling Language (UML) module

integrated in the netbeans IDE. Also central to our design was the importance of ensuring that the graphical user interface and underlying protocols were as separate as possible. The reason being building the ability to cater for future demands of additional functionality for example adding new transaction types like vending mobile phone air time. Further we desired to keep the finished program as small as possible because of the limited amount of memory in mobile phones. To achieve this, a program called an Obfuscater is used. This program is integrated within the netbeans IDE. When executed on a target program it shortens the names of all the variables, classes and methods hence saving space in the jar file. In our design we also assume that the banking application is pre installed on the clients mobile phone.

6.3.1 Use Case Illustration.

In figure 6.1 we illustrate the use case diagram for design purposes of our project prototype. In the diagram the client starts the application on his or her mobile phone in order to perform a remote banking transaction. The choices applicable are check balance and money transfer. The client after selecting the transaction type proceeds and enters her banking details in the application interface and a secure message is created and sent by TCP socket connection to the server. The server decrypts the message, verifies message integrity and carries out account authentication. When this is successfully achieved the server thereafter performs the requested banking transaction. A confirmation is thereafter sent to the client indicating success of the requested transaction.

6.3.2 Class Collaboration of Mobile User Application Package

Figure 6.2 illustrates the class collaboration diagram for the mobile user MIDlet application package. The application interface class is responsible for the interaction between the client and banking application. It has methods to start the application, establish connection, create message, perform base64 encoding/decoding, send the message and receive the message. It also has methods for encrypting and message integrity checks.

The padding class has the pad method used to extend the input string to a desired length by using either spaces or zeros. The pad front method appends random characters at the front of an input string to attain a desired length.

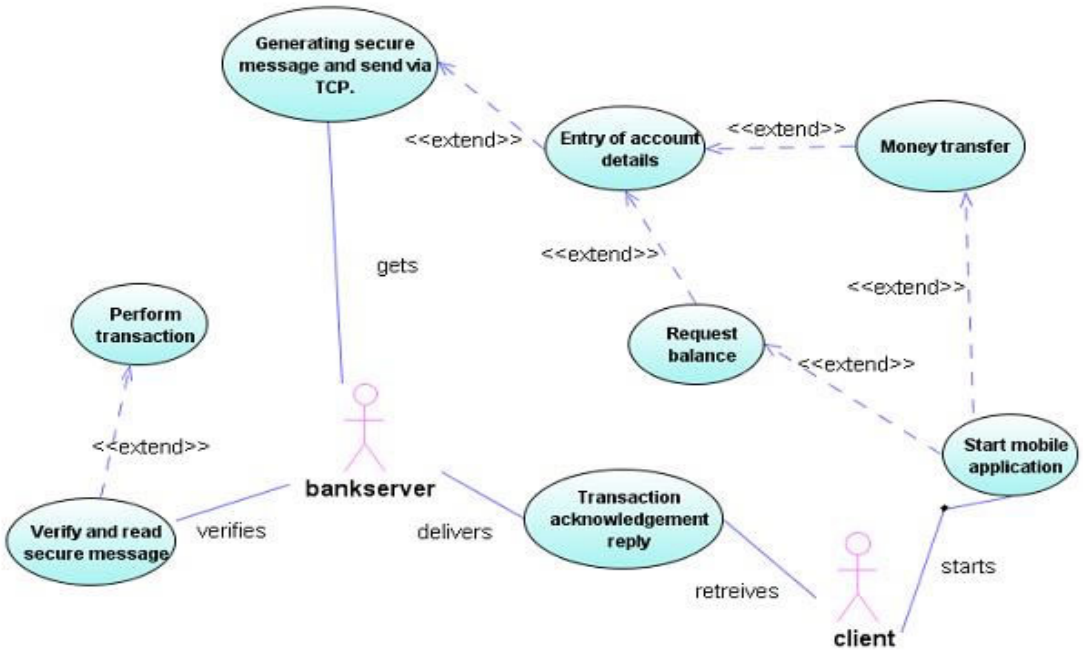

Figure 6.1: Use Case Diagram of Prototype

The security class is responsible for ciphering and integrity check of the message. It has methods for decoding, encoding and processing the digest of the message.

6.3.3 SMS Server Package

In figure 6.3 we illustrate the class collaboration in the server package. The echo class starts the server, performs a base64 decoding and establishes connection to the client. The reply class has methods to process the reply and send it back to the client. The message handler class is responsible for breaking the message into unprocessed message fields this include version number, account number, sequence number, cipher text and message digest. The class has various methods to retrieve the requested message fields.

The database connect class has methods to connect to the MySQL database. The banking detail class has methods to retrieve message fields used by the bank to authenticate the client and perform the transaction. It processes the decoded message to retrieve these message fields. Finally the security and padding classes have the same functionality as in the MIDlet application package explained in section 6.3.2.

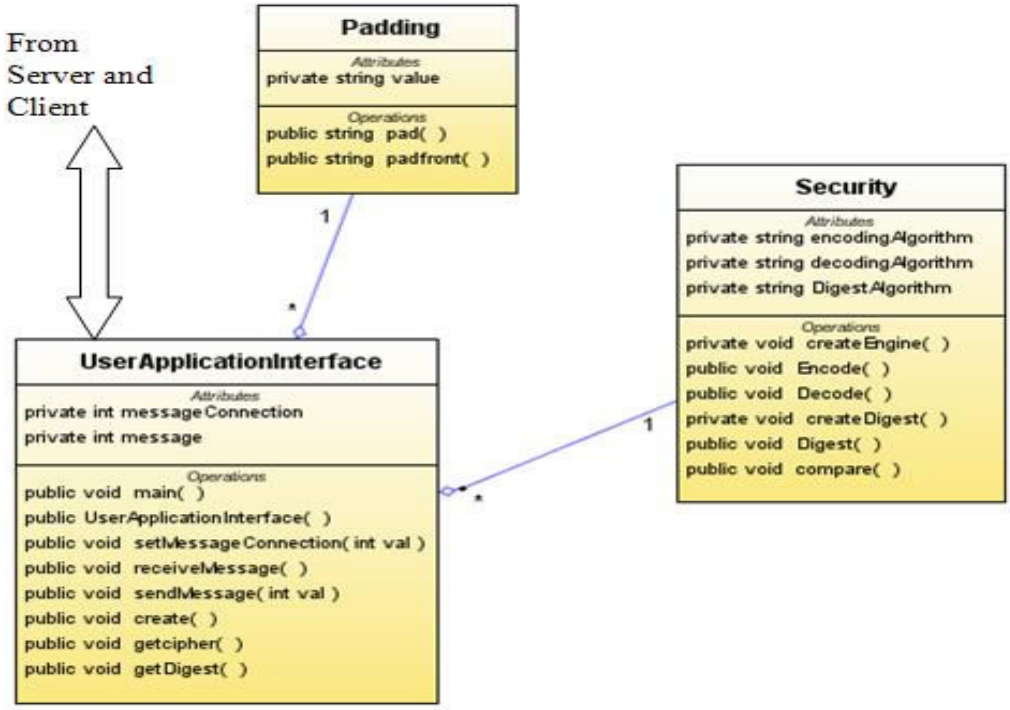

Figure 6.2: Class Diagram of Mobile User Application Package

6.4 Prototype Implementation

In this subsection we describe the implementation of the various packages as illustrated in the class diagrams. Our major focus was on the MIDlet and to be able to test we had to develop a server side application and build a connection using TCP socket. However in an ideal situation the connection should be through an SMS gateway we believe though that our prototype is a close representation of an ideal system.

6.4.1 Security Technologies Used

To be able to attain encryption, decryption and integrity check offered in the security class both at the MIDlet and the server side the bouncy castle cryptographic package was used. Bouncy castle is a collection of application programming interface used in cryptography. For encryption and decryption we chose the light weight Advanced Encryption Standard (AES) using the chain block cipher mode implemented by bouncy castle. We chose this algorithm because of its simple nature making it suitable for cellular phones

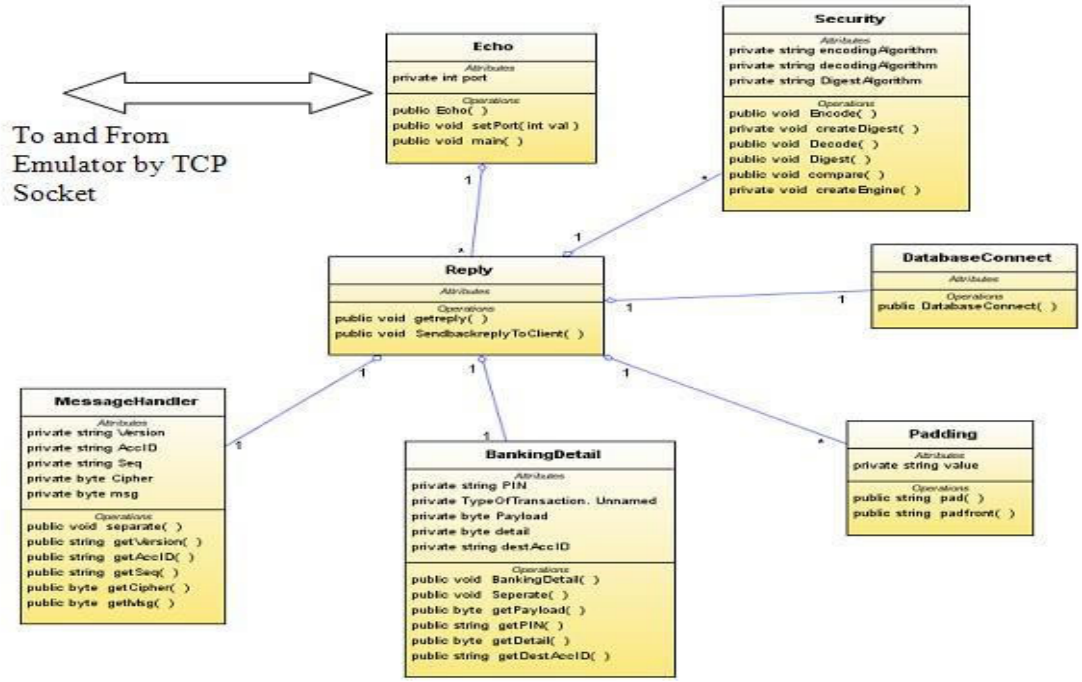

Figure 6.3: Class Diagram of Server Package

given their constraint of memory capacity. The advanced encryption standard is a symmetric algorithm utilising block encryption of 128 bits in size and supporting key sizes of 128, 192 and 256 bits as a minimum [10,21].

The digest algorithm used is the Secure Hash Algorithm (SHA–1) developed by the National Institute of Standard and Technology (NIST)[14]. The algorithm takes as input a message with a maximum length of less than 264 bits and produces an output of 160 bits message. SHA–1 was chosen because there is no weakness so far that has been identified and published as compared to Message-Digest algorithm (MD5) that has been suggested to have a vulnerability to cryptanalysis [10].

The java code snippet below is taken from the security class of our implementation. It shows the method for encryption and calculation of the message digest corresponding to our protocol in section 4.3.4. In the encryption method the message content is taken as a byte array and the password as a string. A cipher algorithm is selected and cipher engine is created in our case its the lightweight AES. The digestAlg parameter enables the caller select the message digest algorithm in our case we used the SHA–1 algorithm. The digest engine object creates a SHA–1 message digest object. The engine

is used to retrieve the message digest.

```
public static byte[ ]
encodeMessage (byte[ ] plainText, String password)
throws Exception
{
byte content[ ] = plainText;
byte key[ ] = password.getBytes( );
// Create the cipher Engine.
BufferedBlockCiphercipherEngine =
new PaddedBufferedBlockCipher(createEngine(cipherAlg));
// Initialize the cipher for encryption
cipherEngine.init(true, new KeyParameter(key));
byte[ ] cipherText = new byte[cipherEngine.getOutputSize(content.length)];
byte [ ] digest = null;
// Do encryption
int cipherTextLength = cipherEngine.processBytes(content, 0, content.length,
cipherText, 0);
cipherEngine.doFinal(cipherText, cipherTextLength);
{
//Calculate digest
Digest digestEngine = createDigest(digestAlg);
int digestSize = digestEngine.getDigestSize();
digest = new byte[digestSize];
digestEngine.update(content, 0, content.length);
digestEngine.doFinal(digest, 0);
}
// Create temporary streams
ByteArrayOutputStream out = new ByteArrayOutputStream();
DataOutputStream dout = new DataOutputStream(out);
// write length
dout.writeShort(cipherText.length);
// write cipher text
out.write(cipherText);
// write digest
out.write(digest);
return out.toByteArray();
}
```

6.4.1.1 Sequence–Password Generator

In our prototype we used a password generator to write passwords and sequence numbers in to the database. It also outputs a copy as an html page as an option. The application displays client account identification numbers

existing in the database and one can be selected for purposes of generating or retrieving the sequence number and passwords for a particular client. The generated passwords are formed by random characters which the clients can easily enter into their mobile phones. The algorithm used for password generation in this prototype is SHA1 pseudo random number generator. It generates non predictable random numbers used for password generation.

6.4.2 Communication between Client and Server

In our prototype implementation we used Transport Control Protocol (TCP) socket connection. TCP enables exchange of data using stream sockets it provides connections that need to be established before data is sent and uses the notion of port numbers to identify sending and receiving application end points on a host. It has the following three phases

- Connection establishment
- Data transfer
- Connection termination.

The major problem experienced in our implementation was loss of packet hence effecting the decryption of the transmitted message. To solve this problem we used base64 encoding. Base64 is an encoding that allows to encode a sequence of arbitrary bytes as a sequence of printable ASCII characters [35].

6.4.3 Testing

The major drawback to the ideal testing of the MIDlet by loading the application on a mobile phone and testing with SMS gateway was lack of supporting resources. In an attempt to test the client a server was written that was directly connected to MYSQL database. The client was simulated using mobile toolkit emulator and connected to the server using TCP socket connection as explained in section 6.4.3. However since the client is meant as prototype we consider this form of testing sufficient. The client successfully communicated with the server and retrieved the desired requested information.

Figure 6.4 illustrates the screenshots run with the wireless toolkit emulator taken from an example test of our prototype implementation. The first screen displays the application to be launched. The user launches the application enters the destination server number in the next display and selects type of transaction in the subsequent one in this case we selected check balance transaction. The client is then presented with a screen to enter banking

details and a menu to send the requested transaction. The last screen displays the received reply from the server.

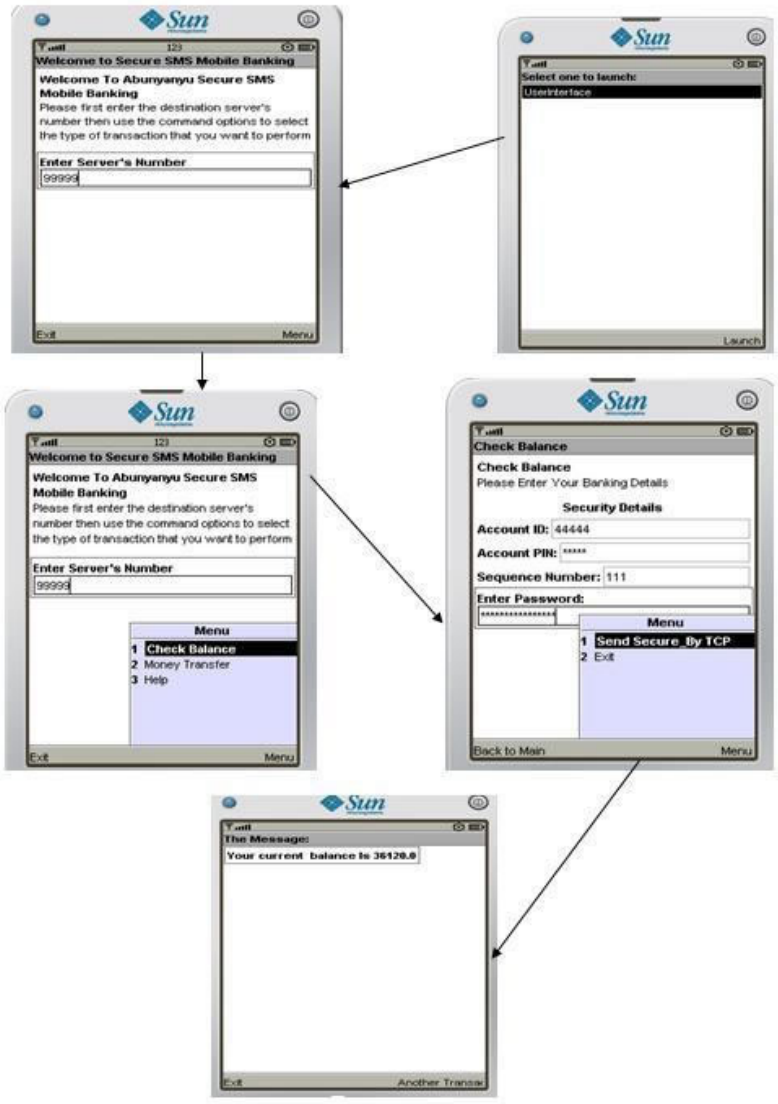

Figure 6.4: Screen Shot Depicting Check Balance Test Transaction

Chapter 7

Conclusion and Reflections.

In this chapter we give our reflections on the application of mobile banking services in developing countries especially in Africa. We present the challenges and opportunities of such systems and end the chapter with a conclusion of this work and highlighting future areas for further exploration.

7.1 Reflections

The potential for SMS mobile banking services is particularly high in countries were internet infrastructure hinders the access to electronic banking services. In most developing countries particularly in Africa internet connectivity and bandwidth are low and the population is not urbanised and averagely poor hence making realization of internet banking services not viable in most parts of the continent. We therefore note that for the foreseeable future given the high diffusion rate of mobile telephony in developing countries SMS banking is the most viable option for offering affordable remote banking transactions.

The success of SMS banking in developing countries will mainly depend on the banking industry on how they perceive the usefulness of the system to enable them offer better services to their clients and help reduce in their operating costs. The potential of the general adoption of SMS banking is there given the fact that clients give a lot of attention to convenience and accessibility of personal accounts when choosing a banking service provider

7.2 Conclusions

In this thesis we have reviewed the current state of SMS banking services in developing countries taking Uganda as our domain of study and highlighted the technologies and security shortfalls of enabling systems utilising ordinary plaintext SMS message communication. From the study of these

systems and related ones we have come up with a protocol and implemented a prototype system. Based on the current trend in mobile banking we are quite optimistic that internet banking will remain the most attractive service for developed countries whereas SMS banking will gain more in-roads in developing countries.

In chapter five we presented a security analysis of our proposed protocol. We note that our protocol is quite resilient to cryptanalysis attacks and offers optimal communication costs given that a single message is used by the client per transaction. However given the limited memory capacity of the current low end mobile phones predominantly being used in our project domain, the exchange of sequence-password list poses a security loophole that can easily jeopardise the system security. Thus there is need for further research to solve this problem probably through use of optimised asymmetric cryptosystems suitable for low memory capacity mobile terminals.

In our design we have assumed that the mobile banking application be pre–installed on the clients phone, in reality this may not be cost effective since the mobile phone manufacturers may have to be contracted and besides its a cross platform application intended for various mobile phone operating systems. It is therefore important that ways of distributing this application without breach of security be explored. Another area for future work could be how to integrate anonymity in our prototype given that it can be extended to offer payments between customers. In summary we are quite optimistic our system can gain acceptance in society given the security benefits it offers

Reference

[1] http://conventions.coe.int/Treaty/EN/Treaties/Html/185.htm

[2] Tommi Laukkanen (2005), Comparing Consumer Value Creation in Internet and Mobile Banking Proceedings of the International Conference on Mobile Business pg 655-658.

[3] http://www.ucc.co.ug/marketInfo/marketstatistics.php

[4] Key Pousttchi and Martin Schurig (2004), Assessment of Todays Mobile Banking Applications from the View of Customer Requirements, Proceedings of the 37th Hawaii International Conference on System Sciences, pp 1-10.

[5] Niina Mallat, Matti Rossi, and Virpi Kristiina Tuunainen (2004), Mobile Banking Services Communications of the ACM, 47, 5 pp 42-46

[6] http://www.c-sam.com/

[7] GSM technical specification (1997), Digital cellular telecommunications system (Phase 2+); General description of a GSM Public Land Mobile Network (PLMN), http://www.etsi.org.

[8] John W. Muchow (2002), Core J2ME Technology & MIDP, The Sun Microsystems Press-Java Series, California USA.

[9]Jonathan Knudsen (2002), MIDP Application Security : design Concerns and Cryptography, Sun Developer network ;
http://developers.sun.com/techtopics

[10] William Stallings (2003), Network Security Essentials, Pearson Education, Inc. Upper Saddle river, New Jersey USA.

[11]Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone (2001), Handbook of Applied Cryptography, CRC press, USA.

[12] Lam, K.Y and Golmann, D. (1992), Freshness assurance of authentication protocols. Proceedings of the second European Symposium on Research in Computer Science, Toulouse, France, LNCS 648.

[13] Lam, K.Y, Cung, S., Gu, M., and Sun, J. (2003). Lightweight Security for Mobile Commerce Transactions. Computer Communications 26, 2052 -2060.

[14]Jonathan Knudsen (2002), MIDP Application Security : design Concerns and Cryptography, Sun Developer network ;
http://developers.sun.com/techtopics

[15] Banji Oyelaran-Oyeyinka and Catherine Nyaki Adeya (2004),Internet Access in Africa: Empirical Evidence from Kenya and Nigeria Telematics and Informatics, 21, 1 , pp 67-81

[16] [16] Charles J. Kenny (2000), Expanding Internet Access to the Rural poor in Africa. The African Internet and Telecom Summit Banjul, Gambia;
http://www.itu.int/africainternet2000/Documents/doc7$_e$.*htm*

[17] Tasneem G. Brutch and Paul C. Brutch(1998), Mutual Authentication, Confidentiality, and Key Management (MACKMAN) System for Mobile Computing and Wireless Communication, IEEE Annual Computer Security application conference.

[18] Michael Minges, Walyer Brown and tim Kelly (2001), The Internet in an African LDC: Uganda Case Study;
http://www.itu.int/ITU-D/ict/cs/uganda/material/uganda.pdf

[19] G.J. Simmons (1979), Symmetric and asymmetric encryption. ACM Computing Surveys, 11(4):305330

[20] W.Diffie and M.E. Hellman (1979). Privacy and authentication: An introduction to cryptography.
Proceedings of IEEE, 67(3):397427.

[21] National Bureau of Standards (1977) U.S. Department of Commerce, Washington, D.C. Data Encryption Standard FIPS Pub 46.

[22]R.L. Rivest, A. Shamir, and L. Adleman (1978). A method for obtaining digital signatures and public-key cryptosystems Communications of the ACM, 21(2):120126.

[23] Simon S. Lam and Thomas Y.C. Woo (1997), Authentication for Dis-

tributed Systems, Internet Besieged: Countering Cyberspace Scofflaws ACM.

[24] Kwok-Yan Lam and Dieter Gollman (1992) Freshness Assurance of Authentication Protocols Proceedings of the Second European Symposium on Research in Computer Security Lecture Notes In Computer Science; Springer-Verlag Vol. 648 261 - 272

[25] Beliovin, S.M., Merrltt, M(1990).: Limitations of the Kerberos Authentication System. ACM Computer Communications Review 20(5) 119-132

[26] Bruce Schneier (1999), Attack Trees: Modelling security threats, Dr. Dobb's Journal http://www.schneier.com/paper-attacktrees-ddj-ft.html.

[27] Klaus Vedder (1998), GSM: Security, Services, and the SIM, Springer-Verlag Berlin Heidelberg, LNCS 1528, pp. 224-240.

[28] Donar Porter and Lawrence D Wiess (1998), Interactive Voice Response System for Banking by Telephone, United States Patent, Patent Number 5,825,856.

[29] WAP Forum, Wireless Application Protocol Architecture Specification, Version 12-July-2001, Available from http://www.openmobilealliance.org

[30] Web Pro Forums (2007), Wireless Short Message Service (SMS) available at International Engineering Consortium: http://www.iec.org/

[31] Steve Lord (2003), Trouble at the Telco: When GSM Goes Bad Network Security,2003(1): 10-12.

[32] Biryukov, Shamir, Wagner - Real Time: Cryptanalysis of A5/1 on a PC Department of Computer Science, The Weizmann Institute, Rohovot Isreal.

[33] Elad Barkan, Eli Biham1 and Nathan Keller(2003),Instant Ciphertext-Only Cryptanalysis of GSM Encrypted Communication, CRYPTO 2003, LNCS 2729, pp. 600616.

[34] Mike Bond, Piotr Zielinski (2003), Decimalization table attacks for PIN Cracking, University of Cambridge http://www.cl.cam.ac.uk/

[35]N. Freed, Innosoft and N. Borenstein (1996) Network Working Group, Request for Comments: 2045. http://www.ietf.org/rfc/rfc2045.txt