JARAMOGI OGINGA ODINGA UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF INFORMATICS AND INNOVATIVE SYSTEMS

PROJECT TITLE: MESSAGE ENCRYPTION APPLICATION FOR ANDROID.

A project submitted in Partial Fulfillment of the Requirements for the award of the

DEGREE OF

BACHELOR OF SCIENCE

In

COMPUTER SECURITY AND FORENSICS

Authors:

JOHN MWANGI NJUGUNA **-**I132/**0865**/2013

LILIAN MWENDE MUTHUI **-**I132/**0870**/2013

Under the guidance of

JOSHUA AGOLA

DECLARATION

This project, presented on this report is our original work and to the best of our knowledge has not been presented for any degree or diploma examination in any other university or collage.

STUDENT'S NAME : LILIAN MWENDE MUTHUI

REGISTRATION NO: I132/**0870**/2013.

SIGNATURE: ………………………………………………………………

DATE: …………………………………………………………………….

STUDENT'S NAME : JOHN MWANGI NJUGUNA.

REGISTRATION NO: I132/**0865**/2013.

SIGNATURE: ………………………………………………………………

DATE: …………………………………………………………………….

SUPERVISOR'S NAME : JOSHUA AGOLA

SIGNATURE ……………………………………………….

DATE ………..………………………………………

DEDICATION.

Dedication of this entire work carried herein goes to our Parents for their inspiration all through our course. We would like to appreciate all our teachers who instilled in us lifelong values and the desire for education. Finally we would like to appreciate all those who have contributed to our success and have not been mentioned above, all your efforts are highly appreciated and you will never be forgotten for your stake in our life, God bless you all.

ACKNOWLEMENT.
We consider ourselves exceptionally fortunate that we had indulged guides, from our supervisor Joshua Agola and caring friends to successfully steer us through one of the most challenging assignment of our academic career. Today when our Endeavour has reached its fruition, we look back in mute gratitude to one and all without whose help we are sure; this reality would have remained a dream.

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 worlds 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 is to develop a secure messaging platform for android phones to reduce these vulnerabilities and loopholes. To implement the Advanced Encryption Standard (AES) 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, android phones 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, GSM, mobile applications, AES.

Table of Contents

DECLARATION ii

DEDICATION. iii

ACKNOWLEMENT. iv

ABSTRACT v

LIST OF TERMS AND ABBREVIATIONS ix

CHAPTER ONE. 1

1.0 INTRODUCTION. 1

1.1 BACKGROUND INFORMATION. 1

1.2 PROBLEM STATEMENT. 2

1.3 OBJECTIVES. 2

1.3.1General objective: 2

1.3.2 Specific objectives: 2

1.4 JUSTIFICATION. 3

1.5 SCOPE 3

1.6 ASSUMPTIONS 3

CHAPTER TWO. 4

2.0 LITERATURE REVIEW. 4

2.1 EXISTING SOFTWARE 5

2.2 CONCEPTUAL FRAMEWORK 7

CHAPTER THREE 8

3.0 METHODOLOGY 8

3.1 Introduction 8

3.2 Development methodology 8

3.3 Planning 9

3.4 Analysis 9

3.5 Design 9

3.6 Coding and Implementation 9

3.7 Testing 9

3.8 Deployment 10

3.9 Reason for using Waterfall Model 10

CHAPTER FOUR 11

4.0 SOFTWARE DEVELOPMENT 11

4.1. ANALYSIS AND REQUIREMENTS 11
4.2 DESIGN. 15
4.3 IMPLEMENTATION 18
4.4 TESTING AND VALIDATION 24
CHAPTER FIVE 26
5.0 DISCUSSION 26
5.1 Introduction 26
5.2 Achievement of Objectives 26
5.3 Discussion of Results in Relation to Specific objectives 26
5.4 Challenges 26
CHAPTER SIX 27
6.0 CONCLUSION 27
REFERENCES. 28
Appendix I: Cost Estimation 30
Appendix II: Project Schedule 30
Appendix III: User Manual 31
Appendix IV: SOURCE CODES 36

LIST OF FIGURES

Figure 1: Conceptual Framework 7
Figure 2: Waterfall Model 8
Figure 3: Technical design model 15
Figure 4: User Use case model 16
Figure 5: Interaction diagram 17
Figure 6: Graphical user interface 18
Figure 7: EncDecSMS Codes 19
Figure 8: Decrypted SMS 20
Figure 9: Encrypt Activity 21
Figure 10:Recever and sender field content 22
Figure 11: Display activity 22
Figure 12: R. java file 23
Figure 13: Messaging sender end 31
Figure 14: Messaging platform 32
Figure 15: Field for composing SMS 33
Figure 16: Receiver decryption interface 34
Figure 17: Decrypted text 35

LIST OF TABLES.

Table 1: Cost Estimation 30
Table 2: Project Schedule 30

LIST OF TERMS AND ABBREVIATIONS

GSM- (Global System for Mobile Communications) - it is an open, digital cellular technology used for transmitting mobile voice and data services.

Plaintext: This is the original intelligible message or data that is fed into the algorithm as input.

Cipher text: This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different cipher texts. The cipher text is an apparently random stream of data and, as it stands, is unintelligible.

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

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

Encryption algorithm: The encryption algorithm performs various substitutions and transformations on the plaintext.

Decryption algorithm: This is essentially the encryption algorithm run in reverse. It takes the cipher text and the secret key and produces the original plaintext.

Symmetric encryption -transforms plaintext into cipher text using a secret key and an encryption algorithm. Using the same key and a decryption algorithm, the plaintext is recovered from the cipher text.

Secret key: The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key.

AES- Advanced Encryption Standard

RIM-Record information management system

SMS-Short Messaging Services

CHAPTER ONE.

1.0 INTRODUCTION.

Message Encryption is getting more popular now-a-days. SMS was first used in December 1992, when Neil Pap worth, a 22-year-old test engineer used a personal computer to send the text message “Merry Christmas” via the Vodafone GSM network to the phone of Richard Jarvis in the UK. Presently many business organizations use SMS for their business purposes. SMS’s security has become a major concern for business organizations and customers. There is a need for an end to end SMS Encryption in order to provide a secure medium for communication. Security is main concern for any business company such as banks who will provide these mobile banking services. Till now there is no such scheme that provides complete SMSs security. The transmission of an SMS in GSM network is not secure at all. Therefore it is desirable to secure SMS for business purposes by additional encryption. SMS Message Service (SMS) is a textual form of communication which is of precise length. SMS’s are very much in use. So it is must to secure SMS’s.

1.1 BACKGROUND INFORMATION.

Short message service (SMS) is most important communication medium in many daily life applications, including healthcare monitoring, mobile banking, mobile commerce, and so on. When an SMS is sent from one mobile phone (MS) to another MS (Mobile subscriber), the information contained in the SMS to be transmitted is in the form of a plain text. Sometimes this information may be confidential like account numbers, passwords, license numbers, and so on, and it is a major drawback to send such information through SMS while the traditional SMS service does not provide encryption to the information before its transmission, However telecom service providers are ensuring at server end some security provided as Using A3, A8 and Kc algorithms, but not providing during the message transmission.

Many times when data is exchanged electronically the privacy of the data is a requirement. The use of encryption restricts unintended recipients from viewing the data, which are deemed confidential and potentially dangerous if made known to irresponsible parties. Since there is no security for the messages, it indirectly leads to a lot of problems where important and confidential information such as passwords is being accessed by unauthorized individual. Apart

from all that, there are also some other cases like the mobile phone owner accidently send messages to the wrong number and it gets worse when the mobile get stolen.
Encryption is the procedure of transforming plaintext, data that can be read by anyone, to cipher text, data that can only be read by someone with a secret decryption key. A message before being changed in any way is called plaintext. Plaintext messages are converted to cipher text via some encryption method. A particular such method is called a cryptosystem.

1.2 PROBLEM STATEMENT.

Confidential information or instructions can only be safely passed on to the required parties by email or verbally in person. This delays time and efficiency of crucial business operations. Currently messages in phone can easily be read if the phone is stolen, or users might simply end and receive private information. At present, banking, commerce and passwords can only be communicated with internet connection or as in person verbally or in written. SMS helps to overcome this intermediate subjects and deliver instructions or messages instantaneously. Although there is no security for message transit over hand phone offered by various mobile operators. SMS usage is threatened with security concerns, such as SMS disclosure, man-in-the-middle attack, replay attack and Impersonation attack. In order to protect confidential information, it is strongly required to provide end to end secure communication between end users.

The purpose of this project is to develop an android message encryption application that will encrypt and decrypt the messages which are sent through mobile devices using GSM connection.

1.3 OBJECTIVES.

1.3.1General objective:

The main objective of the project is to develop an android application that will be used to encrypt SMS during the transmission by ensuring end to end secure communication. This will help to ensure the confidentiality, integrity and availability of sensitive information contained in the text message.

1.3.2 Specific objectives:

1. To develop a secure SMS application.

2. To Protect against misuse of message information.

3. To ensure high confidentiality and improved security.

1.4 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 because only the users with the right keys will be able to access the messages. Secondly, the messages remains 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.

1.5 SCOPE

The application is built on Android platform. Therefore, it can be used on any device which runs on Android operating system. This application can be used in industries for secured data transfer. Apart from commercial and business use, this application can be used for non-commercial and personal use. The purpose of this application is to send secured data between two devices.

1.6 ASSUMPTIONS

- 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.

CHAPTER TWO.

2.0 LITERATURE REVIEW.

Lisonek et al, (2008) in Burnaby proposed an algorithm to send message through GSM using an asymmetric Rivest, Shamir and Adleman (RSA) cipher. This application prevents taping and substituting techniques to secure SMS. It is achieved by storing the public key in a certificate which can be signed by the certification authority.

The study only focused on securing the keys rather than the content of the SMS which is our main concern.

Rupa et al, (2009) in India proposed a cost effective scheme which uses a concept called Cheating Text. The original message is embedded in a meaningful text cheating text. Here, index table called (Real Message Index File) RIF file is hashed and sent to the receiver along with the cheating text in which the original message is embedded. Authentication is achieved by verifying the hash value of the plain text.

The study lacks a mechanism for ensuring SMS availability and integrity and these are the core components in our project.

Agoyi et al, (2010) in Cyprus evaluated encryption and decryption time for three algorithms RSA, Elliptic-curve and EIGamal to which plain text of different sizes is provided based on results one is chosen for further encryption. Their performance evaluation in securing SMS shows that key generation, encryption and decryption time increases with an increase in the key size: Large key size algorithms are not suitable for SMS encryption due to small memory and low computational power of mobile phones.

The study focused on evaluating how the keys should be generated and not a secure means of sharing the keys.

Ding (2012) in USA did improvements on RSA algorithm because the SMS will filter out the characters which are out of prescribed limit, thus the cipher text can’t reach the destination.

Thus, they also used FPGA based on high speed processing tools to implement the RSA algorithms and apply it in mobile phone short message encryption system.

The study focused on the use of RSA Algorithm that filters certain characters in SMS while our study focuses on AES algorithm for encrypting the SMS without limiting the characters.

Madhwani (2012) in India performed a work, "Cryptography on Android Message Application Using Look up Table and Dynamic Key (Cama)". The algorithm proposed for this cryptography is based on static Look Up table and Dynamic Key. Symmetric encryption and decryption is used in this algorithm. The proposed algorithm is more secure and simple to implement. This application makes use of built in android Intents and SMS Manager to send and receive messages.

This study is concerned with encryption of SMS but it does not focus on security during SMS transmission which our encryption application is aiming to achieve.

2.1 EXISTING SOFTWARE

Wits SMS Encryption is professional text messages encryption software for Windows Mobile phones. It can encrypt and hide the SMS messages (including existing SMS messages and future incoming/outgoing SMS messages) on the windows mobile devices. The hidden SMS messages are encrypted and will be stored separately. You can only decrypt and show your messages with a correct password. That means, others cannot view any of your secret or important SMS messages on your mobile phone without your approval, even if you lost your phone or lend your mobile phone to your friend or relative. Your privacy will be safe

What Sapp for instance has currently introduced the end to end encryption of messages. What Sapp changed the end to end encryption game in April 2016, when it added end to end encryption for all its users, on every platform? What Sapp is used by 1 billion people on android, iPhone, blackberry, windows phone and noni phones and it has a web-based version so you can use the service on windows, mac and Linux. There is a good chance that your friends already use it, and now it protects communications on just about every platform available, including text, group chat, photo, and video, file and voice message.

Other encryption applications are **silent phone** founded by Phil Zimmermann, inventor of the widely known but hard to use email encryption tool pretty Good Privacy, Silent Circle also makes the Black phone, a security and a privacy focused Android phone that comes with its suite of encrypted apps preinstalled.

Wicker is also another messaging only service that lets you set messages to self-destruct after they have been read. It works a bit like snap chat, except that it includes end to end encrypted communication with other people who have installed and are using the app on an Android, is windows mac or Linux device.

2.2 CONCEPTUAL FRAMEWORK

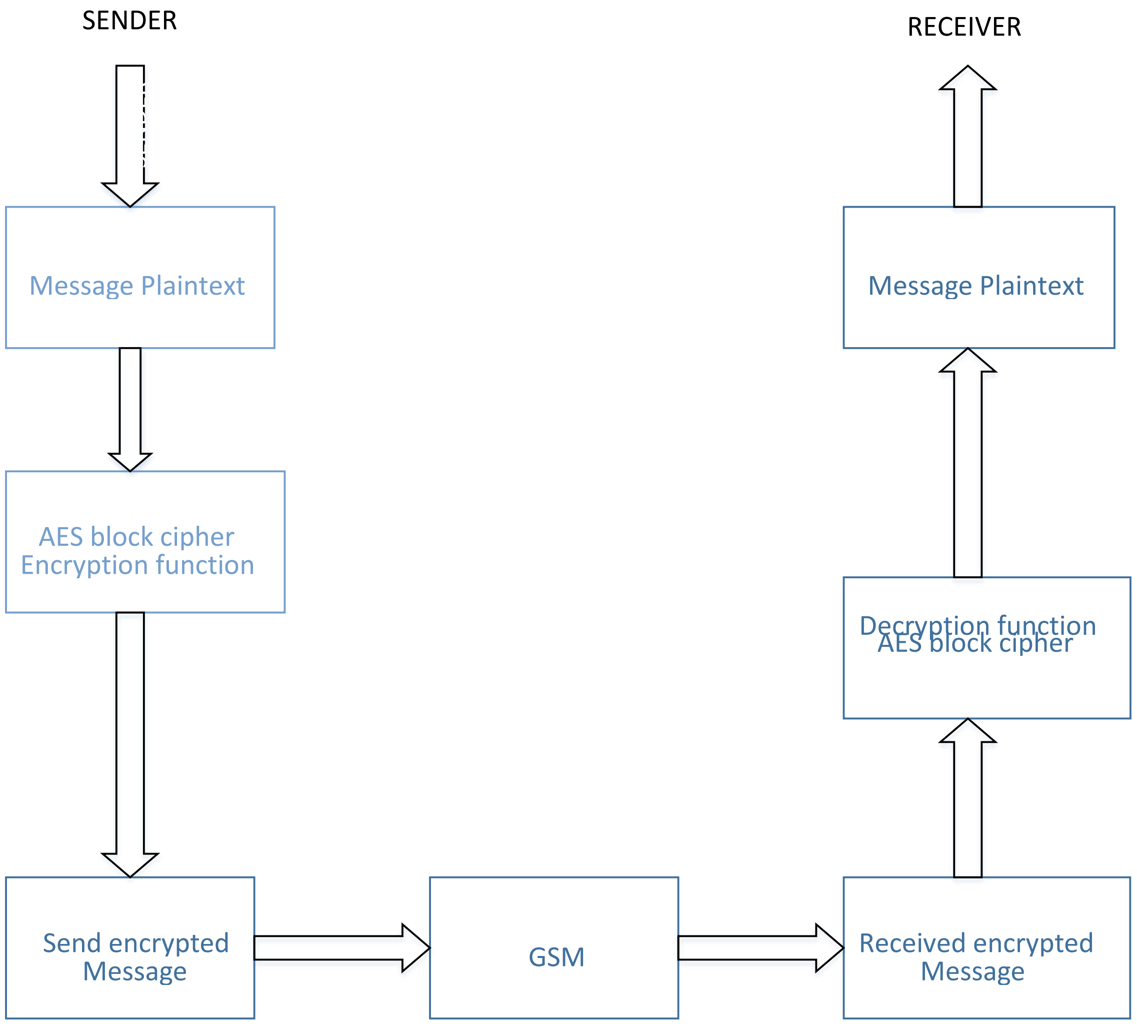

SENDER
RECEIVER
Message Plaintext
Message Plaintext
AES block cipher Encryption function
Decryption function AES block cipher
Send encrypted Message
GSM
Received encrypted Message

Figure 1: Conceptual Framework

CHAPTER THREE

3.0 METHODOLOGY

3.1 Introduction

The project methodology that we will use in the development of the android application is the System Development Life Cycle (SDLC). The SDLC is the process of understanding how information system can be valid to the user needs, then designing the system, building it and delivering it to the potential users. This methodology is composed of some phases. The structured design methodology will be waterfall development.

3.2 Development methodology

The Waterfall Model was first Process Model to be introduced. It is also referred to as a **linear-sequential life cycle model**. It is very simple to understand and use. In a waterfall model, each phase must be completed fully before the next phase can begin. This type of model is basically used for the project which is small and there are no uncertain requirements. At the end of each phase, a review takes place to determine if the project is on the right path and whether or not to continue or discard the project. In this model the testing starts only after the development is complete

In the course of developing the system we chose to use the Waterfall Development Model as illustrated in the figure below.

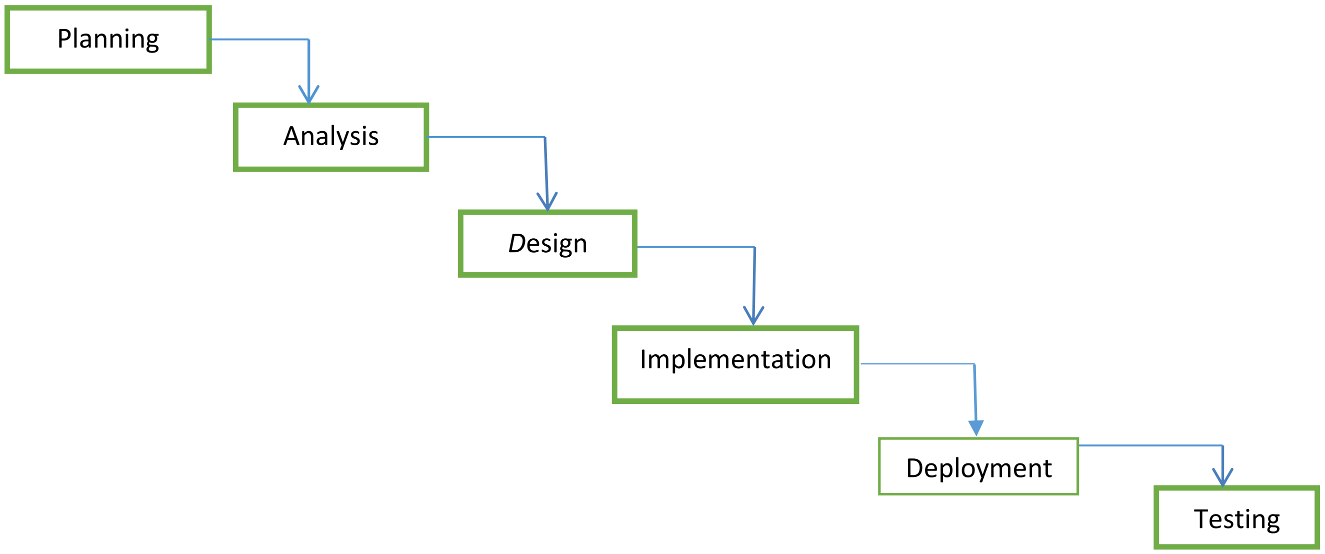


Figure 2: Waterfall Model

3.3 Planning

In planning phase which was first step was to identify needs for the android message encryption application. This was the first phase of the system and it entailed determining the necessary information that was required for the system to be fully operational as well as function in the expected manner. The requirements that we captured was subjected to thorough scrutiny to determine the level of essence of it as well as eliminate the unwanted requirements.

3.4 Analysis

In this phase, we analyzed and considered the current applications and investigated any problems associated with it. Other sources of information about the application and the new requirements were also investigated at this time. The important information from the planning phase was highly used in this phase, and the valid information gathered from the users was analyzed for the design stage.

3.5 Design

After the requirements having already been captured and analyzed, the design of the information flow was done here. It is in this phase that the technical design, use case diagrams and interaction diagrams were drawn to show flow of information.

3.6 Coding and Implementation

After the design of the interfaces as well as the indication of the information flow through the use case diagrams and the interaction diagrams, the next step was to develop code that performed the hidden functionality of the application to realize the already set objectives. The code was developed highly depending on the information flow. The requirements documentation were referred throughout the rest of the application development process to ensure the developing project aligns with the needs and requirements or scope. A proper execution of the previous stages ensured an easier realization of this phase in the course of our development. Upon completion of the coding, the various components of the application were then integrated into one system in order to function collectively as a single component.

3.7 Testing

Last phase is application testing done when development is complete and the application is ready for deployment. The testing phase come next to determine if the earlier intended objective have been realized by then. Testing was done based on whether completeness will have been realized or functional testing that determined whether the software is doing what it is expected correctly and in the right way. User testing was then carried out to ascertain that the users will be contented with what will have been achieved then.

3.8 Deployment

Once the application has undergone testing and passed all the validations and verifications, then the application would be termed to be acceptable and therefore it will be handed over to the users for maintenance and operation as well.

3.9 Reason for using Waterfall Model

a) Simple and easy to understand and use.
b) Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
c) Phases are processed and completed one at a time.
d) Works well for smaller projects where requirements are very well understood.
e) Clearly defined stages
f) Well understood milestones.
g) Easy to arrange tasks.
h) Process and results are well documented.

CHAPTER FOUR

4.0 SOFTWARE DEVELOPMENT

4.1. ANALYSIS AND REQUIREMENTS

4.1.1. Feasibility Study:

Feasibility Study is conducted to see if the proposed application is a feasible one with all respects. There are three main aspects in the feasibility study. The feasibility of a project can be ascertained in terms of technical factors, economic factors, or both. A feasibility study is documented with a report showing all the ramifications of the project. In project finance, the pre-financing work is to make sure there is no "dry rot" in the project and to identify project risks ensuring they can be mitigated and managed in addition to ascertaining "debt service" capability.

4.1.2 Economic Feasibility:

In economic feasibility cost/benefit analysis is done. Here we determine the benefits and time savings that are expected from the system and compare them with cost. The proposed application is economically feasible. We developed java package having classes and methods for AES encryption and decryption so it will be very easy and less costly to implement. It is economically feasible.

4.1.3 Operational Feasibility:

An operationally feasible application is one that will be used effectively after it has been developed. If users have difficulty with a new system, it will not produce the expected benefits. The proposed application is found to be operationally feasible because of the following reasons. It is very simple to use. There is no difficulty in using the front end which has been developed. Even the users who don’t have any knowledge in android mobile the user friendliness and help section provides them convenience and case. The system is designed, in such a way that not only the person currently handling this work can operate the system but a person who is new to the system with case. Hence this application is found to be operationally feasible.

4.1.4 Technical Feasibility

Technical feasibility centers on the existing system and to the extent it can support the proposed application. This encryption package and application is built in java language so they are platform independent. This encryption can done on computer also using this package. Hence this system is found to be technical feasible.

4.1.5 Market Feasibility
Due to the increased need for information security this message encryption application is crucial. It uses less computing resources efficiently and do not compromise with security. So it is not costly and provides better security.

4.1.6 APPLICATION REQUIREMENTS AND SPECIFICATION.
4.1.6.1Hardware Requirements
a) Internet connection with a minimum speed of 100kbpS.

b) Computer with a minimum of following requirements: Processor-Intel(R) Pentium (P) Dual CPU, 32 bit operating system.

c) Android hand- set.

4.1.6.2 Software Requirements

a) Android SDK

A software development kit that enables developers to create applications for the Android platform. The Android SDK includes sample projects with source code, development tools, an emulator, and required libraries to build Android applications. Applications are written using the Java programming language and run on Dalvik a custom virtual machine designed for embedded use which runs on top of a Linux kernel. It enables application’s compatibility with android platform by means of API.

b) Eclipse

Eclipse is a software development environment comprising an integrated development environment (IDE) and an extensible plug-in system. It is written mostly in Java and can be used to develop applications in Java and, by means of various plug-ins, other programming languages. The IDE is often called Eclipse ADT (Android Development Toolkit) Eclipse JDT for Java.

c) Android Development Tools (ADT)

It is a plug-in for the Eclipse IDE that is designed to give you an integrated environment in which to build Android applications.

4.1.7 FUNCTIONAL REQUIREMENTS
Plain text- This is 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 and thus, that can be easily read from and understood.

Encryption key/ Decryption key- a piece of information (or parameter) that will control the execution of a cryptography algorithm. Encryption key will be used e for encryption (obtaining the cipher text out of the plaintext) and decryption key will be for decrypting (obtaining the plaintext out of the cipher text).

Cipher text- This is a sequence of characters (either letters from the alphabet, numbers or bytes of data) that is encrypted using an encryption algorithm.

Analysis of AES algorithm

Now there are so many encryption techniques available to convert a plain text message to Encrypted message. But the most popular technique is AES encryption technique that we have used here.

In this technique there are different length of encryption key might be used as per need.

But here we implemented this android application by using 128 bits. As 128 bits encryption key is used in this project so we need 16 digits to enter to encrypt the message (Per digit 8 bits). In this algorithm different transformations are being performed. There are total 4 transformations to be performed in total 10 rounds for each.

I. Substitute bytes Transformation: Uses an S-box to perform a byte-by-byte substitution of the block

II. ShiftRows Transformation: A simple permutation

III. MixColumns Transformation: A substitution that makes use of arithmetic over

IV. AddRoundKey Transformation: A simple bitwise XOR of the current block with a portion of the expanded key

The structure is quite simple. For both encryption and decryption, the cipher begins with an AddRoundKey stage, followed by nine rounds that each includes all four stages, followed by a Tenth round of three stages depicts the structure of a full encryption round.

Only the AddRoundKey stage makes use of the key. For this reason, the cipher begins and ends with an AddRoundKey stage. Any other stage, applied at the beginning or end, is reversible without knowledge of the key and so would add no security.

4.1.7 NON-FUNCTIONAL REQUIREMENT

4.1.7.1 Confidentiality requirement

Confidentiality means that the contents of a message sent over a certain channel cannot be read by other people. In general, this property can be broken down into two dimensions: (i) one must not be able to read the contents of the message and (ii) one must not be able to know the contents of a message without reading it.

4.1.7.2 Integrity requirement

While confidentiality ensures that the contents of a message are not exposed to others than the receiver, integrity ensures that the message as received by the customer has not changed during the transportation process.

4.1.7.3 Authentication requirement

When confidentiality and integrity are ensured, it is still possible that one sends a message pretending to be someone else. Therefore, a mechanism to verify the identity of the sender is of importance and this is achieved through encrypting text messages during transmission.

4.2 DESIGN.

4.2.1 TECHNICAL OF THE DESIGN

We have this GUI for sender part and receiver part of encryption and decryption of a SMS. Here in sender part, we have to insert recipient phone number in recipient number field. We have to insert a 16 bit secret key in Encryption key field. This secret key should be previously known to the recipient.

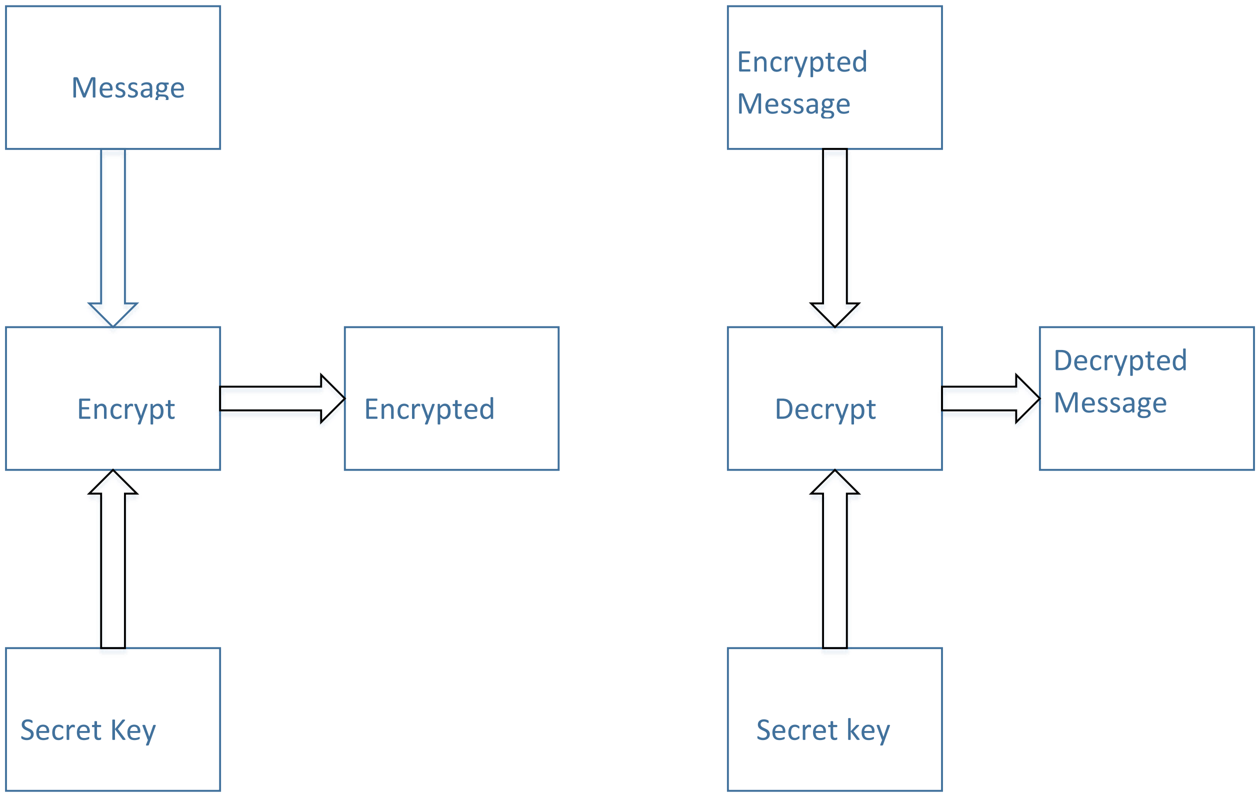


Figure 3: Technical design model

In receiver part there is a decryption key field, received encrypted message and decrypted message field. After inserting the same secret key original message can be seen.

4.2.2 Use case diagram

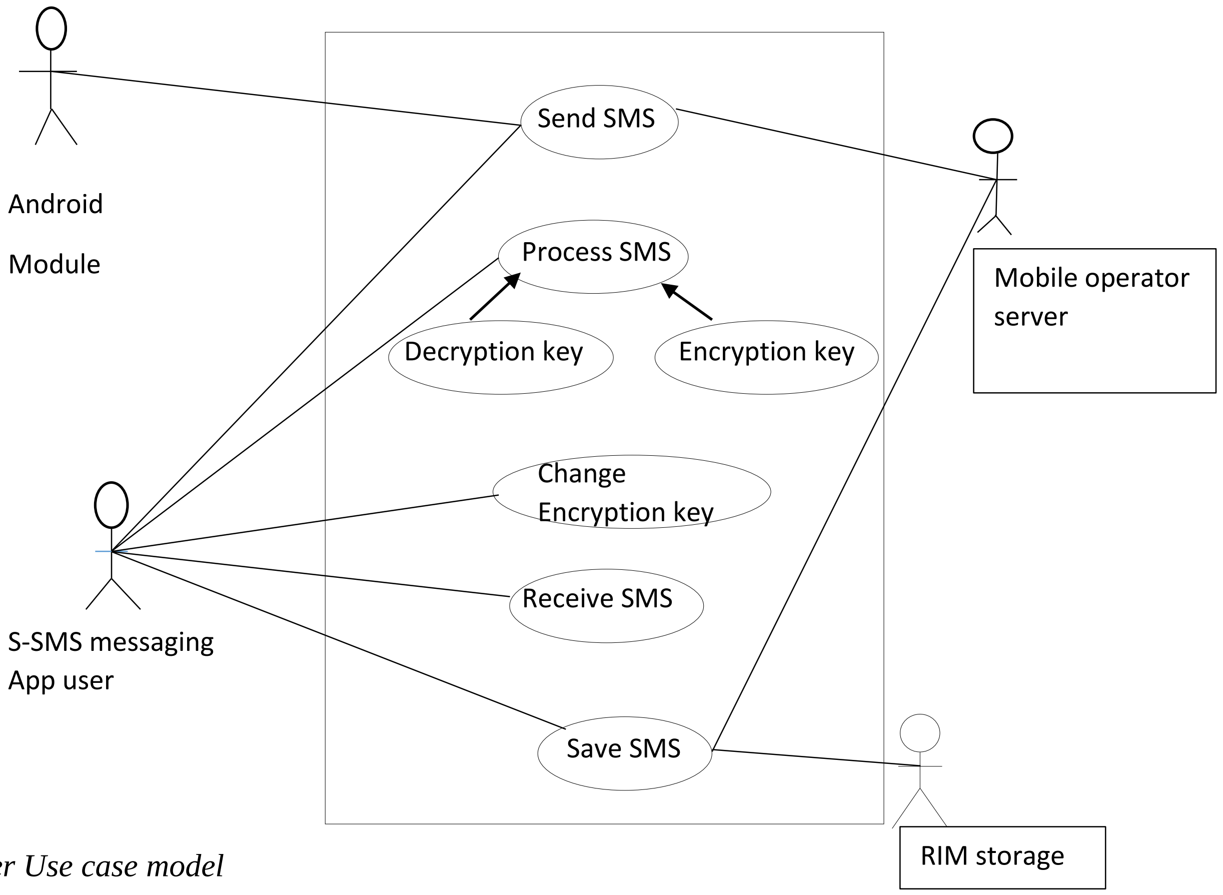


Figure 4: User Use case model

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

4.2.3 Level 1: Interaction diagram.

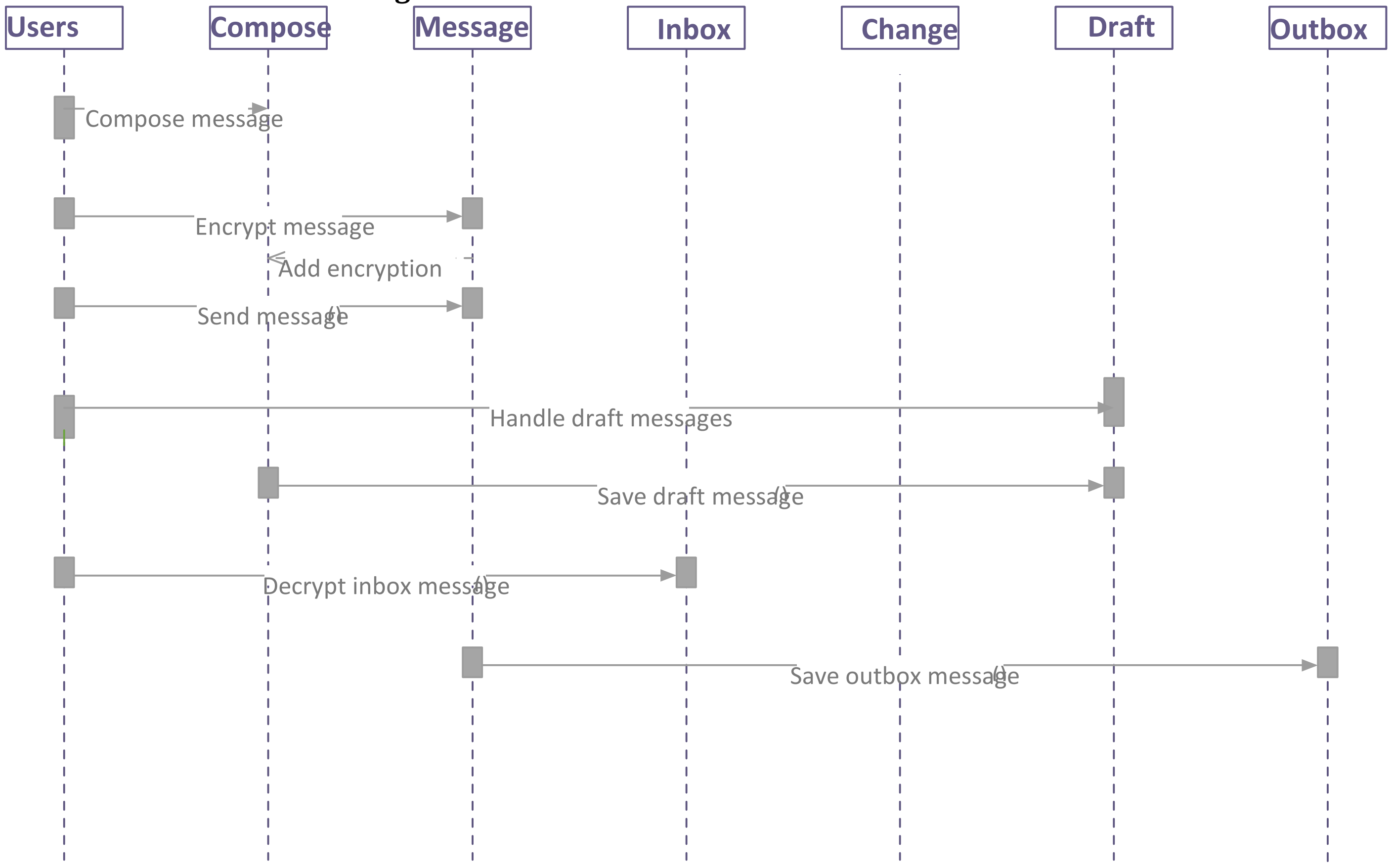


Figure 5: Interaction diagram

4.3 IMPLEMENTATION

4.3.1 GRAPHICAL USER INTERFACE & CODING IMPLEMENTATION

In our project, '**Sms Encryption using AES algorithm in android platform'** we have designed the interface of the sender as well as receiver. The GUI figure is given below:

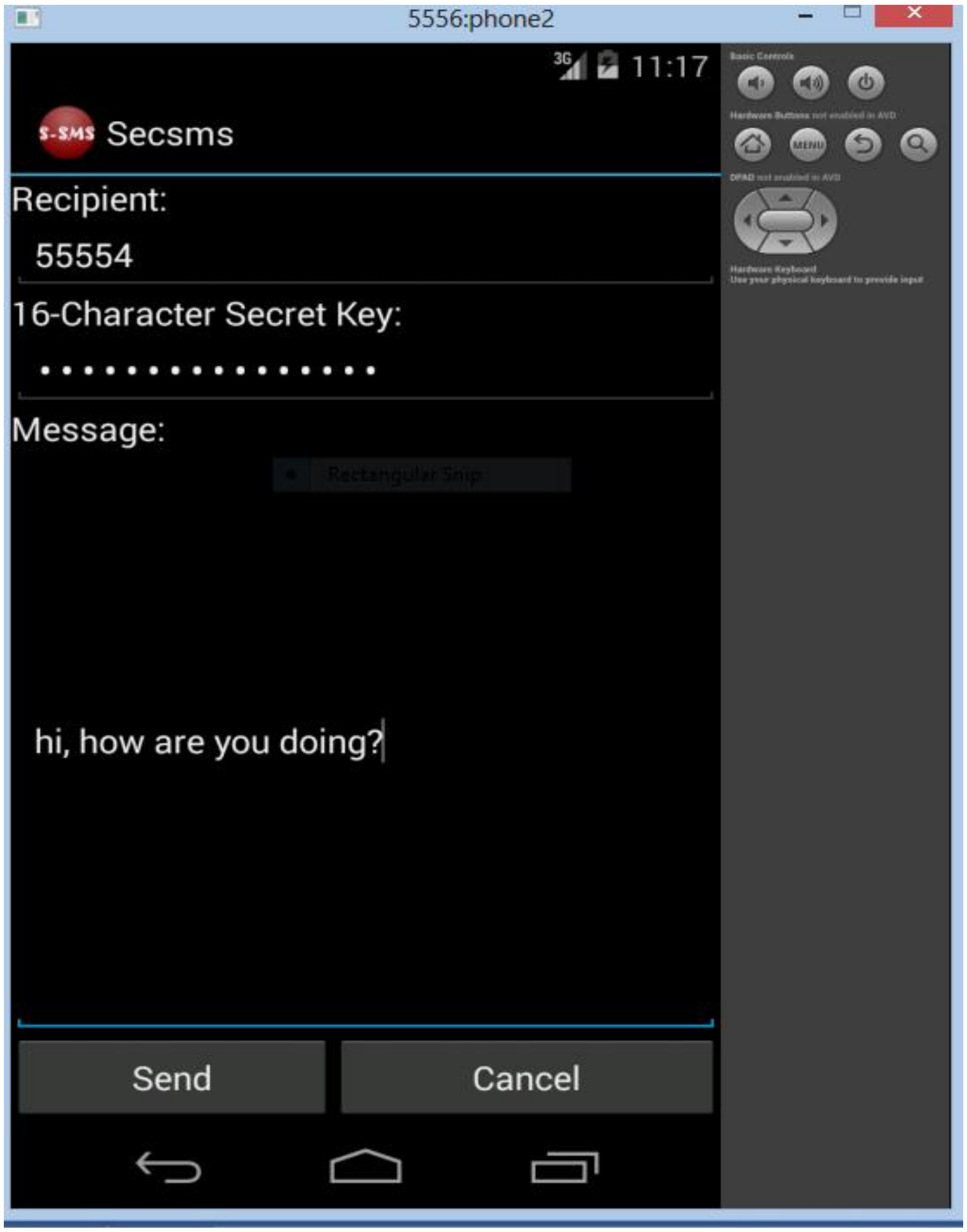
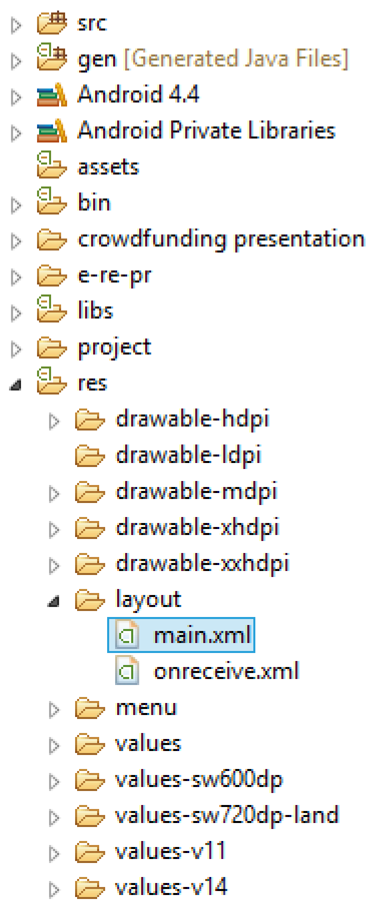


Figure 6: Graphical user interface

In the eclipse software we have started a new Android project name **EncDecSMS.**

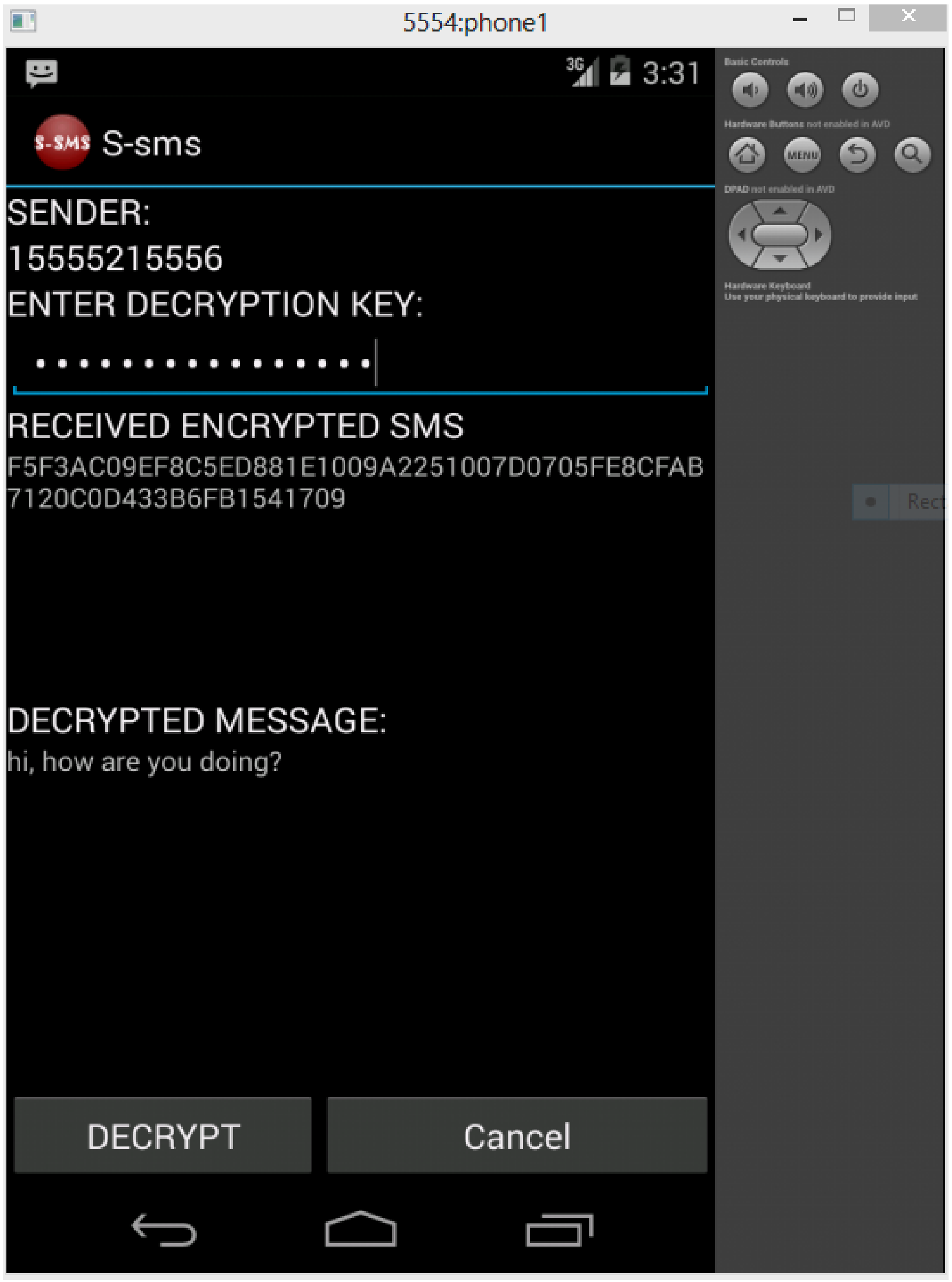
In the **res/layout** we created **Main.xml file** for creating sender layout.



```
▷ src
▷ gen [Generated Java Files]
▷ Android 4.4
▷ Android Private Libraries
  assets
▷ bin
▷ crowdfunding presentation
▷ e-re-pr
▷ libs
▷ project
◢ res
    ▷ drawable-hdpi
      drawable-ldpi
    ▷ drawable-mdpi
    ▷ drawable-xhdpi
    ▷ drawable-xxhdpi
    ◢ layout
          main.xml
          onreceive.xml
    ▷ menu
    ▷ values
    ▷ values-sw600dp
    ▷ values-sw720dp-land
    ▷ values-v11
    ▷ values-v14
```

Figure 7: EncDecSMS Codes

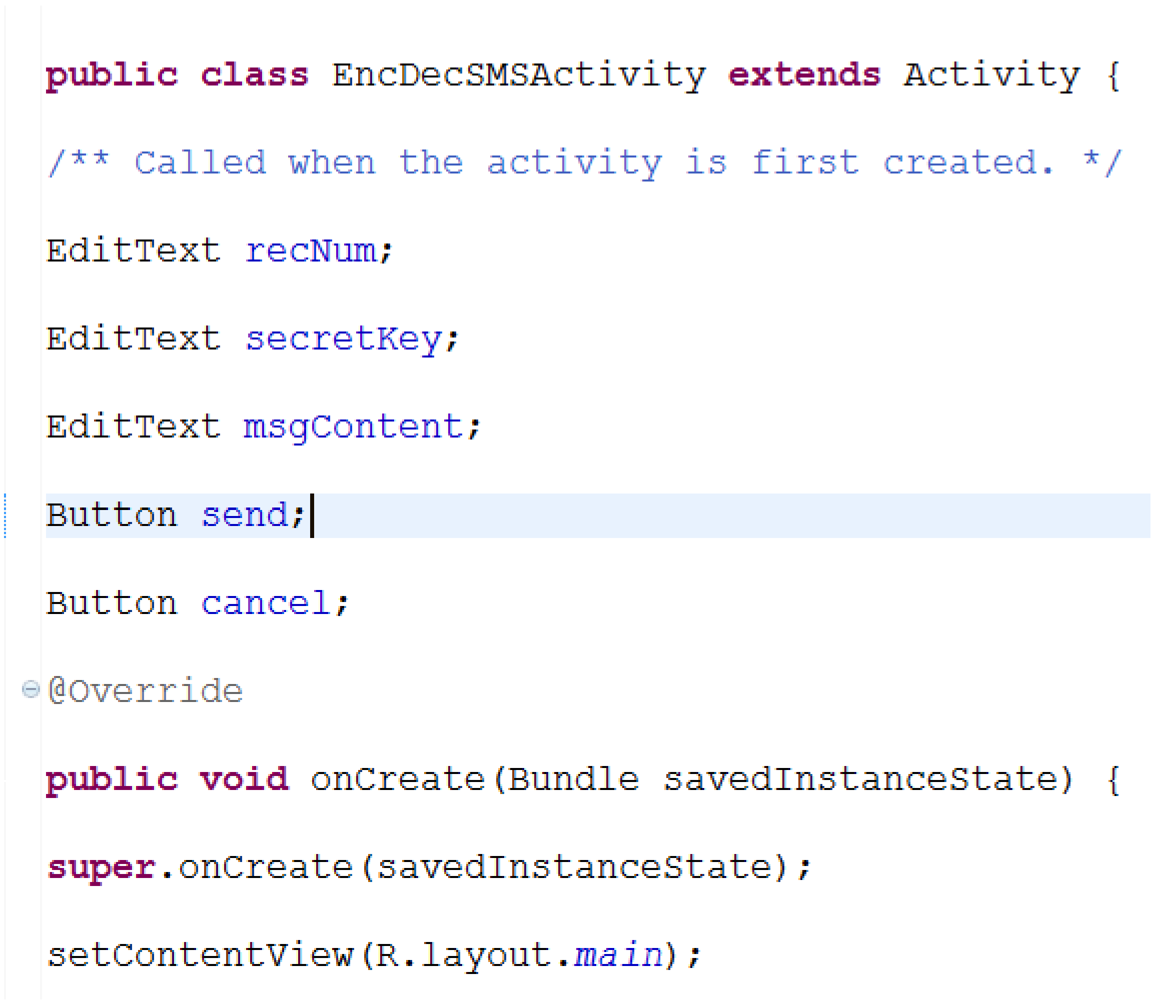
This GUI is responsible for receiver side. This GUI is designed under the same folder **Res/layout/onreceive.xml**. In this layout we are getting decrypted message by entering the secret key of 16 bits after pressing the Decrypt button.



SENDER:
15555215556
ENTER DECRYPTION KEY:
••••••••••••••••
RECEIVED ENCRYPTED SMS
F5F3AC09EF8C5ED881E1009A2251007D0705FE8CFAB7120C0D433B6FB1541709
DECRYPTED MESSAGE:
hi, how are you doing?

Figure 8: Decrypted SMS

To access the data from the main.xml we created java class named **EncActivity.java** where we are accessing the value of receiver no, secret key, message content, message send button, and cancel button value .



```
public class EncDecSMSActivity extends Activity {
/** Called when the activity is first created. */
EditText recNum;
EditText secretKey;
EditText msgContent;
Button send;
Button cancel;
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
```

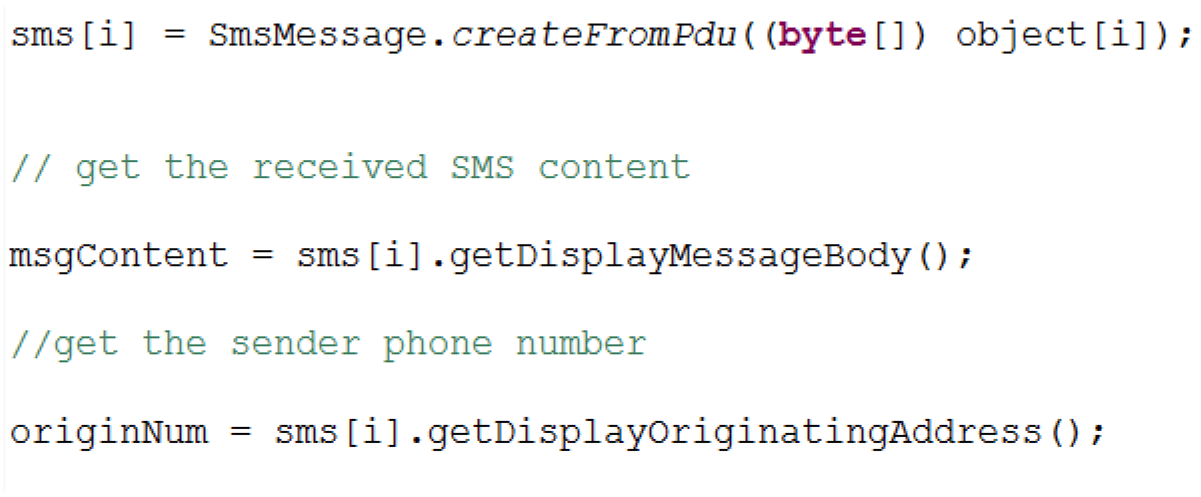
Figure 9: Encrypt Activity

There are another two java classes has been implemented named DisplayActivity.java and Broadcast.java. To broadcast the message through wireless network Broadcast.java file is responsible where some functions is implemented.

For Android application a special permission is needed to send a message one to other. The function which is responsible for permission is given below:

Bundle bundle = intent.getExtras ();

All other necessary functions which are needed to broadcast as well as receive sender number, message content are given below:

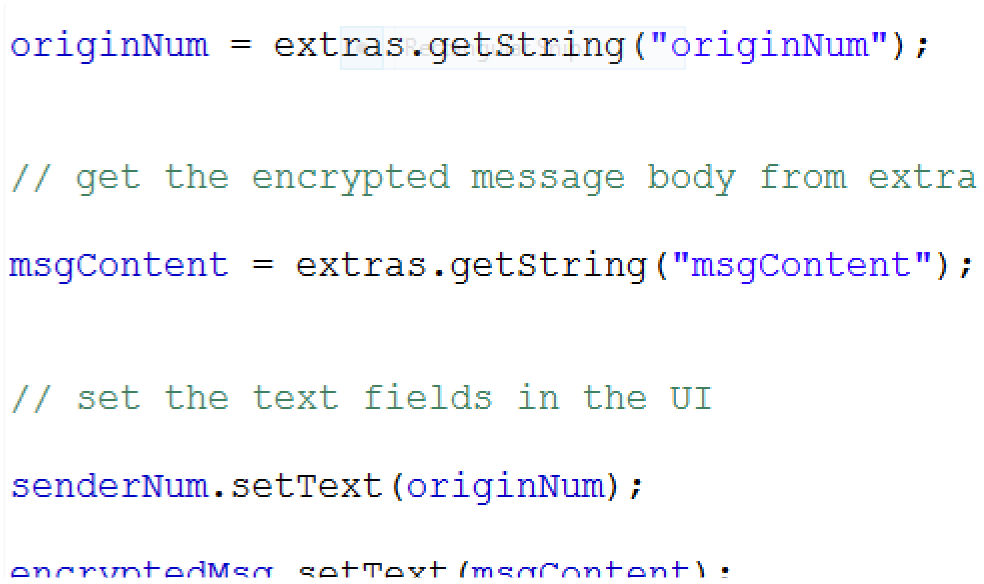


```
sms[i] = SmsMessage.createFromPdu((byte[]) object[i]);

// get the received SMS content
msgContent = sms[i].getDisplayMessageBody();
//get the sender phone number
originNum = sms[i].getDisplayOriginatingAddress();
```

Figure 10: Receiver and sender field content

DisplayActivity.java is responsible to get back the data from broadcast message. So there are also some functions which are used to catch the data in the corresponding fields in the receiver end.



```
originNum = extras.getString("originNum");

// get the encrypted message body from extra
msgContent = extras.getString("msgContent");

// set the text fields in the UI
senderNum.setText(originNum);
encryptedMsg.setText(msgContent);
```

Figure 11: Display activity

The layout values will come through these 2 methods from the sender end as well as receiver end respectively. In the Eclipse with SDK software **R.java** file is generated automatically to create some integer dimensions of the field. This file is made under the **gen folder**

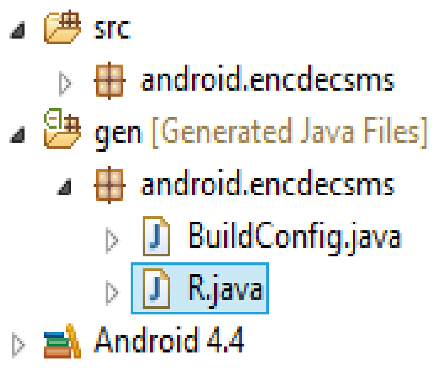


Figure 12: R. java file

4.4 TESTING AND VALIDATION

Testing is a process by which application software developed for hand held mobile devices is tested for its functionality, usability and consistency. Mobile application testing can be automated or manual type of testing. Mobile applications either come pre-installed or can be installed from mobile software distribution platforms. Mobile devices have witnessed a phenomenal growth in the past few years.

The following types of testing are performed on applications.

Usability testing- To make sure that the mobile app is easy to use and provides a satisfactory user experience to the customers. Usability testing is carried out to verify if the application is achieving its goals and getting a favorable response from users. This is important as the usability of an application is its key to commercial success. According to the user view this is more reliable and easy to understand that encrypted message is being sent.

Compatibility testing- Testing of the application in different mobiles devices, browsers, screen sizes and OS versions according to the requirements. It is supporting in different company android mobiles

Interface testing- Testing of menu options, buttons, bookmarks, history, settings, and navigation flow of the application. Buttons, text fields are successfully working.

Services testing-Testing the services of the application is offline.

Low level resource testing- Testing of memory usage, less time, less data are required for this application. Total size of the software is to be less after installing in android device, so less memory will be occupied.

Performance testing-Testing the performance of the application by changing the connection from 2G, 3G to WIFI, sharing the documents, battery consumption, etc.

Operational testing-Testing of backups and recovery plan if battery goes down, or data loss while upgrading the application from store.

Installation tests-Validation of the application by installing /uninstalling it on the different versions of devices and different versions of operating system

Security Testing-Testing an application to validate if the information system protects data or not. Here as sending the encrypted message so it is maintaining high security.

- ❖ **GUI (**Graphical User Interface) is very attractive to user and easy to implement.

- **Emulators –** The use of these is extremely useful in the initial stages of development, as they allow quick and efficient checking of the app. Emulator is a system that runs software from one environment to another environment without changing the software itself. It duplicates the features and work on real system.

CHAPTER FIVE

5.0 DISCUSSION

5.1 Introduction

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

5.2 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.

5.3 Discussion of Results in Relation to Specific objectives

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 secure messaging application can send, receive, encrypt, decrypt and change messages key as per objective.

5.4 Challenges

The project ensured the development of a secure messaging application that targeted Android 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.

CHAPTER SIX

6.0 CONCLUSION

The outcome of the system evaluation showed that the system could send and receive 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 Application 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.

REFERENCES.

Agoyi M. and Seral Devrim, SMS Security: An Asymmetric Encryption Approach, IEEE International Conference on Wireless and Mobile Communications, 2010, 448-452.

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.

Bhaskar Sarma, "Android Permissions: A Perspective Combining Risks and Benefits", SACMAT'12, June 20–22, 2012, Newark, New Jersey, USA. ACM 978-1-4503-1295-0/12/06 (pp 13-22)

David Barrera, "Understanding and Improving App Installation Security Mechanisms through Empirical Analysis of Android", SPSM'12, October 19, 2012, Raleigh, North Carolina, USA. ACM 978-1-4503-1666-8/12/10 (pp 81-92)

Ding Jing, The Implementation of FPGA-based RSA Public-Key Algorithm and Its Application in Mobile-Phone SMS

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

Kathy Wain Yee Au, "PScout: Analyzing the Android Permission Specification", CCS'12, October 16–18, 2012, Raleigh, North Carolina, USA. ACM 978-1-4503-1651-4/12/10 (pp 217-228)

Lisonek David and Martin Drahanský, SMS Encryption for Mobile Communication, IEEE International Conference on Security Technology, 2008, 198 – 2011

Madhwani Manisha, "Cryptography On Android Message Application using Look Up Table and Dynamic Key (Cama)", IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661, ISBN: 2278-8727 Volume 6, Issue 2 (Sep-Oct.2012), PP 54-59

Manoj. B, Manjula N Harihar, "Image Encryption and Decryption using AES" International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-5, June 2012.

NIST Computer Security Division's (CSD) Security Technology Group (STG) (2013). "Block cipher modes". Cryptographic Toolkit. NIST. Retrieved April 12, 2013.

NIST Computer Security Division's (CSD) Security Technology Group (STG) (2013). "Current modes". Cryptographic Toolkit. NIST. Retrieved April 12, 2013.

Rupa CH. and Avadhani P.S., Message Encryption Scheme Using Cheating Text, IEEE International Conference on Information Technology, 2009, 470-474.

Swati Paliwal and Ravindra Gupta, "A Review of Some Popular Encryption Techniques", International Journal of Advanced Research in Computer Science and Software Engineering Research Paper, Volume 3, Issue 2, February 2013, ISSN: 2277 128X Available online at: www.ijarcsse.com

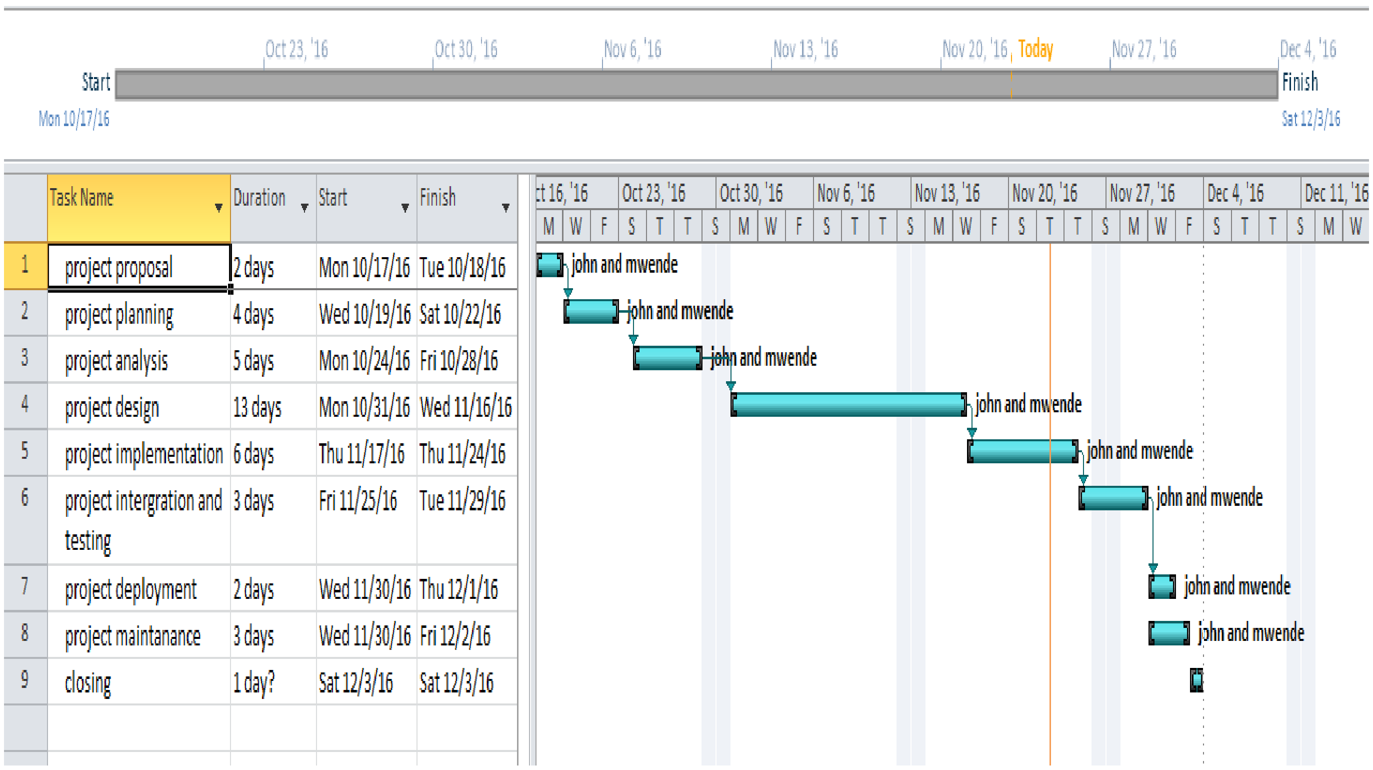
Toorani M. and. Behesti A. A, SSMS – A Secure SMS Messaging Protocol for the M-Payment Systems, IEEE Symposium on Computers and Communications, 2012, 700-705

Appendix I: Cost Estimation

Resources	Quantity	Cost	
Computer	2	@49000	98,000
Android phone	2	@10000	20,000
Internet			7,400
Android SDK	1		Free
Android ADT	1		Free
Eclipse	1		Free
Totals			125,400

Table 1: Cost Estimation

Appendix II: Project Schedule



	Task Name	Duration	Start	Finish
1	project proposal	2 days	Mon 10/17/16	Tue 10/18/16
2	project planning	4 days	Wed 10/19/16	Sat 10/22/16
3	project analysis	5 days	Mon 10/24/16	Fri 10/28/16
4	project design	13 days	Mon 10/31/16	Wed 11/16/16
5	project implementation	6 days	Thu 11/17/16	Thu 11/24/16
6	project intergration and testing	3 days	Fri 11/25/16	Tue 11/29/16
7	project deployment	2 days	Wed 11/30/16	Thu 12/1/16
8	project maintanance	3 days	Wed 11/30/16	Fri 12/2/16
9	closing	1 day?	Sat 12/3/16	Sat 12/3/16

Table 2: Project Schedule

Appendix III: User Manual

Messaging Menus

Sender end

Step 1: Go to Mobile Applications on your Phone and choose S-SMS.

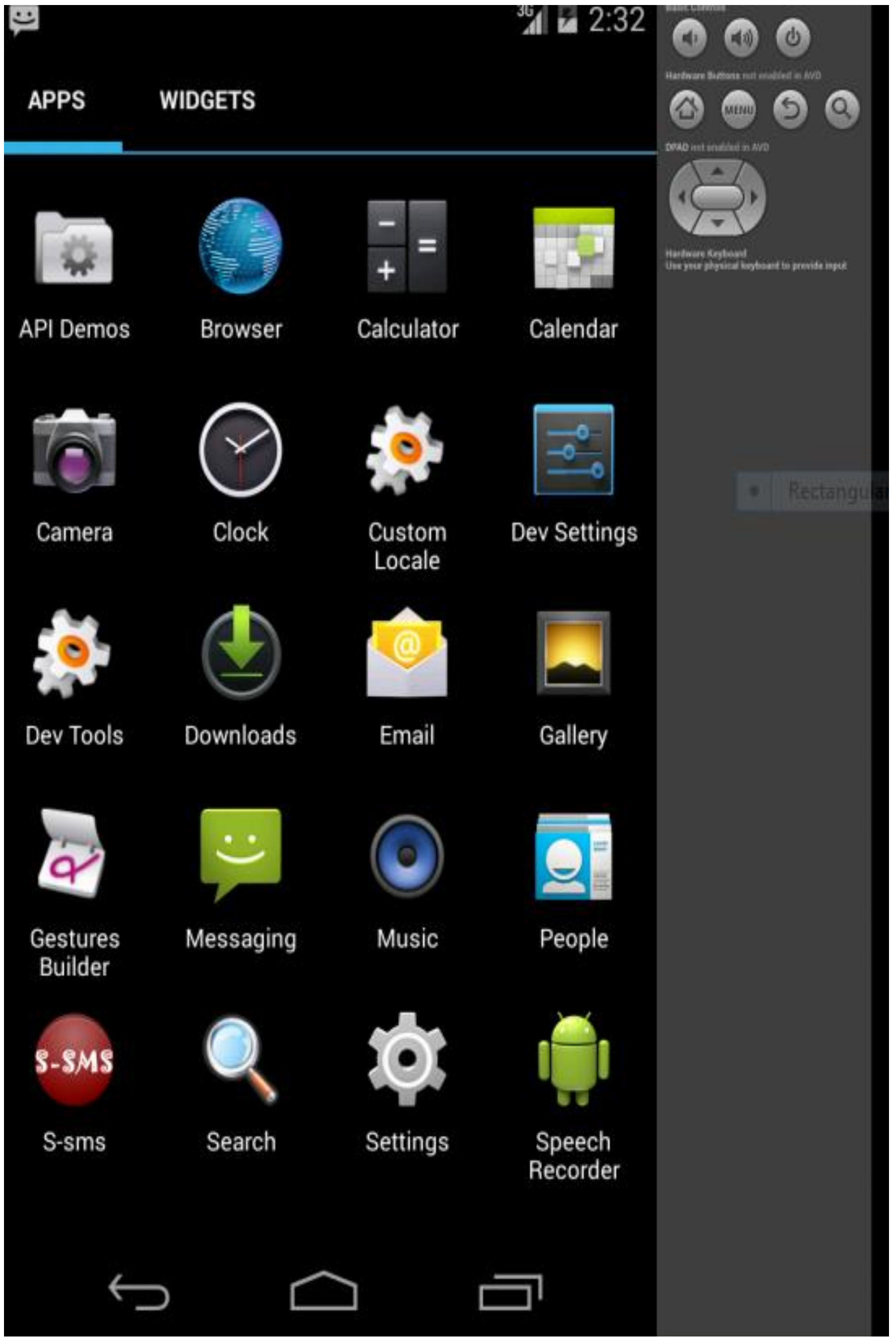


Figure 13: Messaging sender end

Step 2: Double click the S-SMS, the following messaging platform will appear.

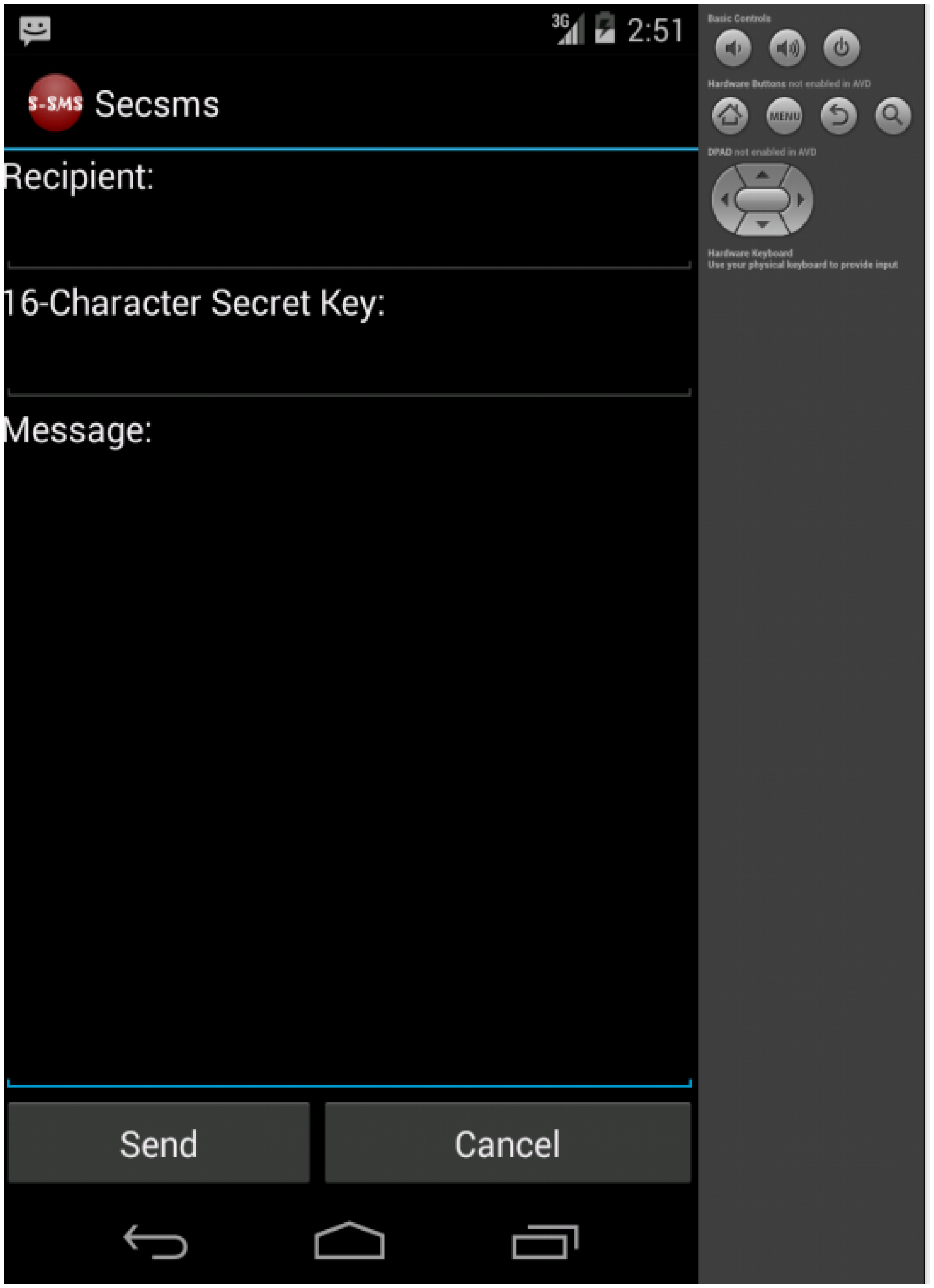


Figure 14: Messaging platform

Step 3: To compose a message, enter the recipient's phone number then type in your text message to be sent, Add the secret KEY for encrypting the messages before sending (the key should be 16 characters long). Once you click **Send button** your message will be encrypted and then it will be sent.

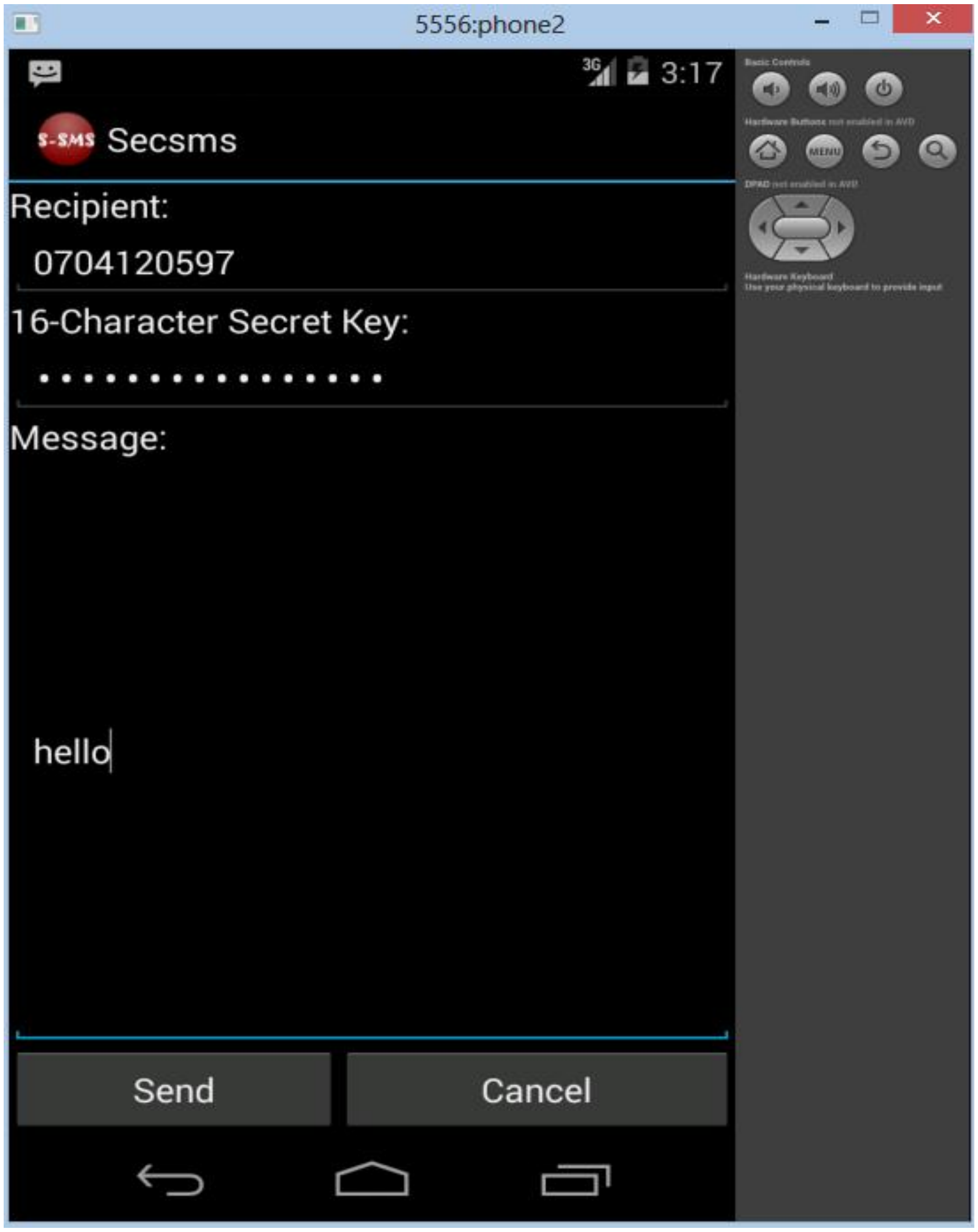
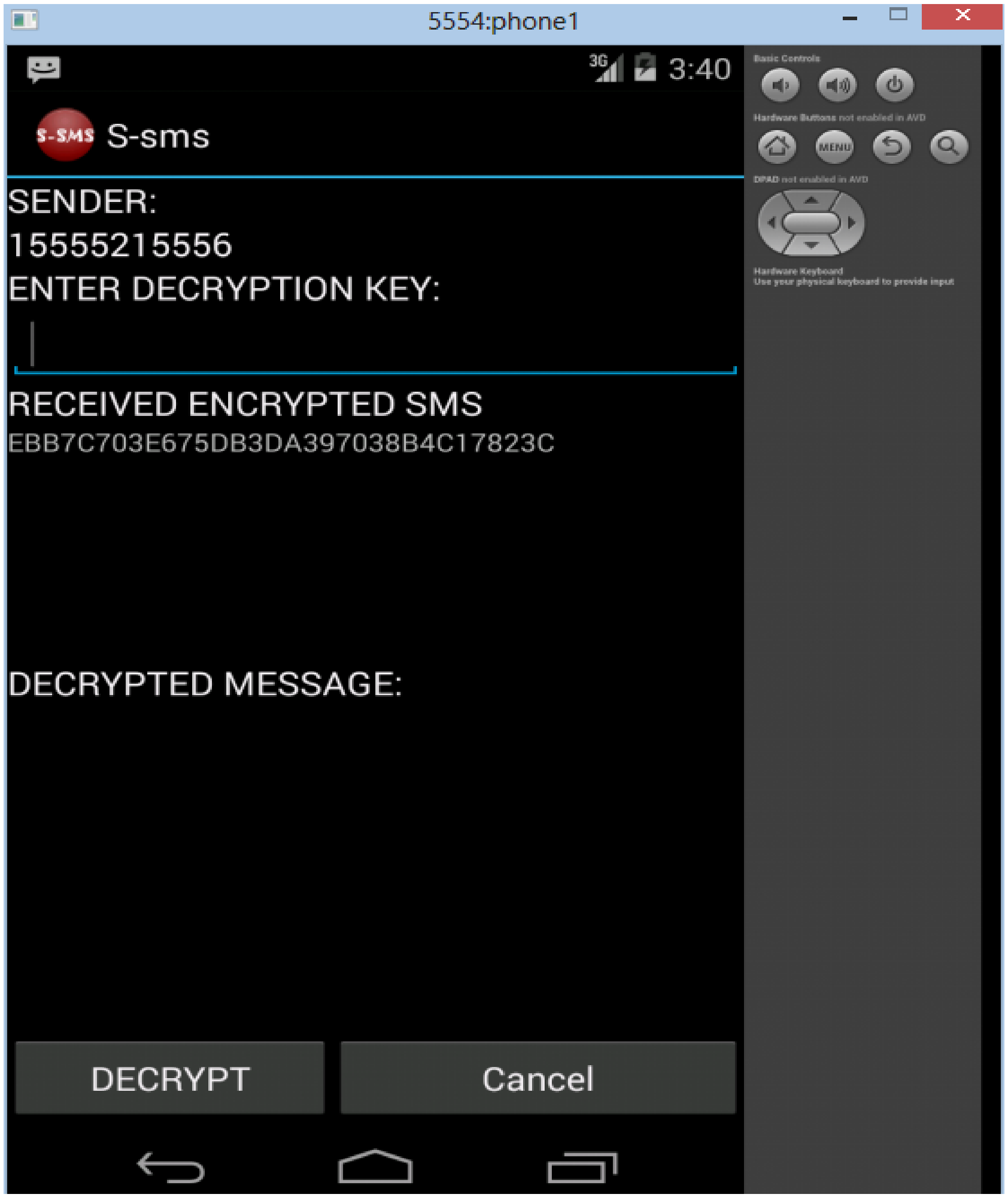


Figure 15: Field for composing SMS

Receiver end

Step 1: When the receiver receives the message, he/she should enter the decryption key and then press the **decrypt button.**



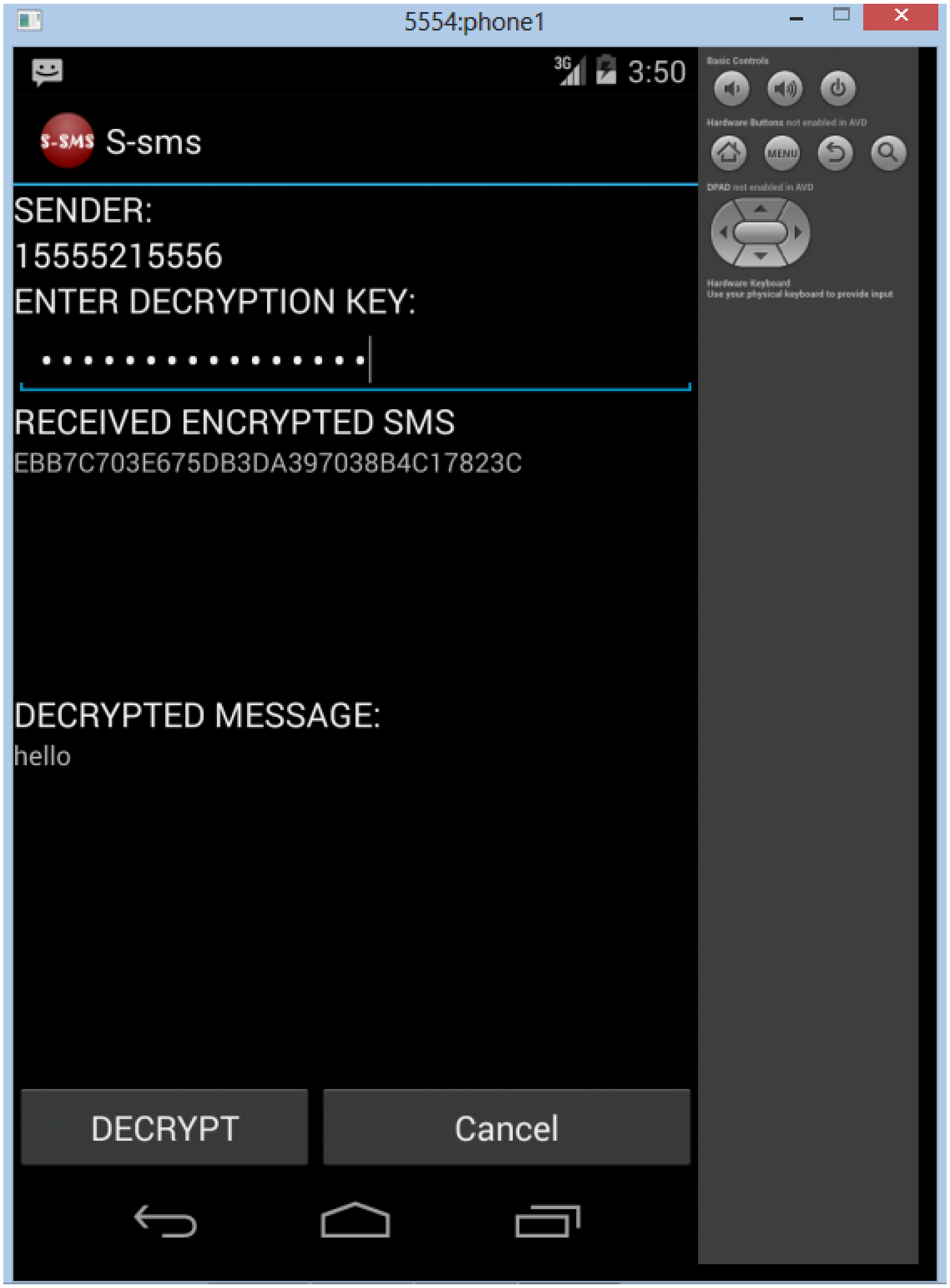
SENDER:
15555215556
ENTER DECRYPTION KEY:

RECEIVED ENCRYPTED SMS
EBB7C703E675DB3DA397038B4C17823C

DECRYPTED MESSAGE:

Figure 16: Receiver decryption interface

Step 2: The message will be converted from the cipher text to plain text.



SENDER:
15555215556
ENTER DECRYPTION KEY:
••••••••••••••••
RECEIVED ENCRYPTED SMS
EBB7C703E675DB3DA397038B4C17823C
DECRYPTED MESSAGE:
hello

Figure 17: Decrypted text

Appendix IV: SOURCE CODES

Encryption decryption source codes

```
public class EncDecMSActivity extends Activity {
public void onCreate(Bundle savedInstanceState) {
super.onCreat (savedInstanceState);
SetContentView (R.layout.main);

recNum = (EditText) findViewById(R.id.recNum);
SecretKey = (EditText) findViewById (R.id.secretKey);
MsgContent = (EditText) findViewById (R.id.msgContent);
Send = (Button) findViewById (R.id.Send);
cancel = (Button) findViewById(R.id.cancel);
cancel.setOnClickListener (new View.OnClickListener () {
Public void on Click (View v) {
Finish ();
}
});

send.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
String recNumString = recNum.getText().toString();
String secretKeyString = secretKey.getText().toString();
String msgContentString = msgContent.getText().toString();
if (recNumString.length() > 0 && secretKeyString.length() > 0
&& msgContentString.length() > 0
&& secretKeyString.length() == 16) {

byte[] encryptedMsg = encryptSMS(secretKeyString,
msgContentString);
String msgString = byte2hex(encryptedMsg);
sendSMS(recNumString, msgString);
finish();
byte[] encryptedMsg = encryptSMS(secretKeyString,
msgContentString);
String msgString = byte2hex(encryptedMsg);
sendSMS(recNumString, msgString);
finish();
} else
Toast.makeText(
getBaseContext(),
"Please enter phone number, secret key and the message. Secret key must be 16 characters!",
Toast.LENGTH_SHORT).show();
}
});
}
public static void sendSMS(String recNumString, String encryptedMsg) {
try {
SmsManager smsManager = SmsManager.getDefault();
ArrayList<String> parts = smsManager.divideMessage(encryptedMsg);
smsManager.sendMultipartTextMessage(recNumString, null, parts,
null, null);
} catch (Exception e) {
e.printStackTrace();
}
}
public static String byte2hex(byte[] b) {
String hs = "";
String stmp = "";
for (int n = 0; n < b.length; n++) {
stmp = Integer.toHexString(b[n] & 0xFF);
if (stmp.length() == 1)
hs += ("0" + stmp);
else
```

```
hs += stmp;
}
return hs.toUpperCase();
}
public static byte[] encryptSMS(String secretKeyString,
String msgContentString) {
try {
byte[] returnArray;
Key key = generateKey(secretKeyString);
Cipher c = Cipher.getInstance("AES");
c.init(Cipher.ENCRYPT_MODE, key);
returnArray = c.doFinal(msgContentString.getBytes());
return returnArray;
} catch (Exception e) {
e.printStackTrace();
byte[] returnArray = null;
return returnArray;
}
Key key = generateKey(secretKeyString);
}
private static Key generateKey(String secretKeyString) throws Exception {
Key key = new SecretKeySpec(secretKeyString.getBytes(), "AES");
return key;
}}
```