UNIVERSITY OF NAIROBI

AN IMPLEMENTATION OF ADVANCED ENCRYPTION STANDARD IN MOBILE COMMUNICATION: SECURE MESAGING APPLICATION

NAME: VERONICAH MUTUA

REG/NO: P56/8806/2005

SUPERVISOR: ANDREW MWAURA

Project submitted for the partial fulfillment of Masters of Science in Computer Science

DECLARATION

This research project, as presented on this report is my original work and to the best of my knowledge has not been presented for any other university award.

Veronicah Mutua

Signed: …………………………

Date :…………………………

This project has been submitted as part of fulfillment of the requirements for the award of Masters of Science in Computer Science of the School of Computing and Informatics of the University of Nairobi ,with my approval as the university supervisor

Mr. Andrew Mwaura

Signed:…………………………………………………..

Date:……………………………………………………

ACKNOWLDGEMENT

This project would not be successful without the immense support of my supervisor, Mr. Andrew Mwaura, who was very instrumental and supportive throughout the project. Secondly, my sincere gratitude goes to the Project Presentation Panelists (Prof. Waema, Prof. waiganjo and Dr. Orwa) who were keen to point out the major issues in my project and patient with me while I presented my project to them.

Special gratitude goes to my Children Luther, Laura and Cyril who understood that I had to finish this project and provided the necessary co-operation, patience and supportive atmosphere.

Most importantly I wish to state here that without the Support and encouragement from my Dearest and Loving Husband, Roger Juma, I would never have completed this project.

Thanks.

ABSTRACT

The emergence of Short Messaging services (SMS) has extensively transformed the nature of communication and information sharing. Billions of text messages are sent daily in the world sent in plain text format and hence user privacy and security is not assured. With the increasing number of software crackers available for free in the internet, SMS continues to suffer from security vulnerabilities and loopholes. Information security has long been thought to be inclusive of only personal computers and networks. However, with the technological trend shifting from computers to mobile devices, malicious attackers are now targeting mobile devices and their users

The aim of this project was to develop a secure messaging platform for Java enabled phones to reduce these vulnerabilities and loopholes. To implement the Advanced Encryption Standard (AES), the application was developed in MIDP 2.0 and CLDC 1.1configurations and applied symmetric encryption concepts. The experimental results revealed that the system was able to encrypt, decrypt, send and receive text messages without adding changing the size of the packets.

The decision to use AES was based on the status of AES as the currently accepted standard for data encryption, and its nearly ubiquitous use in encryption-offering software (and hardware). It is a thoroughly analyzed and accepted algorithm, offering powerful encryption with a small key size. More so, Java Micro-edition for small devices (J2ME) comes with advanced encryption standard classes that can be applied during the development of mobile applications therefore AES is easier to implement in Java platforms as compared to DES which is much slower-depending on the size of the key.

Key words: SMS, encrypt, decrypt, SMSC, malware, GSM, mobile applications, AES.

TABLE OF CONTENTS

LIST OF TABLES AND FIGURES

LIST OF TABLES

DEFINATIONS

GSM- an internationally accepted cell phone network specification the European Telecommunications

SMSC - A short message service center (**SMSC**) is a network element in the mobile telephone network which delivers SMS messages.

MSC- The mobile switching centre server is a soft-switch variant of the mobile switching centre, which provides circuit-switched calling, mobility management, and GSM services to the mobile phones roaming within the area that it serves.

HLR- The home location register (HLR) is a central database that contains details of each mobile phone subscriber that is authorized to use the GSM core network.

Plaintext - a sequence of characters (either letters from the alphabet, numbers or bytes of data) that is in a form at which no effort has been made to render the information unreadable andthus, that can be easily read from and understood. [wiki]

Ciphertext ( ) - a sequence of characters (either letters from the alphabet, numbers or bytes of data) that is encrypted using an encryption algorithm. Plaintext cannot be deduced from properly encrypted ciphertext.

Encryption - the process of turning plaintext into ciphertext (encoding).

Decryption - the process of turning ciphertext into plaintext (decoding).

Key ( ) - a piece of information (or parameter) that controls the execution of a cryptography algorithm. A key can be used either for encryption (obtaining the ciphertext out of the plaintext), or decryption (obtaining the plaintext out of the ciphertext).

CHAPTER ONE: INTRODUCTION

1.0 Introduction

There has been a tremendous growth in the number of mobile phone users across the world. Not only have new business models (such as mobile commerce) emerged but also many lives have been transformed by this technology. However, mobile technology has brought about many challenges in relation to information security. The wireless networks and mobile ad hoc networks have become vulnerable to various malicious attacks especially with the existence of tools such as 'juju'- a software tool that can sniff record and spy on text messages and voice. Billions of short messages are sent daily across the world in an insecure channel that can easily be sniffed. With the current need for privacy of information, it is expedient for us to come up with a solution (a mobile application) that will ensure confidentiality, integrity and availability of short messages. This project is aimed at developing a secure mobile messaging application (in Java) that will allow users to encrypt and decrypt their messages for secure transmission. The researcher will aim at creating a new messaging model that could be adopted and integrated with the current mobile banking services. The application is to be built using the advanced encryption standard (AES)concepts and will mainly be tested using java enabled phone.

1.1 Background information

Mobile phones have become an integral part of the modern world, providing human connectivity in a way never before possible (Jeff B., Bill S., and Vetter .R, 2007). A recent United Nations report   estimated that the total number of mobile phone subscribers in the world now exceeds 2.68 billion. It is estimated that around 80 percent of the world's population has mobile phone coverage, with 90 percent coverage forecast by 2015(Njenga, 2009).

The ever growing number of mobile phone users has provided a wide platform for both corporate organizations and government institutions to provide services to their clients.

Mobile phones are very handy devices and are widely used by people around us for day-to-day functionalities, (Sreenarayan, 2011). People are becoming more and more dependent on mobile phones for performing critical functionalities like bank transactions, etc. Subsequently, when people depend more on phones, for faster processing, a lot of sensitive data are stored in the phone and a considerable amount is also transmitted to the server.

This has in-turn led to technological shift from the traditional personal computers (PC) to mobile handsets. Nay, new business models have emerged with mobile banking taking a center stage in countries such as Kenya where MPESA money transfer system has shaped banking(Safaricom 2009) .

While most cell phones are used for their original intent—making telephone calls wirelessly—these devices are also loaded with other features that are often little used or even ignored.

According to Jeff (Jeff *et al* 2007) ‘One feature that users have begun to fully exploit in recent years is the *short message service* or text messaging. This basic service allows the exchange of short text messages between subscribers’.

With billions of text messages being sent, malicious hackers and crackers have taken notice of this new field and they are busy developing software tools that can intercept, replay and sniff short messages on transit(CERT, 2012).

Based on these numbers, it is evident that SMS messaging is becoming a widely used communication mechanism for cell phone users .The short messages are sent in plain text and binary format and can easily be intercepted and replayed with existing cracking tools. This loophole provides great opportunities for hackers to take advantage of . A good example is the ‘phone Snoop’ mobile application that is able to intercept text messages and record conversations without the users knowledge.

It is for these reasons that the researcher proposes to undertake an in-depth study of how the SMS works system and develop and secure messaging application for java enables phones.

1.2 Problem statement

The existing messaging platform allows texts to be sent in plain text hence the SMS can easily be intercepted and replayed using the existing software crackers. In this age where privacy is highly valued, its expedient for us to come up with a secure messaging application that will ensure a secure messaging system for short messaging services.

1.3 Objectives

The main objective of the project was to come up with a secure mobile messaging application for java enabled mobile handsets and devices. This was achieved by:

1. Developing a secure messaging mobile application in J2ME(Java Micro-Edition for Mobile devices)
2. Testing the application with the existing mobile banking application

1.4 Research questions

1. What architecture is used in SMS and how does it work?
2. What is the appropriate encryption criteria that can be used in Short Messages Services?
3. What security needs exist in the domain of users using SMS?

1.4.1 Justification

With the current number of mobile users increasing and the corresponding increase in numbers of messages sent daily, the proposed application will come in handy for both the users and the researchers .To begin with ,the proposed mobile app will provide a secure messaging channel that will enhance the privacy of the users .with features such as encrypting the 'drafts', users will not be worried about anyone viewing their inbox or draft messages because the content of the messages will be encrypted. Only the users with the right keys will be able to access the messages. Secondly, the messages remain encrypted during the transit. When it goes to the wrong recipient, the recipients will not be able to read or decrypt and hence ensuring confidentiality and integrity of the SMS. For the future researchers, Applying the same concepts from the proposed application and integrating it to the current mobile banking will form an integral part of the future research work.

1.4.2 Expected output

At the end of the project, a fully functioning secure messaging mobile application is expected to be developed and tested. The application will be tested using different types of phones from various manufactures to ensure that it meets its requirements.

1.5 Advantages of the system

The intended application is envisaged to provide the following advantages to the users:

1. Ensure privacy of information sent via the public GSM network.

2. Improve efficiency of private communications by ensuring integrity of the SMS sent.

3. The application can be modified and customized for use in security agencies such as the national security.

4. Target millions of users without smart phones and who still would want to achieve confidentiality and integrity of

1.6 Disadvantages

1. The first version of the system targets only java phones but the other versions are bound to run on other operating systems such as Android and Windows mobile OS.

2. The system could be used maliciously by users to pass messages with malicious intents especially during elections

3. The concept of Key distribution still remains a challenge and hence a bottleneck of the system.

1.7 Scope of the project

The secure messaging application version 1.0 only runs on feature phones running Java MIDP 2.0 and CLDC 1.1. The application does not run in Android 2.3.4, Symbian or BlackBerry. However, the researcher intends to implement the same application in the mentioned operating systems.

1.8 Assumptions and limitations

The proposed system will run on both the sender’s phone and the recipient’s phone. The users will have secret keys which is unique and only known to the phone numbers .In case of any breaches, the application will provide users with the ability to change their secret keys. The same key is used to encrypt and decrypt the message. Key distribution will be the main limitation for the proposed mobile application.

CHAPTER TWO: LITERATURE REVIEW

2.1 Introduction

The emergence of mobile technology services in the recent years extensively transformed mobile communication and information sharing. Among the most popular services is the Short Message services (SMS) in which billions of SMS's are sent daily across the world (Baron, Patterson, and Harris, 2006). Short messages service has not only been used for individual conversations but in corporate(mobile banking), social and the political world Stuart (2003) .On the other hand, Studies have shown that the SMS channel is vulnerable for the man in the middle attacks and other hacking attacks. With billions of text messages sent in plain text, the integrity of the text messages and the privacy of the users is bound to be breached especially now with the availability of many software crackers available for free in the internet(Omwansa,2009). Several studies have been carried out with the aim of improving the SMS platform. At the same time, several studies have also shown the need for using encryption in the SMS platform. This chapter elaborates on these studies with the aim of clearly understanding the vulnerabilities and threats that exist in SMS services. The chapter discusses also elaborates on various aspects of SMS architecture and the advanced encryption method.

2.2Trend in mobile technology

Mobile phones are very handy devices and are widely used by people around us for day-to-day functionalities, (Sreenarayan, 2011). Mobile phones have become an integral part of the modern world, providing human connectivity in a way never before possible (Jeff B., Bill S., and Vetter .R, 2007). A recent United Nations report estimated that the total number of mobile phone subscribers in the world now exceeds 2.68 billion. It is estimated that around 80% of the world's population has mobile phone coverage, with 90% coverage forecast by 2015(Njenga, 2009).Other Studies have also shown that 'people are becoming more and more dependent on mobile phones for performing critical functionalities like bank transactions (Omwansa T., 2009). Needless to say, when people depend more on phones, for faster processing, a lot of sensitive data are stored in the phone and a considerable amount is also transmitted to the server. Other studies informs us that the technological trend is drastically moving from large personal computers to digitized mobile handsets(Nysveen, H., Pedersen, P.E., & Thorbjornsen, H. 2005). With the emergence of mobile technology, various business models and security architectures have been developed to ensure secure communication as in the case of mobile banking(Herzberg, 2003).

> Newer versions of security protocols have been developed to make the system resilient to attacks such as fraud. Some of the technology that have been deployed in this channel include; the WAP (Wireless Application Protocol) over GPRS (General Packet Radio Service) and SMS (Short Message Service) using the WIG (Wireless Internet Gateway).
>
> (Herzberg, 2003)

The ever growing number of mobile phone users has provided a wide platform for both corporate organizations and government institutions to provide services to their clients (Njenga, 2010). The report presented by GSM demonstrated that 'One feature that users have begun to fully exploit in recent years is the *short message service* or text messaging' (GSM, 2007).

2.2.1 Mobile malware and vulnerabilities

Information security has long been thought to be inclusive of only personal computers and networks. However, with the technological trend shifting from computers to mobile devices, malicious attackers are now targeting mobile devices and their users as observed by Andrew, (Andrew, 2011). Studies like the one carried out by PC world, show that smart phones and to a larger extent mobile devices are more vulnerable to attacks because consumers of these devices are oblivious of the eminent threats that faces them.

According to Mocana Group, 'Mobile devices have become critical business, military and industrial production tools, carrying valuable data well worth destroying, corrupting and, most importantly, stealing' (Mocana, 2009). It is incontrovertible that mobile phone have become very instrumental in our day to day lives. Most of our personal information is stored in the phone memory and messages. We, for example, have technically moved from traditional banking to mobile banking and a lot of our personal information is stored up on our mobile devices. It is for these reasons that various mobile malware have emerged to ruin us. Studies have shown that most martial lawsuits that occur in the United States are as a result of phone spying using these malware (Macfee, 2012)**.** Reports such as State of Mobile Security report have also demonstrated that that SMS fraud has steadily grown since July 2011(PC-world,2012). SMS hacking tool such as juju are known to have the abilities to spy on text messages, intercept send messages and replay the same short message to several recipients without the knowledge of the owner of the phone.. One of the most common Trojan to attack Java enabled mobile devices is the *Trojan.Redbrowser*. This application sends premium-rate SMS message and ‘attempts to trick users into believing it is a legitimate application that allows users to visit WAP sites without using a WAP connection’(Mocana, 2009).

> The volume of malware designed for mobile devices is a direct response to the speed at which the technology is being adopted, according to Eset's report. "If the market grows and technology is enhanced, then as long as users who use these devices to store an increasing amount of sensitive information do not adopt the necessary measures, it is logical to expect cybercriminals to create computer threats to profit from this situation"(ESET,2012).

However, it is expedient to note that mobile malware is still evolving as compared to PC malware. ‘As the business of mobile malware evolves, a significant number of criminals have settled on apps that secretly bill victims for premium text services, a new study shows’ (PC-world, 2012).

> Security and anti-malware firm Trend Micro indicated in its third quarter 2012 report that mobile malware on the Android OS had swelled approximately sixfold from April to September, when the number of attacks rose from 11,000 to more than 175,000. These include spambots and spyware; tollware that surreptitiously send text messages to services that charge a fee; and apps that secretly record phone calls and intercept texts used to authenticate financial transactions.

Although a lot of research is still being carried on to improve on transmission of the text messages, for now, text messages are sent in plain texts and even an amateur can retrieve and intercept them. As demonstrated by Wang, "Many organizations don't have even basic security such as encryption and DLP [data loss prevention] in place. They also don't spend adequate time educating employees about risks." There is therefore a need for secure encryption applications for mobile SMS services.

Wireless transactions, for example mobile banking which falls under cellular communication is very important in today's business and will not enjoy the maximum benefits unless it is taken a step further in terms of security (Chepken K., 2004)

2.2.2 Why J2ME application

Studies have shown that most mobile phone users in Africa and Kenya in Particular own cheap java enabled phones (Njenga, 2006). Most of the users cannot afford smartphones that are highly available in the market. However the trend shows that 16% of the mobile phone users are shifting to the Smartphone platform with the introduction of 'Ideos' Android phones(Business daily, 2010). On the other hand, the large population of users with Java enabled phones still require privacy in their text messages. The lock message functionalities that comes with the mobile devices does not usually prevent from spying by malicious application such as Juju. Besides, studies show that users sent messages to wrong recipients without their consent. An encryption application would therefore solve this problems and hence the secure messaging APP.

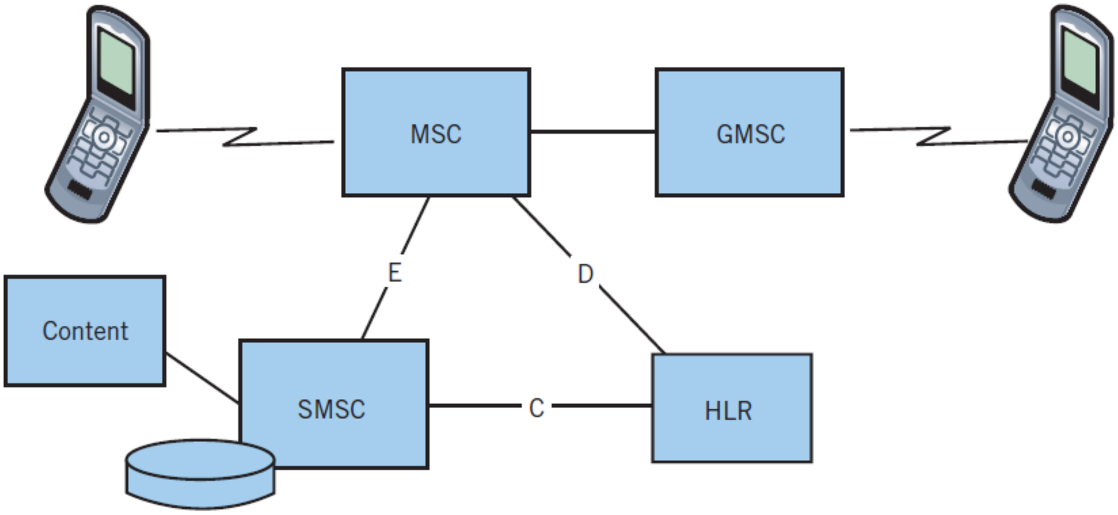


Fig.1 SMS architecure

The secure messaging application will run on both the sender's and the receiver's handsets. The users will be able to encrypt the Message at his/her handset using the application . The system will provide the users with a similar but different messaging menu for sending and receiving secure messages. The encrypted SMS is sent to the MSC which then forwards the message to the SMC in an encrypted format. The SMSC then looks at the recipient's number using the HLR, it then forwards the encrypted text to the nearby MSC which forwards the encrypted message for the recipient.

2.2.3 SMS Architecture

Short messages are delivered in GSM signaling channels between the Mobile Station (MS)* and the Base Transceiver Station (BTS). The messages flow as normal calls, but they are routed from the MSC to a Short Message Service Center (SMSC). The SMSC stores the message until it can be delivered to the recipient(s) or until the message's validity time has elapsed. The recipient can be a normal MS user or a SMS gateway. The gateways are servers that are connected to one or more SMSCs to provide SMS applications for the MS users. These applications include ring tone and icon delivery, entertainment, bank services, and many other beneficial services(NSS,2006).

SMS messages are handled via a *short message service center* that the cellular provider maintains for the end devices. The SMSC can send SMS messages to the end device using a maximum payload of 140 octets. This defines the upper bound of an SMS message to be 160 characters using 7-bit encoding. It is possible to specify other schemes such as 8-bit or 16-bit encoding, which decreases the maximum message length to 140 and 70 characters, respectively. Text messages can also be used for sending binary data over the air. Typically, specific applications on the phone handle messages that contain binary data—for example, to download ring tones, switch on and off animation, exchange picture messages, or change the look and feel of the handset's graphical user interface.The system can segment messages that exceed the maximum length into shorter messages, but then it must use part of the payload for a user-defined header that specifies the segment sequence information.

SMSCs operate in either a storeand- forward or a forward-and-forget paradigm. In the store-and-forward paradigm, the system resends the message for some period of time until it is successfully received. In a forwardand- forget paradigm, the system sends the message to the end device without assurance of receipt or an attempt to redeliver in the case of failure. The SMS protocol stack comprises four layers: the application layer, the transfer layer, the relay layer, and the link layer.

2.3 The encryption systems in use today

Since early 50s, there has been a growing interest in *securing sensitive digital data*, for the purpose of storing it safely, as well as transmitting it securely over unsecured communication channels (El-Khalil,2007). Bruce (Bruce,1996) demonstrated that 'the initial attempts at modern data security came in the form of the Data Encryption Algorithm (DEA), developed in the 1970's and adopted (with some modification) by the United States Government as the Data Encryption Standard (DES)'.

The application of various encryption systems have since then been adopted and are in use today. Among the most commonly used encryption system is Tripple DES and the RC5. Although many encryption standards exist, their scope is beyond this literature. Therefore this chapter only discusses DES and RC5 with the aim of achieving the objectives of this project. As illustrated by Gary, DES is a block cipher, with a 64-bit block size and a 56-bit key. DES consists of a 16-round series of substitutions and permutations. In each round, data and key bits are shifted, permuted, XORed, and sent through 8 S-boxes, a set of lookup tables that are essential to the DES algorithm(Gary, 2008). Decryption is essentially the same process, performed in reverse.

> However, in the recent years, the worries about DES security have spawned a series of DES implementations that are variations of the stock DES algorithm. The most commonly used variation is Triple-DES. Triple-DES is simply triple encryption using the DES algorithm.A 64-bit block is encrypted once with DES. The resulting cyphertext is then encrypted again with DES, and that resulting cyphertext is encrypted a third time, again using the DES algorithm.

Although this series of encryption offers greater security, it is also three times slower than the single DES implementation.

On the other hand RC5 is a symmetric encryption algorithm whose plaintext and ciphertext are fixed-length bit sequences (Rivest,1999). RC5's biggest advantage is its simplicity: encryption and decryption can each be implemented with only five lines of C code. In addition, due to the simplicity of RC5, it has extremely low memory requirements. RC5 also offers flexibility and versatility by giving users the option to change the number of rounds performed, key size, and block size. By adjusting these options users can manipulate the trade-off between speed and security. One unique aspect of RC5 is the usage of data-dependent rotational shifts. Data-dependent rotational shifts involve manipulating bits with the shift amount determined by a block of data, instead of a fixed integer value.

Each encryption and decryption function accepts two blocks (32 bits for each block) of data, either as ciphertext or plaintext.

When the data manipulation occurs, the function outputs two blocks of ciphertext or plaintext ,depending on whether the function encrypts or decrypts. Just like other algorithms, RC5 has a set-up process, and the set-up time is dominated by the creation of an expanded key table whose elements are used as a second argument in XOR operations during encryption and decryption. A thirty-two bit word, sixteen round, and four-byte sized key configuration was used in this analysis.

2.3.1 Why advanced encryption standard?

There are thousands of encryption algorithms, with source code available for study (Laurie, 2008). The decision to use AES is based on the status of AES as the currently accepted standard for data encryption, and its nearly ubiquitous use in encryption-offering software (and hardware). It is a thoroughly analysed and accepted algorithm, offering powerful encryption with a small key size. More so,Java Micro-edition for small devices(J2ME) comes with advanced encryption standard classes that can be applied during the development of mobile applications therefore AES is easier to implement in Java platforms as compared to DES which is much slower-depending on the size of the key.

CHAPTER THREE: METHODOLOGY

3.1 Introduction

Choosing the appropriate methodology for software projects always plays a huge role in determining the success of software product(Sommerville,2004). In the development of secure messaging mobile encryption application, various factors were put into consideration before settling on objectory use case approach that was proposed by L. Jacobson in 1994. The process involved identification of functional requirement from which use case artefacts were developed, after which, the dynamic and static behaviour of the system were analysed and modelled. The modelling of static behaviours was done through identification of objects and classes which were represented using unified modelling language(UML) diagrams. The dynamic aspects of the system were modelled using sequence, interactive, state diagrams and collaboration diagrams. Thereafter, implementation was done ,where the algorithm was implemented using genetic algorithm approach.

3.1.1 Implementation models

The project adopted Agile prototype development methodology. The secret messaging app was given to users at various stages .This made it to involve early development of the users interface to allow users to interact will the proposed mobile app (Martin, 2010).

3.1.2 Why Extreme Programming

Extreme Programming (XP) evolved from the problems caused by the long development cycles of traditional development models (Beck 1999a). It first started as 'simply an opportunity to get the job done' (Haungs 2001) with practices that had been found effective in software development processes during the preceding decades (Beck 1999b). After a number of successful trials in practice (Anderson et al. 1998), the XP methodology was "theorized" on the key principles and practices used (Beck 1999b). Even though the individual practices of XP are not new as such, in XP they have been collected and lined up to function with each other in a novel way thus forming a new methodology for software development. The term 'extreme' comes from taking these commonsense principles and practices to extreme levels (Beck1999b).

Since the project focused on end users of the mobile applications, XP will was the ideal methodology for it was centered on users.

3.1.3 The process

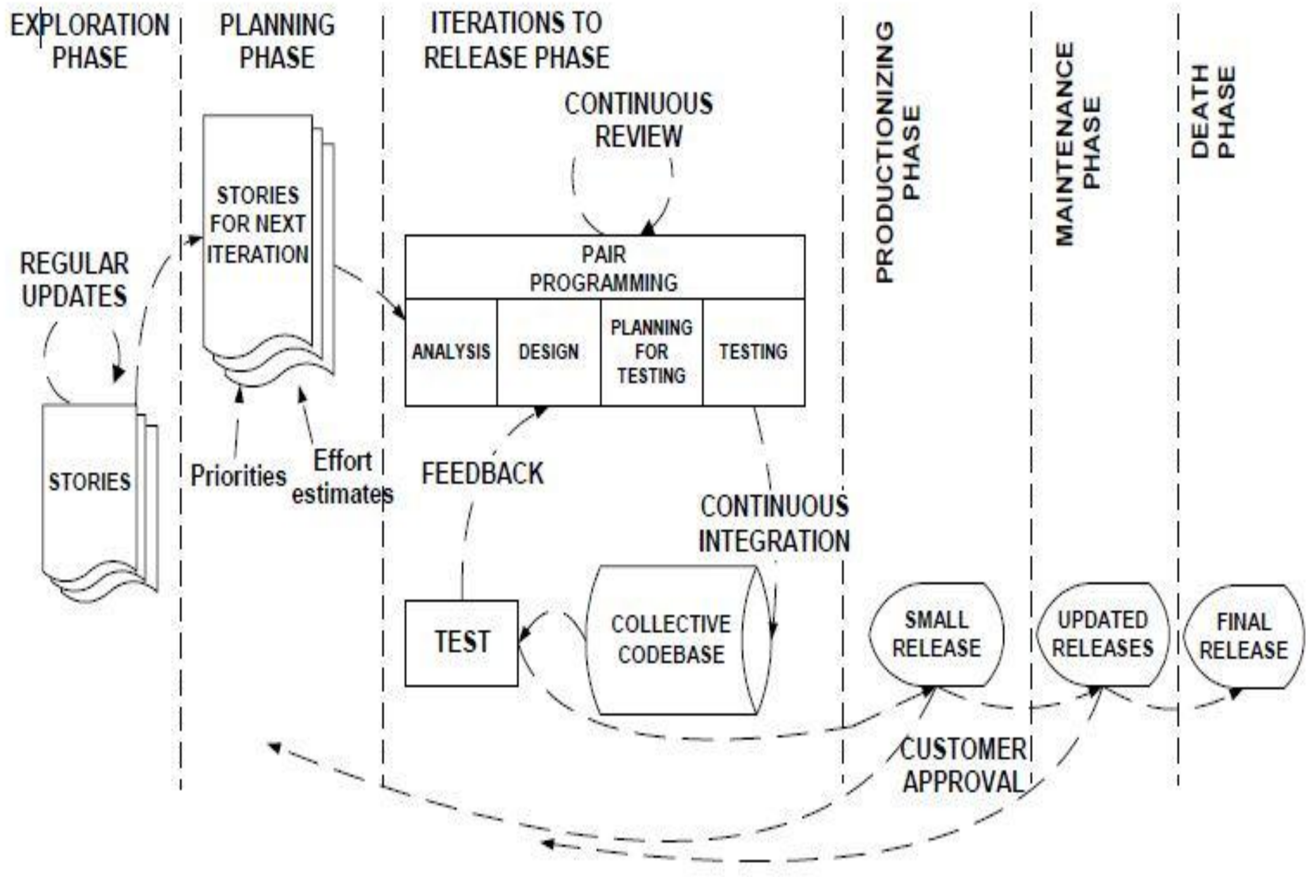


Figure 4. *Extreme Programming ;   source: Agile Software development Reviews*

At the **Exploration phase,** the users or students in this case were given story cards to write about what they would need the system to do to them. Each need formed a part of the features of the messaging application. At the same time I was getting familiar with the tools(Java Micro edition for Small Devices) and the available technology . They needed the story cards that they wished to be included in the first release.

At the Planning **phase,** I set the priority order for the stories and agreed with the users on the contents of the first release of the system. Coding of the first prototype began immediately and took approximately four weeks .

The **Iterations to release** phase included several iterations of the systems before the first release. The schedule set in the planning stage was broken down into a number of iterations that took each one to four weeks to implement. The first iteration created a system with the architecture of the whole system. This was achieved by selecting the stories that enforced building the structure for the whole system. At this point the users helped in deciding the stories to be selected for each iteration. In addition to that, the functional tests created by the users ran at the end of every iteration.

At the end of the last iteration the system wa ready for production.

Productionizing phase –at this phase extra testing and checking of the performance of the system carried out before the system was released to the users. New changes that existed determined whether the inclusion of the features in the other versions were be released.

Maintenance phase – this stage involved training and support for the users as the system ran on their handsets. Hands on experience was also put in check to record any difficulties that the users were facing.

The **Death phase** -this project is limited to the first version because of the time constraint and resource constraints. The project might therefore not meet the dead phase where users will not require any more needs from the mobile application

3.2 The architecture of the proposed system

The mobile application is a multi-layered application consisting of user experience, business and data layers. The application was developed is a rich client .The business and data layer services are located on the device itself. The presentation layers contains the user interface components(messaging menu) by which the user is able to compose the messages, read inbox and interact with other items. The business layer holds the encryption and decryption logic. It also defines the application facades and other customizable components for integrating with existing mobile banking. On the other hand, the data layer is responsible for storing incoming messages and making the messages available through the cache. The figure 2 below shows the secure messaging mobile application architecture with components grouped by the areas of concern.

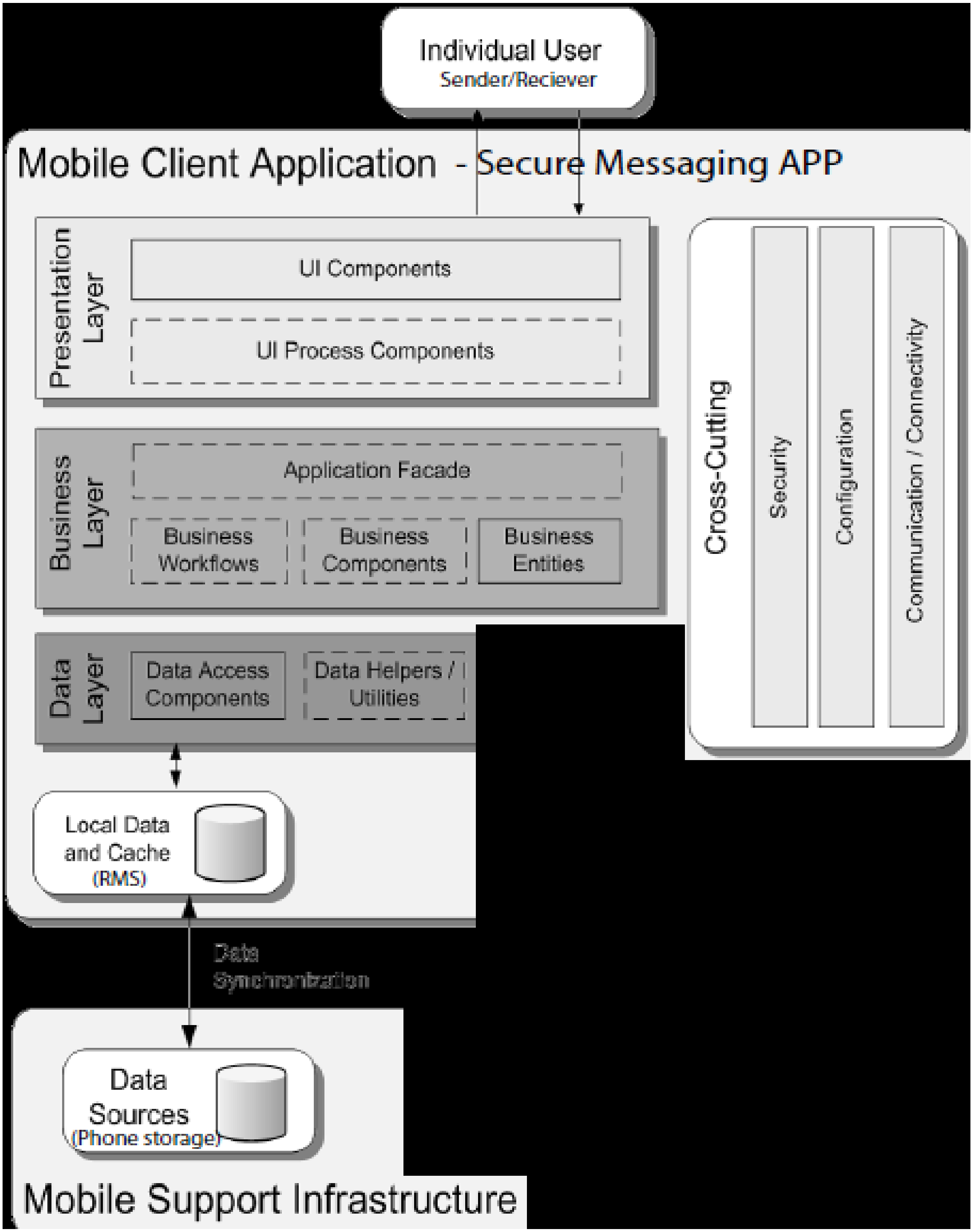
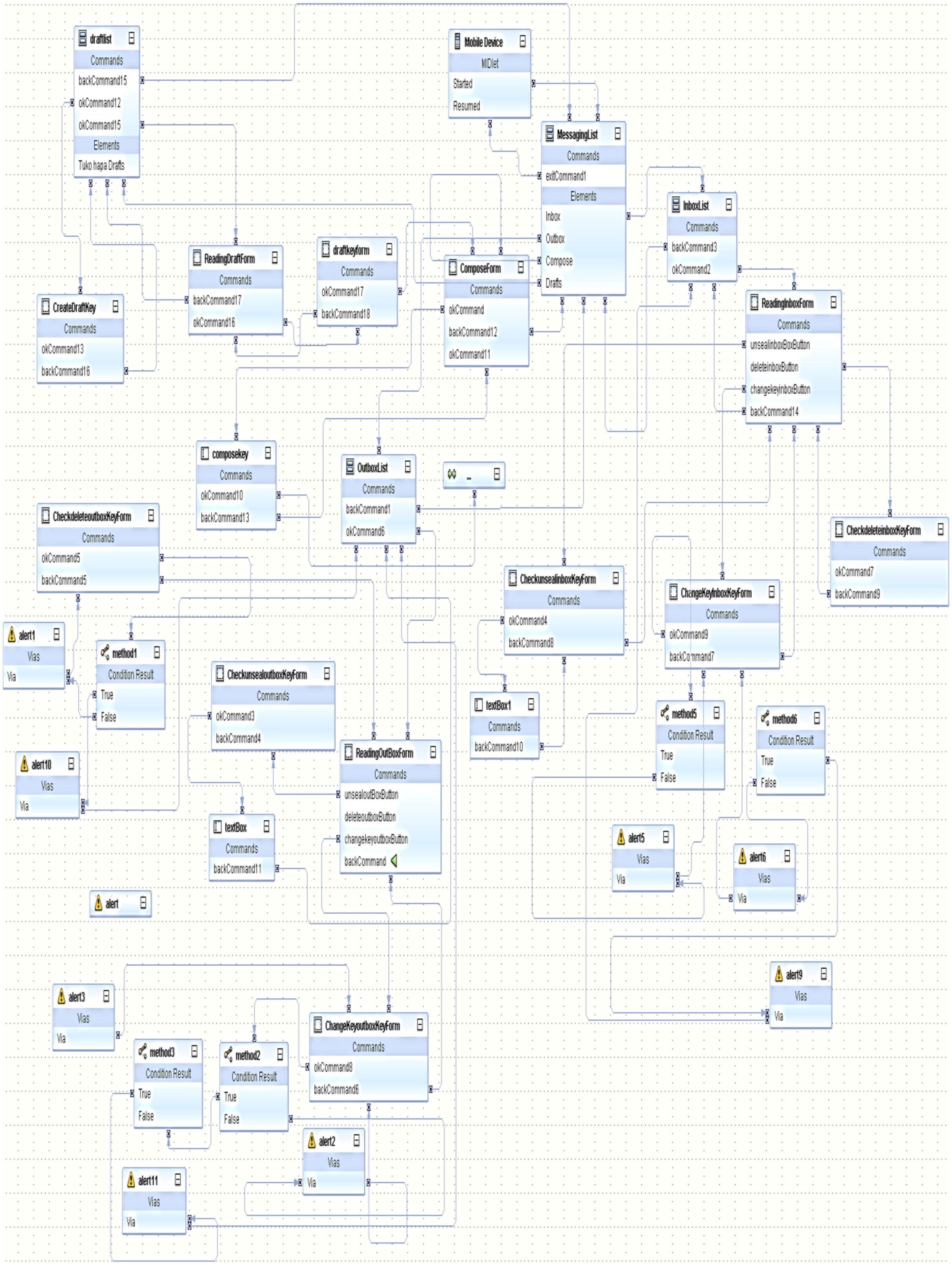


Figure 2: The proposed system architecture

3.2.1 System Logical flow .



3.2.2 Use Case scenario for Messaging Application

Use case name: send Message

Event/Trigger: Users with Java enabled Phones

Brief description: The java application is started to load the secure messaging menu. Messages are composed and secret keys added to convert the plain text to cipher text then messages are sent via the subscriber network

Actors : Java enabled phone, network subscribers

Preconditions: The user must have sufficient amount of money on their phones before sending text messages from their phones. Both the secret Keys must be known to the sender and recipient. The application must be installed on both the recipient's phone and the sender's phone.

Post conditions: Delivery report for the short encrypted message delivered.

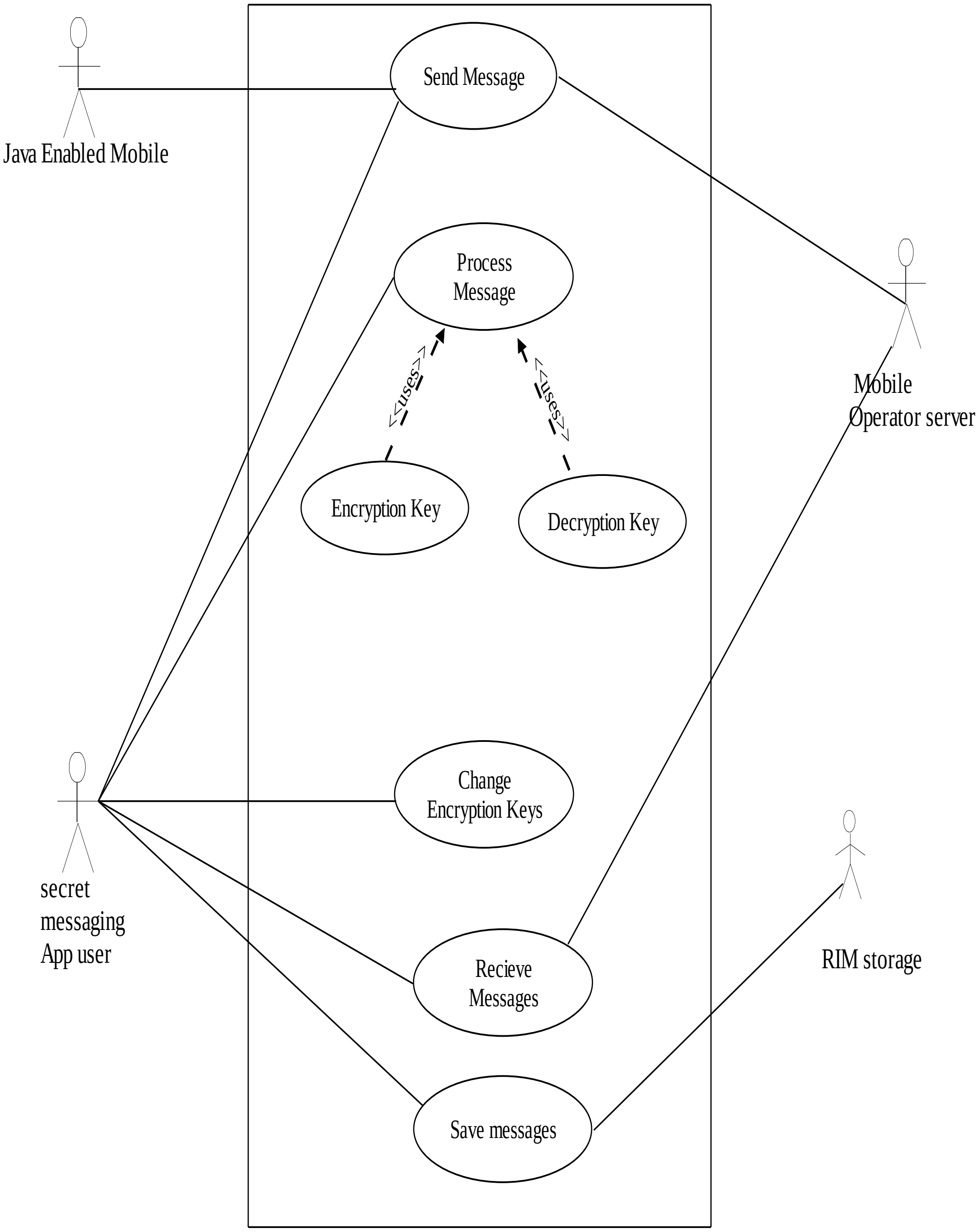


Figure 4: Use diagram of secure messaging app

3.2.3 Level 1: Interaction diagram

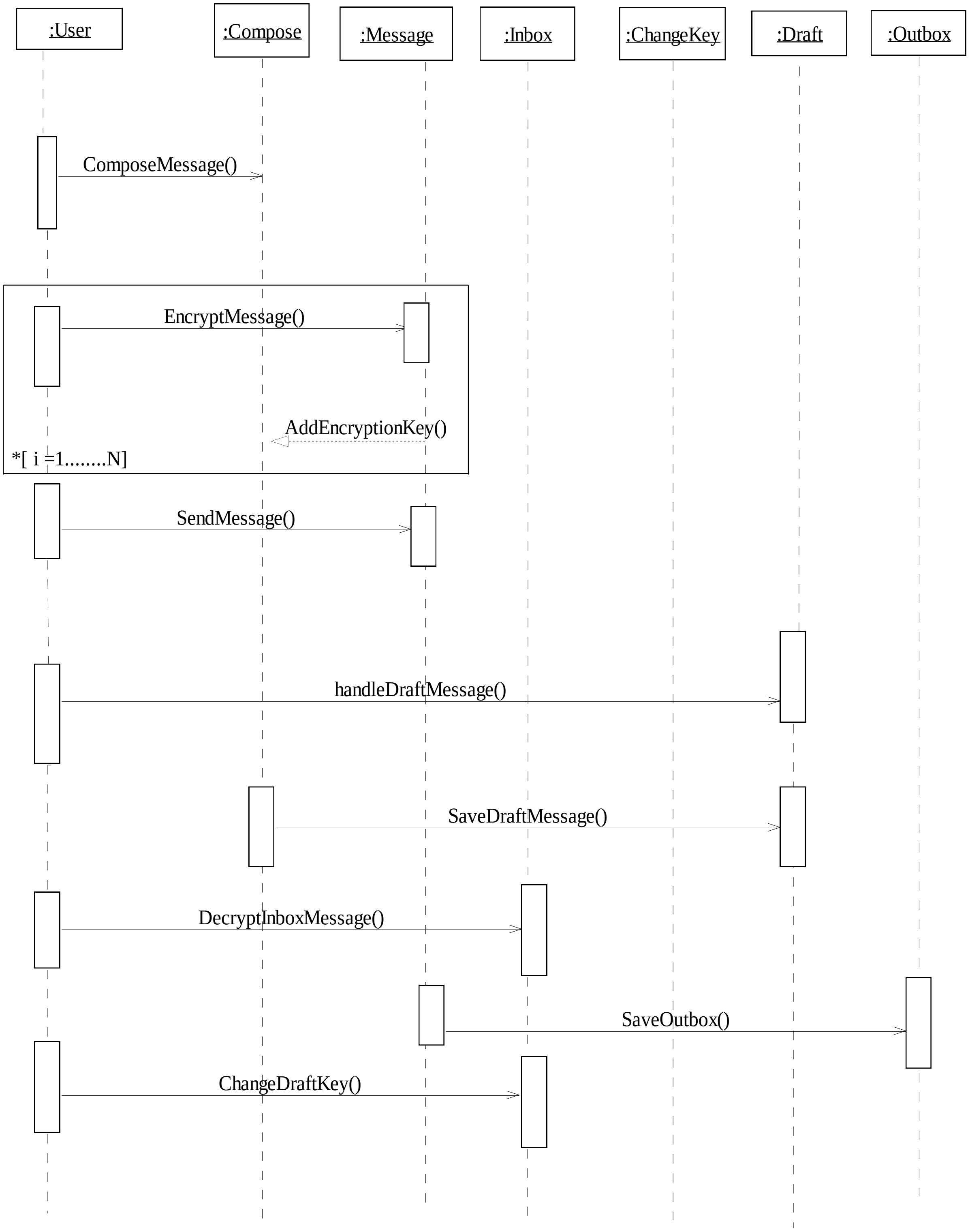


Figure 5: interaction diagram

3.2.4 Tools used for implementation

Operating system:	Windows 7
Approach:	Object-Oriented
Analysis and Design tools:	ArgoUML and SMART Draw
Programming Language:	Java Micro-Edition for small devices (J2ME)
CASE TOOLS:	Netbean 7.0 development environment JAVA Development Kit Version 7 JDK Mobile Emulators
Testing Tools (Mobile Phone Used for Testing):	Nokia 6030, Tecno T25

CHAPTER FOUR: RESULTS AND EVALUATION

4.1 Implementation

The secure messaging application was developed using Java Micro-edition for small devices(J2ME). The version 1.0 of the application targeted users with low end java enabled phones owing to the fact that large number of users(32%) in Kenya own Java enabled phones(Nokia,2012). Netbeans 7.0 was used as the development environment with JDK 3.0 emulators for testing the application. Installation on the phone(Nokia 6030) was done by transferring jar and jad files to the phone via Bluetooth. Other phones(Sony Erickson, China made phones and LG phones) were used for testing and the results are shown below.

4.2 Results

The experimental results were carried out by installing the system on various mobile devices models. The messages were then sent and the results deduced.

4.2.1 Research design

A survey is a powerful and effective tool that can be used to collect data about human attitudes, behaviors, and characteristics. The survey questionnaire is often adopted by researchers as an effective hypothesis testing method (Cavana 2001). A survey was therefore employed in this project in order to determine the need and demand for privacy and security in the current short messaging system.

A survey based on clusters was conducted within Chiromo Campus of Nairobi University. The survey was appropriate for this kind of study as it provided a quantitative description of attitudes, experience and opinions of the sample population (Erastus T, 2011). In addition to that, a descriptive research design was incorporated in order to collect more data on the user experiences in the current messaging application.

4.2.2Target population

The population of the research was mainly the University of Nairobi’s Chiromo Campus computer science students. The research was focused on this population because of the convenience in terms of location, the possible access to the mobile service and their background in information technology.

4.2.3 Sampling design s

The researcher employed purposive sampling. This method was employed because of its ability to allow the researcher to deliberately select particular users (in these case computer science students). A research questionnaire was distributed to a selected sample of the students. The questionnaire included the construct items related to the user interactions with the current system (SMS services) and the challenges that the users face when using the existing system.

The respondents were required to complete the questionnaire voluntarily. A total of 180 questionnaires were distributed at specific points. Where necessary, interviews were carried out in order to obtain in-depth user experiences.

4.2.4 Data collection procedure and instruments

Data was collected using self-administered questionnaire. In order to solicit uniform data from the respondent; similar questions were included in the questionnaire. The researcher opted to use self-administered questionnaires because of the flexibility in terms of time. The questionnaires contained both closed and open ended questions seeking items that would meet the research objectives and answer the research questions.

4.2.5 Data analysis and presentation

Quantitative data analyzed by using descriptive statistics and other standard quantitative methods (Kontio *et al*. 2004). Data collected from the survey was entered into the statistical package, SPSS (statistical package for social science) for analysis, discussion and presentation of the results in this research. To analyze the demographic information, the descriptive statistics was entered onto a Microsoft Excel sheet.

4.2.6 Compatibility level

The application was installed on various phones to test its compatibility level and the user interface(UI). The secure messaging was developed in CLDC 1.1 and MIDP 2.0. This settings therefore played a role in determining the UI when installed on other phones. Other optional configuration settings that determined installation are:

- Advanced multimedia supplements API 1.0
- Java Bluetooth API for mobile devices
- Wireless messaging API 2.0

- Security and Trust services API for J2ME
- Content Handler 1.1

The following results was concluded after successful installation of the secure messaging application as shown below.

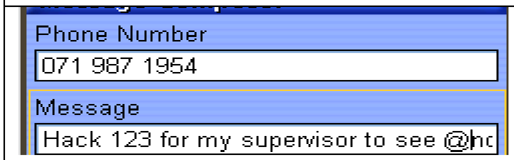
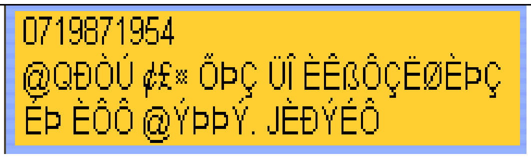
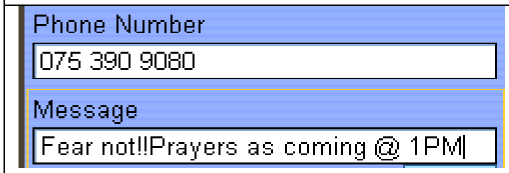
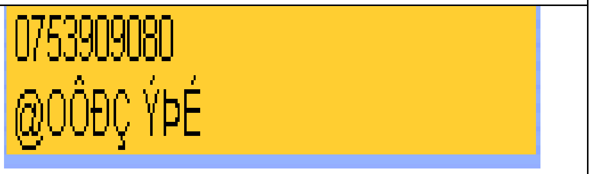
Name of Participant	Phone model	OS/Java capabilities	compatibility	Comments
kinyanjui	Mobile Emulator JDK 3.0 devices	Yes	Successful Installation	Very good. Can safeguard money transfer information
Luther	Nokia 6030	Yes	Successful Installation.The application adopts Phone's UI	Very good. Potential for wide applications.
Laura	ZTel(China model)	Yes	Successful Installation. The application adopts Phone's UI	Very good. Sell it to safaricom
Omondi	Sony Erickson	Yes	Failed Installation. Java not compatible with the MIDP CLDC 2.1	Not happy. Could have used this with my friends to send information that we desire about

				politics
Paul	Huwawei Safaricom phone('Kabambe 3G')	Yes	Successful Installation. The application adopts Phone's UI	Interesting. Will want to buy it.
Alex	Techno T25	Yes	Successful Installation. The application adopts Phone's UI	Wonderful! Now parents can not track our communications with my buddies

Table 1.0 compatibility levels

4.2.6.1 Sample plain-text SMS, encrypted SMS, deciphered SMS

The system was tested to determine whether the encryption and decryption functionalities were working effectively. Table 2.0 below shows the various system results

Plain text	Encrypted text
 Phone Number 071 987 1954 Message Hack 123 for my supervisor to see @hc	 0719871954 @QĐÒÚ ¢£※ ŐÞÇ ÜÎ ÈÊßÔÇËØÈÞÇ ÉÞ ÈÔÔ @ÝÞÞÝ. JÈĐÝÉÔ
 Phone Number 075 390 9080 Message Fear not!!Prayers as coming @ 1PM	 0753909080 @OÔĐÇ ÝÞÉ

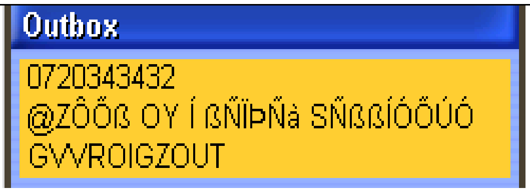
0720343432 This IS a secret messaging Application.	 Outbox 0720343432 @ZÔÕß OY Í ßÑÏÞÑà SÑßßÍÓÕÚÓ GVVROIGZOUT

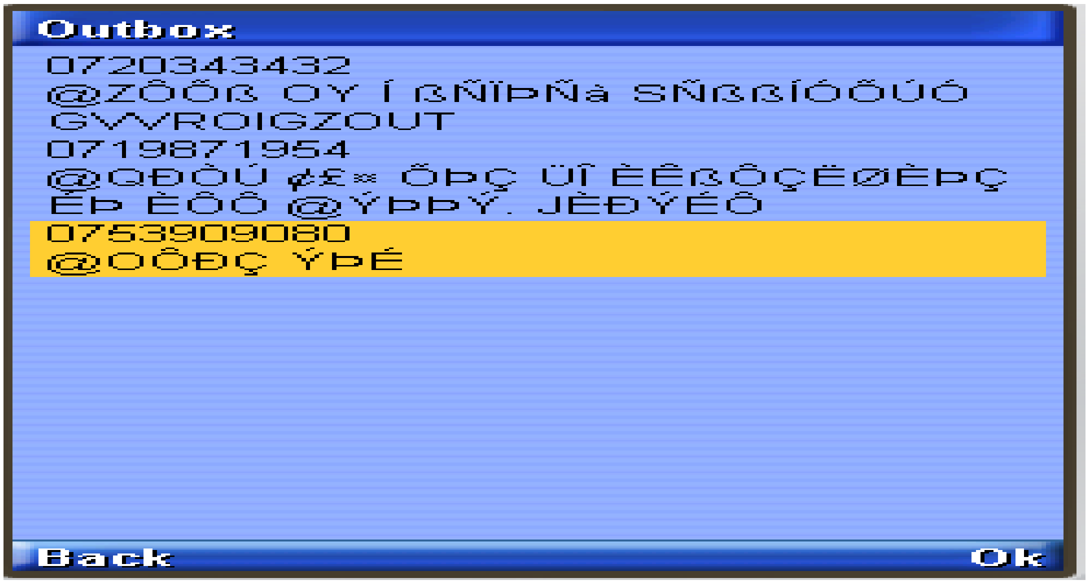
Table 2.0 system results

4.2.6.2 Message Delivery Report and encryption reports

```
The NumberFound boolean variable=false

That number was not found and now we are writting

3 is lenth of mn

[INFO] [sms    ] ## javacall: SMS sending...

0719871954:QÐÒÚ ¢£¤ ÕÞÇ ÜÎ ÈÊßÔÇËØÈÞÇ ÉÞ ÈÔÔ @ÝÞÞÝ. JÈÐÝÉÔ:Tue Jan 29 11:24:17 GMT+03:00 2013

4 is lenth of mn

phone=0720343432body=@ZÔÕß OY Í ßÑÏÞÑà SÑßßÍÓÕÚÓ GVVROIGZOUT

phone=0719871954body=@QÐÒÚ ¢£¤ ÕÞÇ ÜÎ ÈÊßÔÇËØÈÞÇ ÉÞ ÈÔÔ @ÝÞÞÝ. JÈÐÝÉÔ

phone=0719871954body=@QÐÒÚ ¢£¤ ÕÞÇ ÜÎ ÈÊßÔÇËØÈÞÇ ÉÞ ÈÔÔ @ÝÞÞÝ. JÈÐÝÉÔ

0753909080:OÔÐÇ ÝÞÉ!!YÇÐÎÔÇÈ ÐÈ ÒÞÜØÝÖ @ ¢YV:Tue Jan 29 11:35:37 GMT+03:00 2013

key

Have now enumerated key records

key record enumerated 0720343432@1212!

Equality=false

key recordlength=32

From interface=753909080 From Database=(720343432)

The key is=1212 of length=4

key record enumerated 0720343432@1212!

Equality=false

key recordlength=32

From interface=753909080 From Database=(720343432)

The key is=1212 of length=4

key record enumerated 0719871954@1212111!
```

```
Equality=false

key recordlength=38

From interface=753909080 From Database=(719871954)

The key is=1212111 of length=7

outbox

Exception is  0

The keyFound boolean variable=true

The keyFound boolean variable=false

The NumberFound boolean variable=false

That number was not found and now we are writting

5 is lenth of mn

[INFO] [sms     ] ## javacall: SMS sending...


0753909080:OÔĐÇ ÝÞÉ!!YÇĐÎÔÇÈ ĐÈ ÒÞÜØÝÖ @ ¢YV:Tue Jan 29 11:35:37 GMT+03:00
2013

6 is lenth of mn

phone=0720343432body=@ZÔÕß OY Í ßÑÏÞÑà SÑßßÍÓÕÚÓ GVVROIGZOUT

phone=0719871954body=@QĐÒÚ ¢£¤ ÕÞÇ ÜÎ ÈÊßÔÇËØÈÞÇ ÉÞ ÈÔÔ @ÝÞÞÝ. JÈĐÝÉÔ

phone=0753909080body=@OÔĐÇ ÝÞÉ
```

4.2.6.3 Outbox Summary

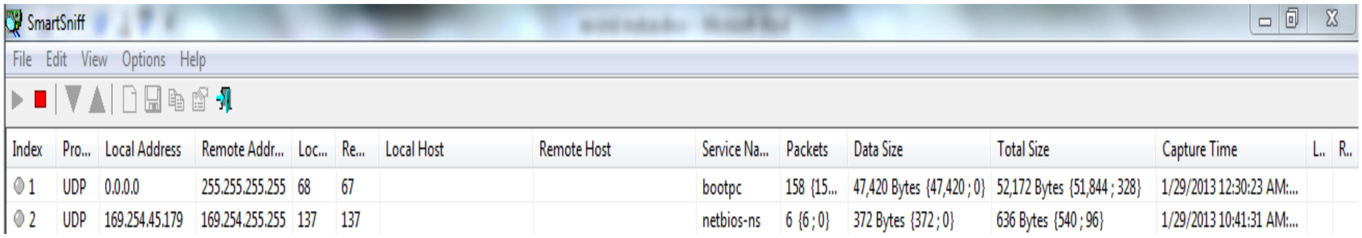


0720343432
@ZÔÕÕß OY Í ßÑÏÞÑà SÑßßÍÓÕÚÓ
GVVROIGZOUT
0719871954
@QÐÒÚ ¢£¤ ÕÞÇ ÜÎ ÈÊßÔÇËØÈÞÇ
ÉÞ ÈÔÔ @ÝÞÞÝ. JÈÐÝÉÔ
0753909080
@OÔÐÇ ÝÞÉ

Figure 6: outbox summary

4.2.6.4Changes in size of the messages packets using sniffer hacking tool

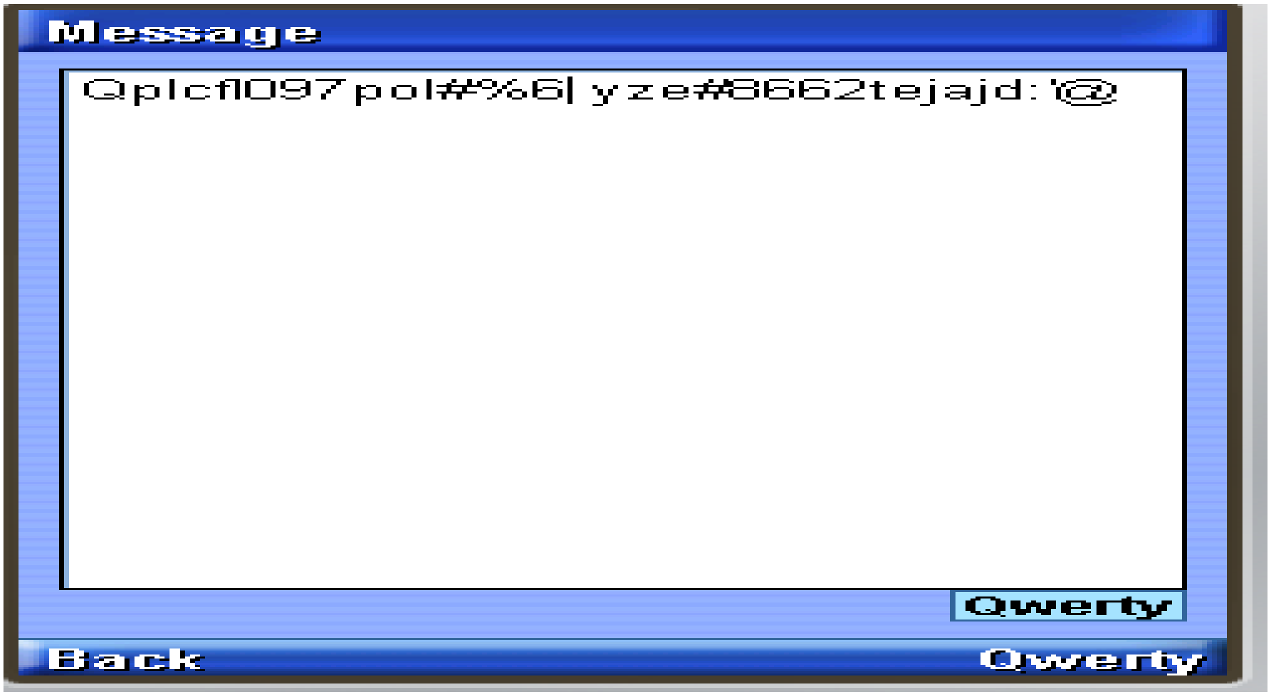
Smart sniff was able to capture SMS packets. However, the hacking tool was unable to crack into the encrypted packets. For example, the attempt to crack into the following packet using smart sniff produced the following results



Index	Pro...	Local Address	Remote Addr...	Loc...	Re...	Local Host	Remote Host	Service Na...	Packets	Data Size	Total Size	Capture Time	L..	R..
1	UDP	0.0.0.0	255.255.255.255	68	67			bootpc	158 {15...	47,420 Bytes {47,420 ; 0}	52,172 Bytes {51,844 ; 328}	1/29/2013 12:30:23 AM:...		
2	UDP	169.254.45.179	169.254.255.255	137	137			netbios-ns	6 {6 ; 0}	372 Bytes {372 ; 0}	636 Bytes {540 ; 96}	1/29/2013 10:41:31 AM:...		

Figure 7: smart sniff results

Using a wrong key also decrypts the text messages further as shown below.



Qplcfl097pol#%6| yze#8662tejajd:'@

Figure 8: Encrypt further

4.3 Analysis of System Results

The system results was observed in terms of compatibility with various phones, the ability to encrypt and decrypt text messages without adding the size of the SMS, the time interval for sending and receiving messages and finally, the robustness of the secure messaging application.

CHAPTER FIVE: CONCLUSION AND DISCUSSION

5.1 Introduction

This chapter focuses on the conclusion drawn from the system and the discussion of the results.

5.2 Achievement of Research questions

The existing SMS architecture is prone to attackers who intercept messages to reveal its content and hence no security. With increasing number of hacking tools such as juju and Smart Sniff, its' evident that security in SMS architecture needs to be enhanced. It is known that SMS travels as plain text and privacy of the contents of SMS cannot be guaranteed, not only over the air, but also when such messages are stored on the handset.

This project therefore set out to investigate the available security algorithms and methods that are used to secure messages in SMS architecture. The SMS architecture is composed of short messages centers (SMSC), Home location register (HLR),mobile switching centres (MSC),SMS-gateway and the visitors location register(VLR). However, it's important to note that the contents of SMS messages are visible to the network operator's systems and personnel. Therefore, SMS is not an appropriate technology for secure communications in business transactions. On the other hand, secure messaging application may be the appropriate solution for the insecure messaging at hand.

5.3 Achievement of Objectives

This project achieved its main objective that was set to develop a secure messaging tool that enhances security and data integrity of SMS. The application developed was able to apply the advanced encryption concepts to encrypt and decrypt the text messages.

5.4 Discussion of Results in Relation to objectives

5.4.1 Objective1: Develop a secure messaging application

The project objectives were very specific as demonstrated on the proposal. The project achieved its main objective of developing a secure messaging tool/application that would ensure encrypted and secure sending of short messages. The application developed was tested using smart sniff. The secure messaging application can send ,receive, encrypt, decrypt and change messages key as per objective.

5.4.2 Objective 2: Test the secure messaging application with the existing mobile devices and smart Sniff hacking tool

Smart Sniff version 3.2 has the capacity to capture and display the short messages packets sent over the network. Although famously used for sniffing SMS, there are other tools for example 'juju' that are more powerful and can sniff the packets as long as its installed on the target phone. The secure messaging application was able to prevent users from such attacks. The sniffer is able to capture the packets but has no capacity to crack into the encrypted SMS.

Compatibility with various models of phones that run on Java determined the successful installation of the application. It's worth to note that , not all Java enabled phones were compatible with the application. The phone requirements included MIDP 2.0 or MIDP 2.2 and CLDC 1.0 and above. Most Chinese made mobile devices and Nokia have these features. However, the researcher would recommend Nokia 3500, 6030, 6200 series among other Java enabled phones because of ease of installation and operation.

5.5 Challenges

The project ensured the development of a secure messaging application that targeted Java enabled phones. However, with the rising number of smart phones, users with high end phones may find this application incompatible with their mobile devices.

The application adopted symmetric encryption concepts where the same key is used to decrypt and encrypt messages. Therefore, Key distribution remains a challenge as the sender and the receiver must know the keys prior to exchanging of messages. `

Although the application gives room for the changing of the Keys, in an event that an attacker identifies the secret keys, he/she can change the keys without the knowledge of the users hence a limitation to the system. Therefore, future research and development needs to be carried to ensure that asymmetric version of the application are developed.

5.6 Conclusion

The outcome of the system evaluation showed that the system could send and receive messages securely encrypted messages. Experimental results showed that the system has the capacity to decrypt and encrypt messages without adding the size of the packet. Unlike the normal SMS that are sent in plain text via GSM network, messages sent via secure messaging APP remained encrypted during the transit.

Future works can be focused on improving the application to accommodate asymmetric encryption systems and improve compatibility to other versions of mobile devices and operating systems.

Appendix i : REFERENCES

Baron, S., Patterson, A., & Harris, K. (2006): Beyond technology acceptance: understanding consumer practice. International Journal of Service Industry Management. Vol. 17 No.2, 2006. pp.111-135. Emerald Group Publishing Limited.

Tonny Omwansa (2009). *Innovations / Mobile World Congress 2009*. Available at: http://www.strathmore.edu/pdf/innov-gsma-**omwansa**.pdf

Beck, K. (1999a). Embracing Change With Extreme Programming. IEEE Computer 32(10): 70-77.

Beck, K. (1999b). Extreme programming explained: Embrace change. Reading, Mass., Addison-Wesley.

Bruce S.(1996) *Applied Cryptograph Second Edition*. John Wiley & Sons, Inc.

Cavana, R. Y., Delahaye, B.L, Sekaran, U. (2001). *Qualitative Data Gathering – Assumptions of Qualitative Research*. In Applied Business Research: Qualitative and Quantitative Methods (pp. 134 - 137). Milton, QLD: J. Wiley.

[Elk07] R. El-Khalil, A. D. Keromytis, *Hydan: Hiding Information in Program Binaries*, The 9th International Conference on Information and Communications Security (ICICS 2007), Zhengzhou, China, 2007

Herzberg, A. 2003. Payments and banking with mobile personal sevices. *Communications of the ACM* .Volume 46, Issue 5 (May 2003) Wireless networking security Pages: 53 58 ISSN: 0001-0782

Jeff B., Bill S., and Vetter .R,(2007) *SMS: The Short Message Service*. California Retrieved from www.dreamfabric.com/sms on 23rd August 2012

Lawrie B.(2008) *Cryptography notes* available at www.cisa.umbc.edu/cmsc/487/slides/ch03.ppt

(Kontio *et al*. 2004). *Experiences in the Software Engineering Context*. In Guide to Advanced Empirical Software Engineering (pp. 98- 99).

Martin Fowler. "Using an Agile Software Process with Offshore Development". Available at Martinfowler.com. Retrieved 6 June 2012.

MobiInfo(2012). *mobile information trends* Retrieved 23rd september 2012 available at : http://www.mobifone.com.vn/web/en/services/mobiinfo.jsp

Mocana(2012). Mobile security Retrieved 23rd september 2012 available at : http://www.mocana.com

Nysveen, H., Pedersen, P.E., & Thorbjornsen, H. (2005): Intentions to Use Mobile Services: Antecendents and Cross-Service Comparisons. Academy of Marketing Science. Journal;

Summer 2005; 33, 3;ABI/INFORM Global.

Njenga, A. D. K.(2009). Mobile phone banking: Usage experiences in Kenya. <http://www.strathmore.edu/pdf/ictc-08/mobile-banking.pdf > Accessed on 30th August, 2012.

Safaricom, (May, 2009), *Financial Year 2008/2009; Annual Results Presentation and Investor update*

Kipchumba Chepken (2004). Cellular Communication Security System (CCSS). *Available at : University of Nairobi Library.*

Erastus T, (2011). M-Pesa Utility, Operation And Entrepreneurial Innovations By Small Enterprises In Kenya. Available at: *http://www.aibuma.org/archive/proceedings2011/aibuma2011-submission 240.pdf*

Nokia (2012). Mobile Java, Shiny And New: Nokia Asha And Nokia SDK 2.0. Available at: *http://terrencebarr.wordpress.com/2012/06/29/mobile-java-shiny-and-new-nokia-asha-and-nokia-sdk-2-0/*

(Gary *et al*.2011). Transitions: Recommendation For Transitioning The Use Of Cryptographic Algorithms And Key Lengths. Available at: http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf

Appendix ii : USER MANUAL

A)Messaging Menus

Step 1: Go to Java Applications on your Phone and choose Secure Messaging then click OK. The following messaging platform will appear. (NB This is quite different from the messaging menu that comes with the phone)

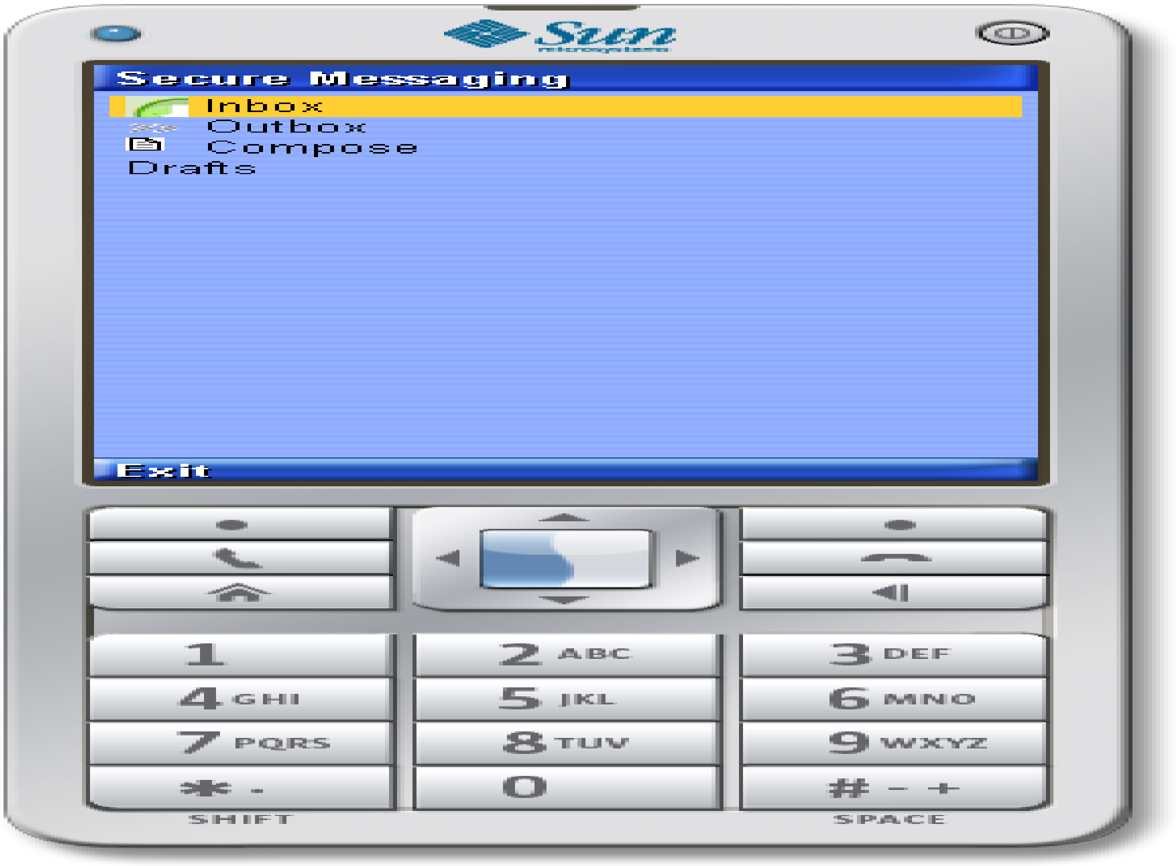
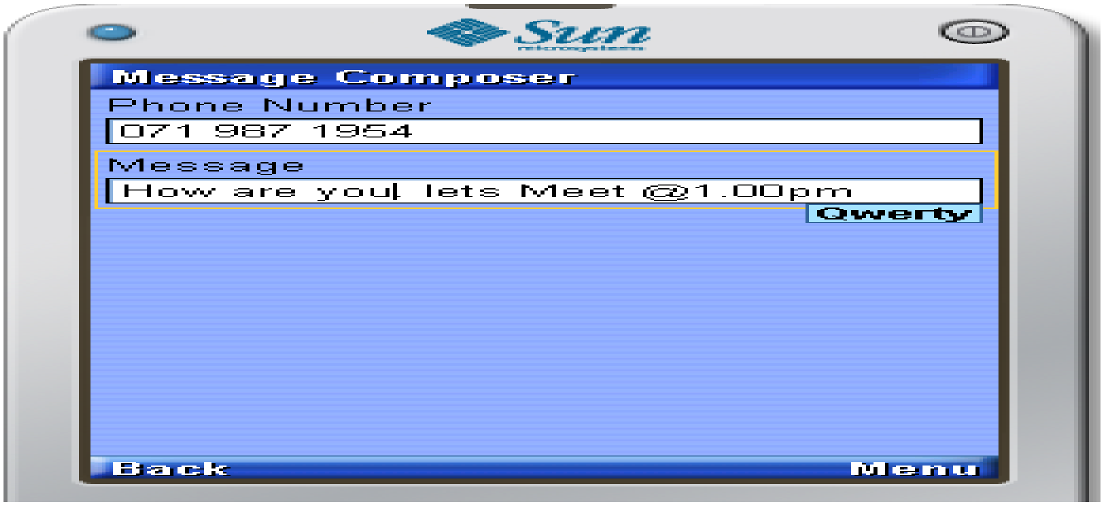


Figure 9 : messaging interface

Your phone will ask you if you want to allow the application to make use of messaging and network resources,click YES.

Step 2: Go to compose from the messaging menu and click oK. Enter the recipients phone number then type in your text message to be sent .



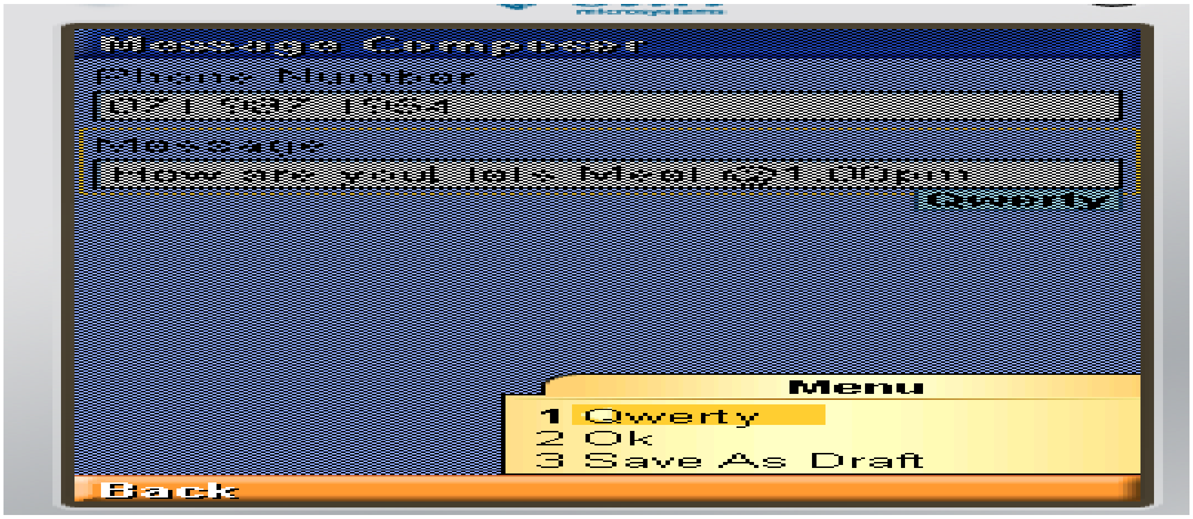
Message Composer

Phone Number: 071 987 1954

Message: How are you| lets Meet @1.00pm

Figure 10: compose message

Step 3: Click on the menu and choose ‘ok’ if you want to send or ‘Save As Draft’(see step 6 for creating Draft Keys)



Message Composer

Phone Number: 071 987 1954

Message: How are you| lets Meet @1.00pm

Figure 11: Add secret keys

Step 4: Add the secret KEY for encypting the messages before sending then click OK. NB)both the sender and the reciver must share their secret keys.

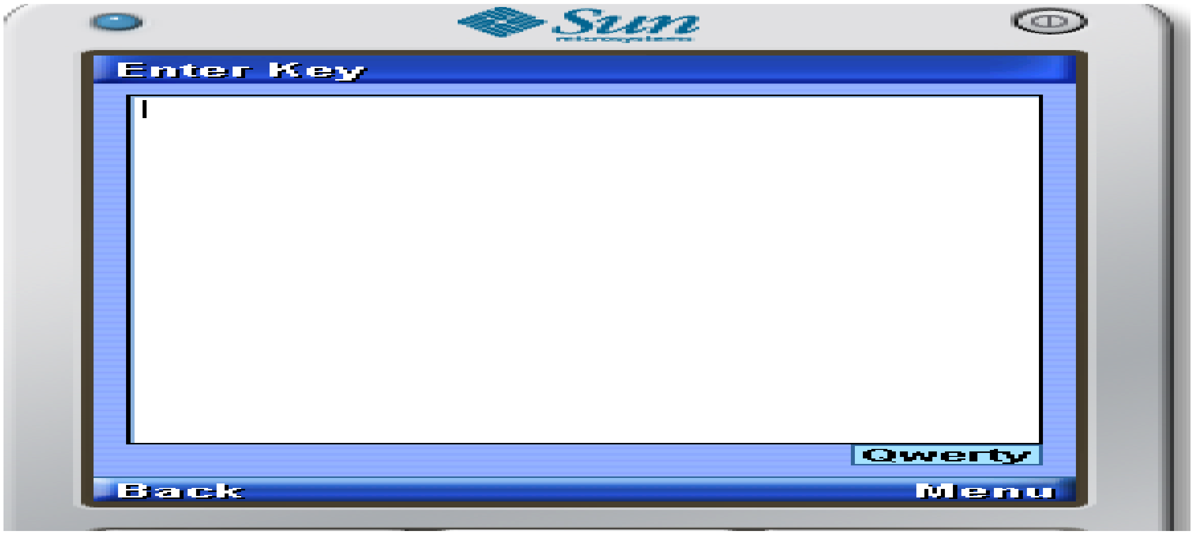
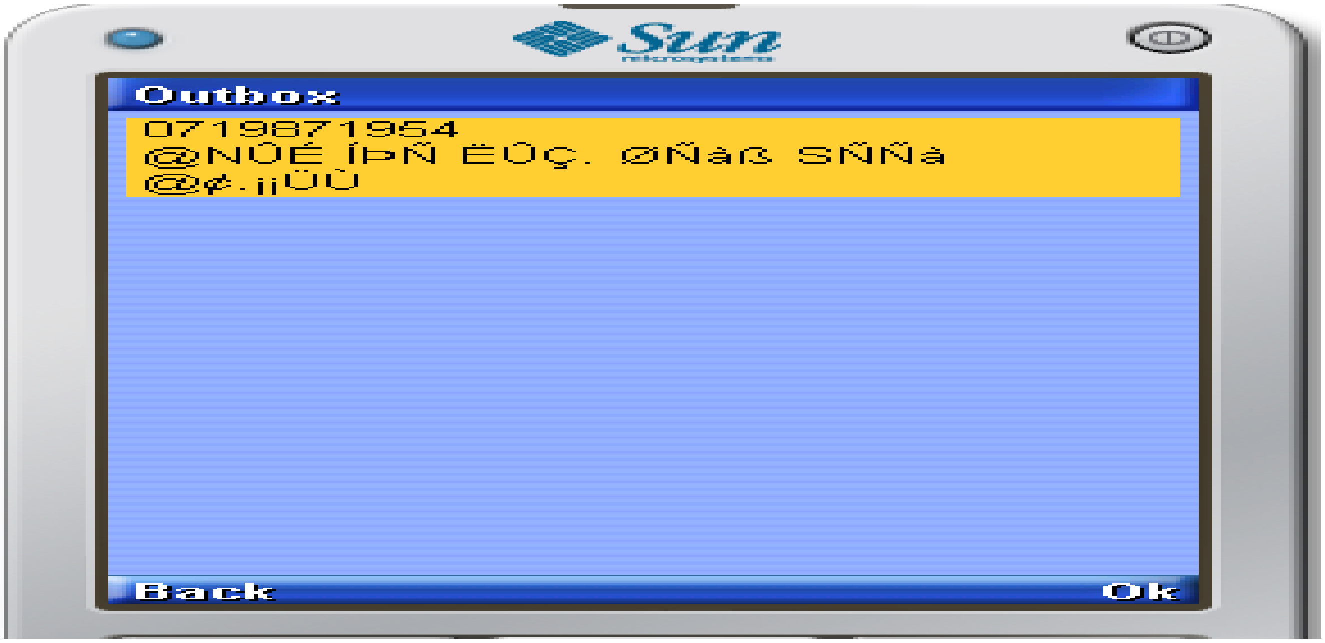


Figure 12: Enter secret keys

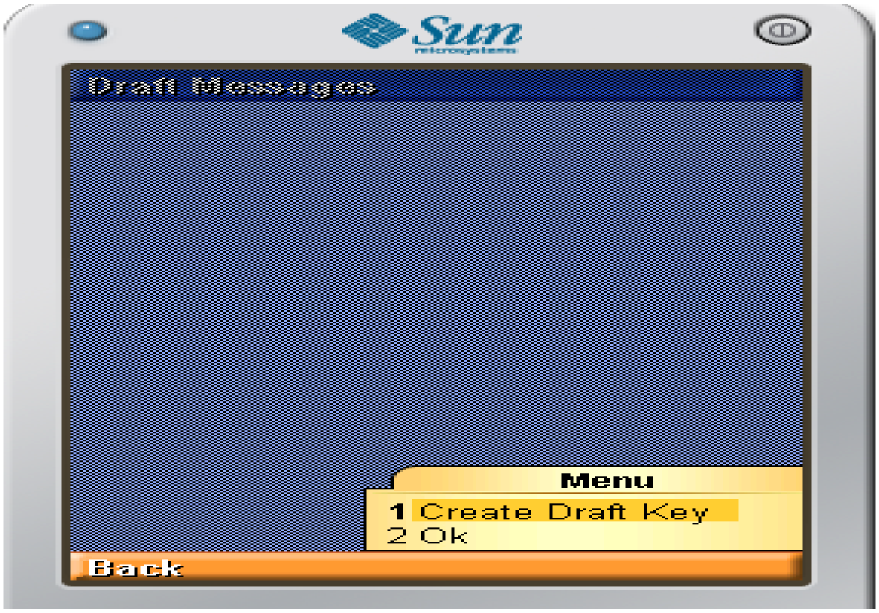
Step 5: Once you click OK your message will be sent. Check OutBox to confirm that your messages has been sent as shown below



0719871954
@NÜÉ ÍÞÑ ËÜÇ. ØÑaß SÑÑa
@¢.¡¡ÜÜ

Figure 13: sent items

Step 6: you must create a Draft Key for encrypting and protecting your draft messages



Draft Messages

Figure 14: create draft keys

Step 7: To read inbox, click inbox>Unseal>enter Key then access your messages.

If you input wrong keys, the application encrypts the messages further and you won't be able to access the.

You can change your secret keys in case you suspect breaching as shown in step 8

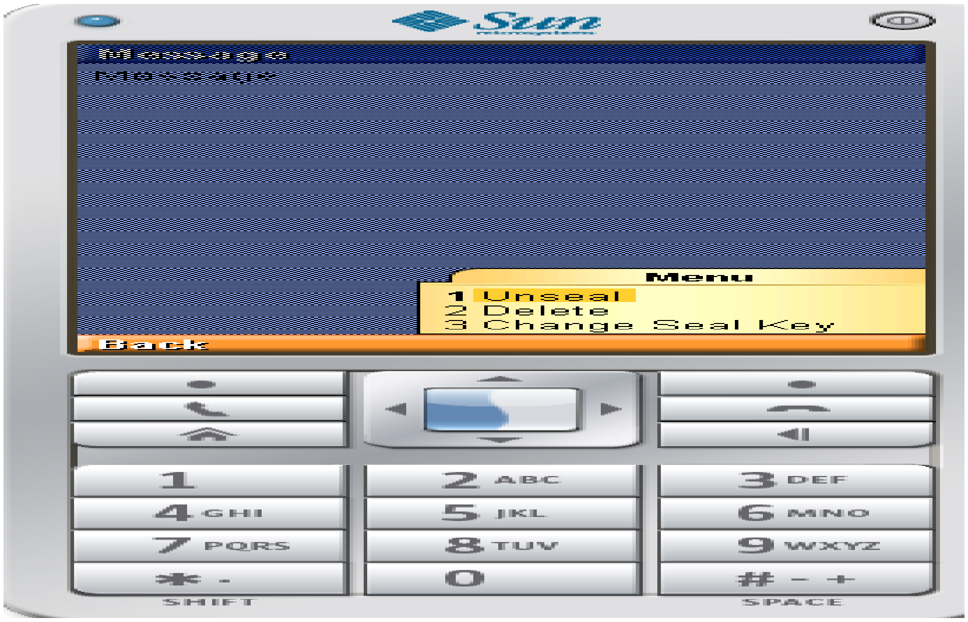
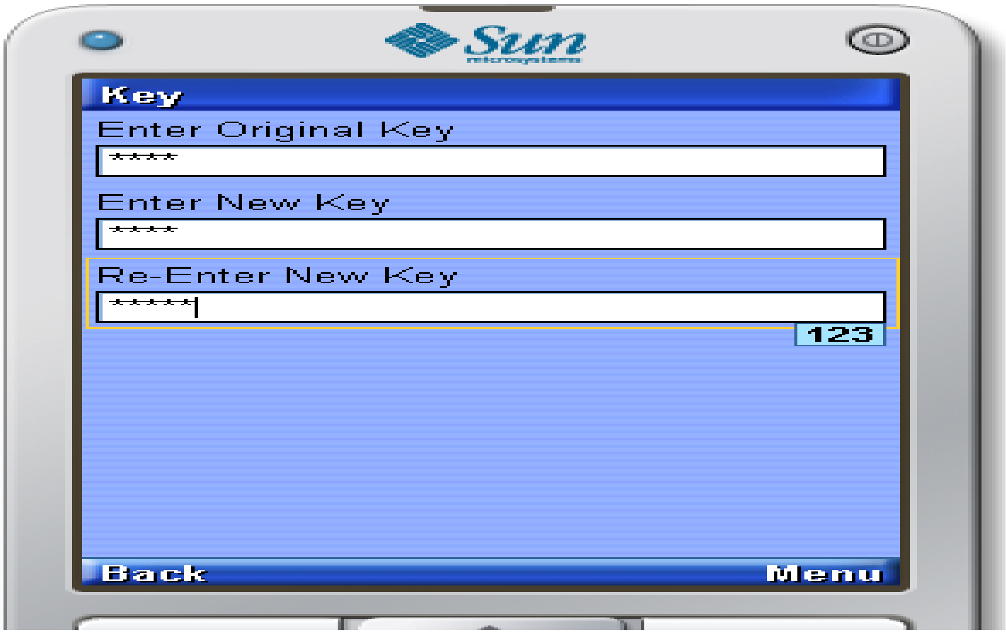


Figure 15: change draft keys

step 8: Changing Keys



Key

Enter Original Key

*****

Enter New Key

*****

Re-Enter New Key

******

Figure 16: Successful change

Appendix iii: SOURCE CODES

Encryption codes

```
class sealer {

static int getIndex(char c){

         int i=0;

         if(!isCapital(c)){

          for(char a='a';a<='z';a++){

               if((c+"").trim().equals((a+"").trim()))

                     break;

                   i++;

          }

         }

         else

            if(isCapital(c)){

               for(char a='A';a<='Z';a++){

               if((c+"").trim().equals((a+"").trim()))

                     break;

                   i++;       }
          }
        return i;
    }
static boolean isCapital(char c){
    int i=0;
    for(char a='A';a<='Z';a++){
              if((c+"").trim().equals((a+"").trim())){
                  return true;
              }    i++;
          }
    return false;
}
//****************************************A method to convert string of message
to an array of characters****************
static char[] toChars(String a){
    char chars[]=new char[a.length()];
    for(int i=0;i<a.length();i++){
        chars[i]=a.charAt(i);
    }
    return chars;
}
```

```
static char moveCharBy(char a,int by){
    char c='a';
    int index=getIndex(a);
    int newIndex=(index+by)>25?(index+by)-26:((index+by)<0)?
(index+by)+26:index+by;
    if(!isCapital(a))
        c=("abcdefghijklmnopqrstuvwxyz").charAt(newIndex);
    if(isCapital(a))
        c= ("ABCDEFGHIJKLMNOPQRSTUVWXYZ").charAt(newIndex);
    return c;
 }
static boolean isAlphabetic(char c){
    if(("abcdefghijklmnopqrstuvwxyz").indexOf(c+"")!=-1||
("ABCDEFGHIJKLMNOPQRSTUVWXYZ").indexOf(c+"")!=-1)
          return true;
          return false;
}
static String encript(String s,int key){
    String en="";
    for(int a=0;a<s.length();a++){
        en=en+((isAlphabetic(s.charAt(a)))?                    (moveCharBy(s.charAt(a),key)+""
):s.charAt(a)+"");

    }

    return en;

}

//***********************************************Advanced
encryption************************************************

public static  String encryptMore(String x,String key){

    int keyLength=key.trim().length();

    String encrypted=x;

    for(int i=0;i<keyLength;i++){

        boolean is=("0123456789".indexOf(key.charAt(i))!=-1)? true:false;


encrypted=encript(encrypted,is?Integer.parseInt(key.charAt(i)+""):getIndex(key.charAt(i
)));

    }

    return encrypted;

}

//***********************************************Advanced
dencryption************************************************

public static  String dencryptMore(String x,String key){

    int keyLength=key.trim().length();

    String encrypted=x;
    for(int i=0;i<keyLength;i++){

        boolean is=("0123456789".indexOf(key.charAt(i))!=-1)? true:false;

        encrypted=encript(encrypted,is?0-Integer.parseInt(key.charAt(i)+""):0-
getIndex(key.charAt(i)));

    }

    return encrypted;

}

public static String complicateMore(String x){

    String complicated="";

    boolean isNumeric=false;

    boolean isSmallChar=false;

    boolean isCapitalChar=false;

    for(int y=0;y<x.length();y++){

       isNumeric="0123456789".indexOf(x.charAt(y))!=-1;

        isSmallChar="abcdefghijklmnopqrstuvwxyz".indexOf(x.charAt(y))!=-1;


isCapitalChar="ABCDEFGHIJKLMNOPQRSTUVWXYZ".indexOf(x.indexOf(x))!=-1;

       if(isNumeric){

           complicated=complicated+(char)((int)(x.charAt(y))+113);

           continue;

         }

        if(isSmallChar){
            complicated=complicated+(char)((int)(x.charAt(y))+102);
            continue;
          }
        if(isCapitalChar){
            complicated=complicated+(char)((int)(x.charAt(y))+107);
            continue;
          }
        complicated=complicated+x.charAt(y);
    }
    return complicated;
}
public static String simplify(String x){
    String complicated="";
    boolean isNumeric=false;
    boolean isSmallChar=false;
    boolean isCapitalChar=false;
    for(int y=0;y<x.length();y++){
        isNumeric="¡¢£¤¥¦§¨©ª".indexOf(x.charAt(y))!=-1;
        isSmallChar="ÇÈÉÊËÌÍÎÏÐÑÒÓÔÕÖ×ØÙÚÛÜÝÞßàá".indexOf(x.charAt(y))!=-1;
       isCapitalChar="¬-®¯°±²³´µ¶·¸¹º»¼½¾¿ÀÁÂÃÄÅÆ".indexOf(x.indexOf(x))!=-1;
       if(isNumeric){
```

```
      complicated=complicated+(char)((int)(x.charAt(y))-113);
      continue;
     }
    if(isSmallChar){
      complicated=complicated+(char)((int)(x.charAt(y))-102);
      continue;
     }
    if(isCapitalChar){
      complicated=complicated+(char)((int)(x.charAt(y))-107);
      continue;       }
    complicated=complicated+x.charAt(y);
  }
  return complicated;
}
public static void main (String args[]){
  System.out.println(simplify("FÏÔÇÒ ÍÇÓËÙ ÊÕÔË ÚÕÓÕØØÕÝ"));}
}
```

Appendix iv: SOFTWARE TEST PLAN

Secure Messaging Application

21/12/2012

Prepared for:

School of Computing and Informatics

University Of Nairobi

Prepared by:

Veronicah Mutua

RECORD OF REVISIONS

Rev	Result of	Pages Affected	Approval/Date
1	Secure messaging version 1.0	Initial Release Testing	21/12/2012

1.0 Introduction

Testing of Mobile Applications is an emerging research area that faces a variety of challenges due to unique features of mobile devices, limited bandwidth, unreliability of wireless networks, as well as changing context. This is a rapidly emerging domain and hence got significance in the world of Mobile Software Testing and development also. This document explains the testing methodology for Secret Messaging mobile application, and is to be used as a guide for the testing activity.

1.1 Scope

The scope of testing as explained in the document is to test the operating characteristics of Secret Messaging application that runs on mobile devices supporting J2ME. The tests are organized by requirement category such as usability, functionality, security, etc. The procedure for carrying out testing in terms of preparation of test cases, test environment setup, defects logging and reporting are explained. The document does not address the following:

- Content censorship (i.e. assessment against standards for violence, gambling, political messaging etc.) for the purpose of preventing the deployment or sale of an application. Distribution, DRM etc.
- Testing requirements specific to a particular manufacturer's (or network operator's) device, user interface, and standards (e.g. WAP) implementation.

2.0 Test Plan and Strategy

2.1 Unit Testing

2.1.1 **Objective**

The objective of Unit testing will be carried out to verify that particular module of source code is working properly. The class sealer will be tested in the Unit tests to verify its ability to encrypt and decrypt messages. Unit test for all other components for messaging will also be carried out.

2.1.2 Entry Criteria

- Test cases is reviewed
- Build is complete and self-test done
- Unit Test environment is set up

2.1.3 Exit Criteria

- All planned test cases are executed
- Units are working as per the expected results
- Defect are fixed in the code and tracked to closure

2.1.4 Logging Tests and Reporting

The developer will fix the defects that are found in unit testing. Additionally, if defects corresponding to other modules or components are found during unit testing, these will be reported.

2.2 System Testing

In System Testing, separate units (packages / modules / components), or groups of units of the application will be united and tested as a completely merged mobile application. The purpose of System Testing is to identify defects that will only surface when a complete system is assembled. Verification of the system at this stage will include: functionality, usability, security, installation etc. It is intended to validate the application as a whole.

2.2.1 Testing Procedure

The steps in testing will consist of:

- Creation of all the test scenarios and test cases
- Preparation of a test case document that has a brief description of the test case , steps to conduct tests and expected result
- Defect Report generation.

2.3 Test Strategy

The main test types that will be performed are

Application Characteristics (AC) – Information about the application is provided to help the testing team in the testing work.

Stability (ST) – Focusing on the application being stable on the device.

Application Launch (AL) – Once an application is loaded it must start (launch) and stop correctly in relation to the device and other applications on the device.

User Interface (UI) -This will include all the screens and their user friendliness, the messaging menus and the help buttons for the new users.

Functionality (FN) - Documented features in the proposal shall be tested to find out if they are implemented in the application and they work as expected. The functionalities will include the application's ability to encrypt text messages, decipher inbox messages, storage of outbox messages, encrypting the Draft box and key generation module.

Connectivity (CO) –this test will be carried out to ensure that the application demonstrates its ability to communicate over a network correctly. This will include testing its capability in dealing with both network problems and server-side problems.

Personal Information Management (PI) - The application accessing user information needs to be able to do it in an appropriate manner and not to destroy the information.

Security – this test will be carried out using *SmartSniff* hacking tool to ensure that the encrypted messages remain un-altered.

2.3.1 Test Metrics

Following metrics will be captured and reported as part of the Test

- Summary Report
- Test Design effort
- Test execution effort
- Number of Test Cases executed
- Number of Defects and their classification
- Test Coverage (Number of test cases executed/Number planned)

4.0 Risks and Assumptions

4.1 Risks

The following may impact the test cycle:

- Device availability
- Any new feature addition/modification to the application which is not communicated in advance.
- Any delay in the software delivery schedule including defect fixes. Any changes in the functional requirements since the requirements were signed-off/formulated

4.2 Assumptions

- Every release to QA will accompany a release note specifying details of the features implemented and its impact on the module under test.
- All “Show-Stopper” bugs receive immediate attention from the development team.
- All bugs found in a version of the software will be fixed and unit tested by the development team before the next version is released
- All documentation will be up-to-date and delivered to the system test team.

- Devices, Emulators and other support tools will be fully functional prior to project commencement.
- In case of lack of required equipment or changes in the feature requirements, the test schedules may need to be reviewed.

5.0 Roles and the responsibilities

5.1 Supervisor in charge

- Approval of the test documents
- Approval of inspections/reviews done as per the test plan
- Providing resources and guidance for the project

5.2 Lead Tester/student developer

- Requirement gathering
- Planning and estimating for testing
- Tracking and monitoring the testing as per the test plan
- Reporting the project status
- Creating the test cases as per the test plan
- Executing the test cases
- Documenting the results and logging errors.

6.0 Schedules for Testing

The test will be carried from the commencement of coding on 4th November 2012 to January 3rd 2013. Specific tests will be determined with the project supervisor.