Development and Analysis of Message Embedding System for Embedded OS Using Spatial Watermarking Technique

BY

LOI KONG LEONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements for the degree of

BACHELOR OF COMPUTER Engineering (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

1 April 2013

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Development and Analysis of Message Embedding System for Embedded OS Using Spatial Watermarking Technique

Academic Session: April 2013

I LOI KONG LEONG

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ _____

(Author's signature) (Supervisor's signature)

Address:
No. 15 Taman Ayer Tawar 2,
32400 Ayer Tawar,
Perak.

_____
Supervisor's name

Date: _____ **Date**: _____

DECLARATION OF ORIGINALITY

I declare that this report entitled

"DEVELOPMENT AND ANALYSIS OF MESSAGE EMBEDDING SYSTEM FOR EMBEDDED OS USING SPATIAL WATERMARKING TECHNIQUE"

is my own work except as cited in the references.

The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : Loi Kong Leong

Date : _____

Acknowledgement

Apart from the efforts of myself , the success of this project I would like to gratefully acknowledge the enthusiastic supervision of Sir Leong Chun Farn during this work. He inspired me to work on this project and help me in enhancing my idea of the project. I also would like to thank him for not only showing me any example but to guide me on how to find resources that related to the topic of this project.

Besides, I would like to thank the authority of University Tunku Abdul Rahman (UTAR) for giving me the opportunities in creating my own project by offering this subject Project 2 (UCCE3506) which would have helped me a lot in gaining experience for the future work and providing us with a natural good environment and facilities to complete this project.

Last but not least I would like to express my sincere gratitude to Sir Albert Einstein and Jason Mraz for their quotes which inspired me on my idea for the project.

"The secret to creativity is knowing how to hide your sources. "

A quote cited by Albert Einstein

"A picture can say 1000 words but it can also inspire you to write 1000 more."

A quote cited by Jason Mraz

Abstract

The main perseverance of this Final Year project is in development and analysis message embedding system for embedded OS using spatial watermarking technique.

There is plenty of Android application are available currently are mainly for entertainment purposes thus users Android phone is lack of security protection in terms of privacy, indeed they does not know how important to have an application to secure their message which may be very important. Under these circumstances, an effulgent idea of analyzing different encryption techniques and develop a message embedding system Android based application is proposed. This project will be using two types of encryption methods which are cryptography and steganography along with the technique implemented.

Throughout the project, these two algorithms will be concisely studied and the pros and cons of each algorithm will be explained in detail. In a nutshell, this project will analyze the methods and also several useful techniques developed. Hence, a report which contains a combination of these 2 algorithms will be documented, alongside with a depiction of comparison tables. The results from this project will greatly benefit researchers as it's useful in understanding the range of cryptography and steganography method and comparison can be made easily, thus act as stepping stones for future application of encryption.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

SMS	Short Message Services
IDE	Integrated Development Environment
JDK	Java Development Kit
ADT	Android Development Tools
USB	Universal Serial Bus
GSM	Global System for Mobile
MMS	Multimedia Messaging Service
AES	Advanced Encryption Standard
DES	Data Encryption Standard
DCT	Discrete Cosine Transform
API	Application Programming Interface
DDMS	Dalvik Debug Monitoring service

Chapter 1: INTRODUCTION

1.1 Project Background

According to the research done by (Smith 2011) on "How Americans Use Text Messaging" between 2010 and 2011 mobile phone users could have sent or receive more than 40 SMS per days. Some of these SMS may have its own privacy value such as the information about the user's current location, account number or password. Under these circumstances, encryption has become the most desirable solution to embark upon this matter. Text encrypted into unknown text or picture is on security protection for both sender and receiver as encrypted text can't be known easily.

Recently a new Android application was reported to have the abilities to spy on Android phone users SMS (Kelly 2012), (SecretSMSReplicator 2010) which means the Android users' privacy send through SMS will be threatened. This information is critical and it can be dangerous if exposed to the anonymous. Therefore another preventive Android application is necessary to protect the Android phone users from it.

Encryption is the process of transforming information or data (the plaintext) using an algorithm (called a cipher) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. The person who intent to decipher the information must firstly obtain the key.

The process will produce a pieces of encrypted information (in cryptography, referred to as ciphertext). The reverse process which is to make the encrypted information readable again, is referred to as decryption (to get the readable form of information). (Encryption 2013)

1.2 Project Objective

To develop an Android Application that has the ability to secure the message from intruders while sending through the network using the knowledge of encryption and message embedding technique.

1.3 Deliverables

- A multi-touch screen phone with Android Operating System installed plus a USB connector cable.
- A computer system with a minimum system requirement of:
    - Windows XP (32-bit), Vista (32- or 64-bit), or Windows 7 (32- or 64-bit)
    - Mac OS X 10.5.8 or later (x86 only)
    - Linux (tested on Ubuntu Linux, Lucid Lynx)
    - GNU C Library (glibc) 2.7 or later
    - On Ubuntu Linux, version 8.04 or later
    - 64-bit distributions must be capable of running 32-bit applications.

1.4 Technical Requirements

- The computer system must be configured with a programming environment for Java development and the software need to be installed on the computer are :
    - Java Development Kit (JDK)
    - Eclipse IDE
    - Android SDK
    - Android Development Tools (ADT) plug-in for Eclipse IDE
- The computer operating system must be configured to access the Android phone via the USB cable for the purpose of installing and debugging Android applications on Android phone.

1.5 Limits and Exclusions

- GSM network is not available to send message embedded image due to the size of the image is too large. The image will be converted into smaller sizes in order to send through MMS which will destroy the important bits in the image that reflects the original message. (GSM 2013)

Limited time is given to develop a Android application that support all mobile devices and fixing the application bugs.

Chapter 2: LITERATURE REVIEW

"Technology is just a tool. In terms of getting the kids working together and motivating them, the teacher is the most important."

A quote cited by Bill Gates

There's two ways to protect Android users from getting their information lost to anonymous. Both have it's own benefits and disadvantages. First is to encrypt the message into a bunch of unknown character that only can be decrypted by the intentioned receiver (Cryptography) (Stallings 2011) and another ways is to make the message concealed whereby no one could have known the existence of the message unless being informed about the message embedded (Steganography) (Steganography 2013). Due to that both method is implemented in this project to enhance the objective of the project. This project will provide the message with characteristics of not being seeing and hard to be obtained in any circumstances.

2.1 Cryptography

Cryptography method (Stallings 2011), (Cryptography 2013) posses an important role in this project on how to process the plain text (a readable form of text) into encrypted form known as cipher text (Unknown character text). After obtaining the cipher text from plain text the cipher is then send to the known recipient through network for the sake of delivering the message. However sending through network is unsafe by itself and for intruder that anticipates to acquire the message can be successful but with the encryption it would be hard and time consuming for the intruder to decrypt the message without the specialized key. It works just like a key and a lock inorder to obtain the message the recipient must use a specialized key to decrypt the message. However, if the intruder acquired the key itself the message can be easily decrypted also. An example illustrates Cryptography scrambling capabilities is shown below:

Input data:

UTAR Kampar (拉曼大學)

Creator: Loi Kong Leong

ID: 09ACB04872

Supervisor:

Mr.Leong Chun Farn

Encoded data:

22A45F0E90CA21285545CB447B74887F8CEF6C75F1DE8A8D3A0B27940955F5383D456DE3A37E1312A2BBE13B67714D41D0C2E5C6191A72CBEAE2E285DC9F42660DB1E4D99946F1A85B5A0ACCEA054F4BFBAB481656B2D9008743AFCCFE71CE326797ECBDAF253DA542513E85C02CBC9346C265421771B45B35DA47162B45FC26

Key Used: UTAR Kampar (拉曼大學)

2.1.1 Types of Cryptography

There are two types of cryptography (Stallings 2011) which is different in terms of the number of keys used and how it is used:

2.1.1.1 Symmetric key cryptography

Symmetric key cryptography is a technique whereby user used the same key for encryption as well as in the decryption process. The key can be in words, numbers, symbols or even in different languages of string. This cryptography technique offers high data rates and can be combined to produce stronger ciphers. However, the security of the message will depends on the key itself. Hence, the cipher text will become vulnerable once the key is exposed to others. So, good care should be taken while transferring the keys between the sender and receiver. This type of encryption is the oldest and yet the best known technique.

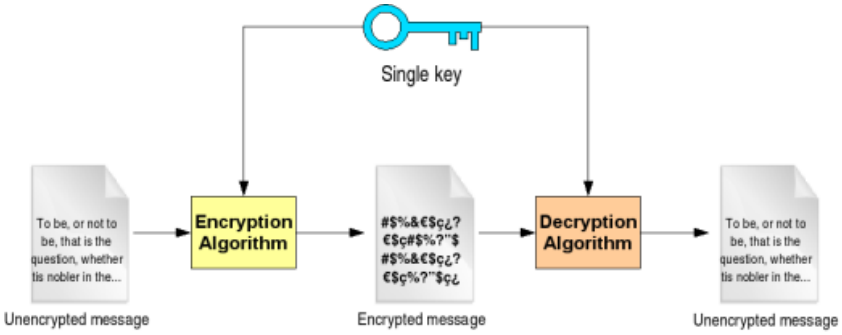

Figure 1-2.1.1.1: Symmetric key encryption and decryption process (Globusonline.org, n.d.)

2.1.1.2 Asymmetric key cryptography

In a groundbreaking 1976 paper, Whitfield Diffie and Martin Hellman presented the concept of public-key (asymmetric key) (Cryptography 2013) to cope with the drawback of symmetric key due to inconvenient needs to transfer key among the user and recipient. Asymmetric key cryptography technique offers two different but correlated keys. The key is used each for encryption and decryption. One of the keys (private key) is used to encrypt the plain text into cipher text and another key (public key) is used to decrypt the cipher text back into readable plain text. Either of the two keys were unique and neither of these key can be used in both functions. The key for decryption can be published whereas the key for encryption were kept private from others. This technique has slower data rate compare to symmetric key technique.

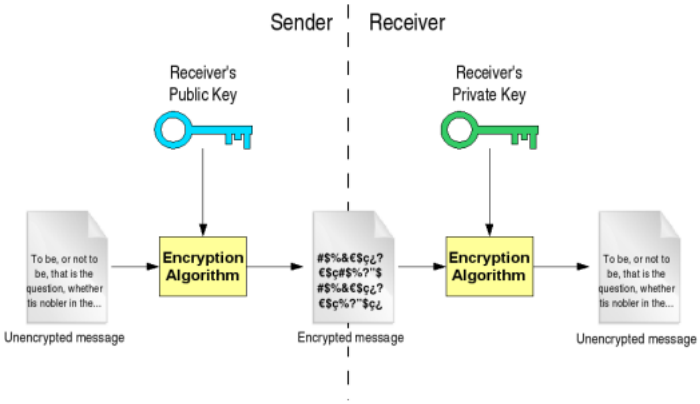

Figure 2-2.1.1.2: Asymmetric key encryption and decryption process (Globusonline.org, n.d.)

2.1.2 The method developed by scholars

2.1.2.1 Advanced Encryption Standard (AES) Method

AES (Rouse 2012), (Kamali, Hedayati, Shakerian & Rahmani 2010), (jamesedwardtracy 2010), (Computer Security Division 2001) is a block cipher intended to replace DES for commercial solicitations. This encryption method requires a 128-bit block size and a key size of 128, 192, or 256 bits for the encryption. Feistel structure is not used in AES encryption (Feistel cipher 2013). Instead, each full round consists of four separate functions:

1. Byte substitution
2. Permutation
3. Arithmetic operations over a finite field
4. XOR with a key.

Figure 3-2.1.2.1 Shows the overall structure of the AES encryption process. The cipher takes a plain text block size of 128 bits (16 bytes) but the key length can be 16 bytes (128 bits), 24 bytes (192 bits), or 32 bytes (256 bits). The algorithm is referred as AES-128, AES-192, and AES-256, depending on the key length.

A 16 byte block is used during the encryption and decryption which is then converted into a 4 x 4 square matrix of bytes. The 4 x 4 square matrix of bytes will be copied into the State array and modified for N number of times at each stage of encryption or decryption. When it reaches the final stage, the State will be copied into another square matrix output. Each word will be represented in four bytes and the total key produced is in 44 words for 128-bits key.

The cipher will undergoes N number of rounds and the number of rounds depends on the key length as shown in Table 1-2.1.2.1. The transformation started with an initial single transformation which is known as Round 0 and continues with the first N – 1 rounds which involves only three transformations. Each transformation takes one or more 4 x 4 matrices as input and output as a 4 x 4 matrix.

Figure 3-2.1.2.1 Shows that the output of each round is in a 4 x 4 matrix and the output of the final round will be the cipher text. At the same time the key expansion function will also generate N + 1 round keys which is a different 4 x 4 matrix. Each round key works as the inputs to the transformation in each round.

No. of rounds	Key Length (bytes)
10	16 (128 bits)
12	24 (192 bits)
14	32 (256 bits)

Table 1-2.1.2.1: Tables of rounds needed respective to the key length

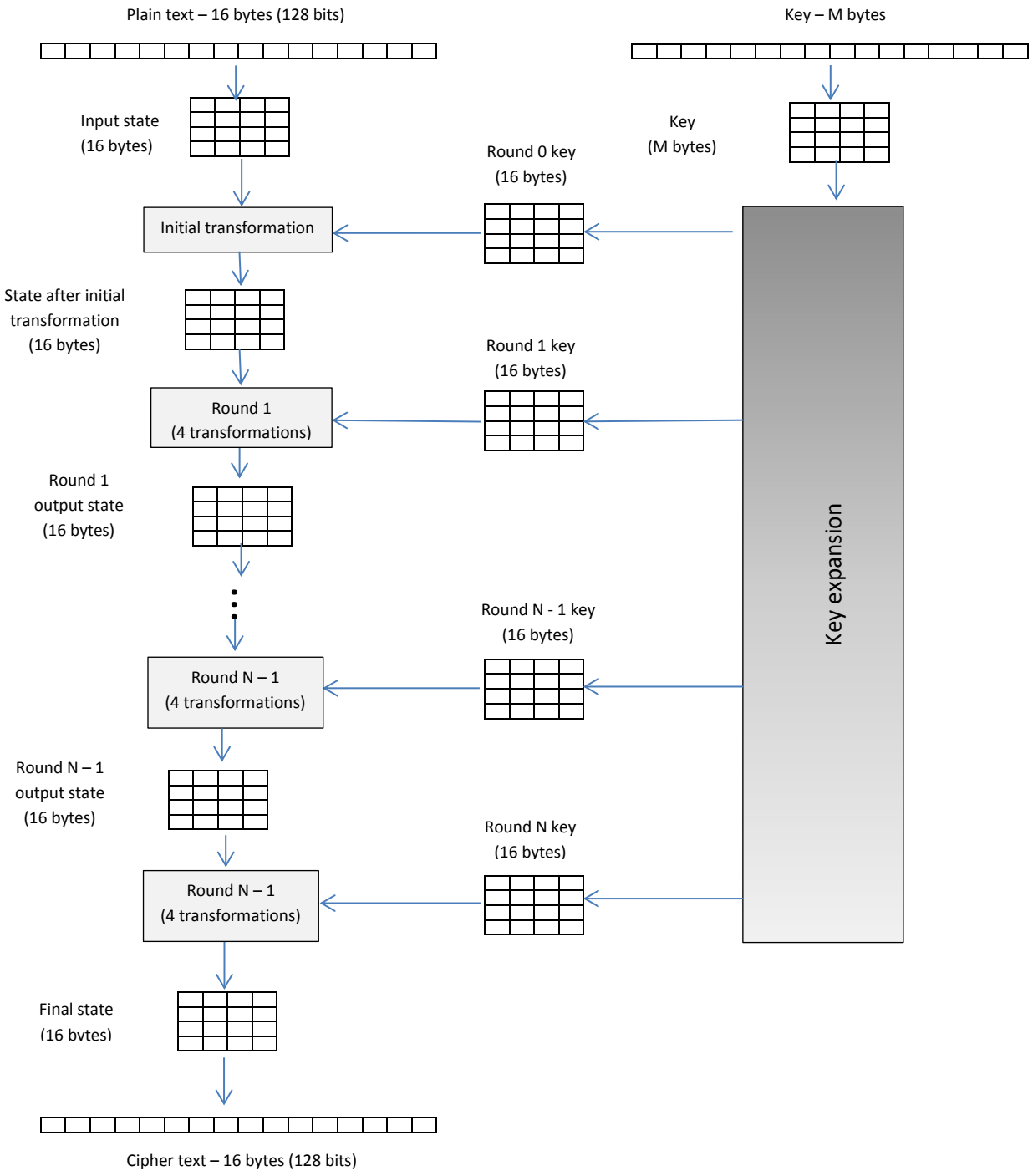

Figure 3-2.1.2.1: AES Encryption process (Stallings 2011)

2.1.2.2 Quasi group Encryption Method

Quasi group (Satti 2007) is a multilevel indexed method that encrypt data by undergo several times of permutations. This method is not only flexible but also able to enhance the security. According to (Brute-force attack 2013) it would be very difficult to break this cipher using brute force even with the knowledge of the indices and group orders because these refer to the isotopes that are present in the database of a legitimate user. It allocates the data stream based on the keys and the order of the quasi group. This unique key (Figure 4-2.1.1.1) which consists of the index numbers and the matrix orders (from the permutations that are performed) is kept secret.

The output depends on the index numbers and the orders of the matrix transformations. The encryption is also dependent on six multiplier elements that are generated by the algorithm based on the index numbers. Quasi group is implemented with group of elements along with a multiplication operator such that unique solution z can be obtained with elements x and y, also belonging to Q, such that the following two conditions are obeyed x * a = z and y * b = z.

The finite quasi group multiplication table was developed in Latin square which is a square matrix with a number of elements such that the elements does not repeat itself in either the row or the column. The figure below shows how Quasigroup data is encrypted:

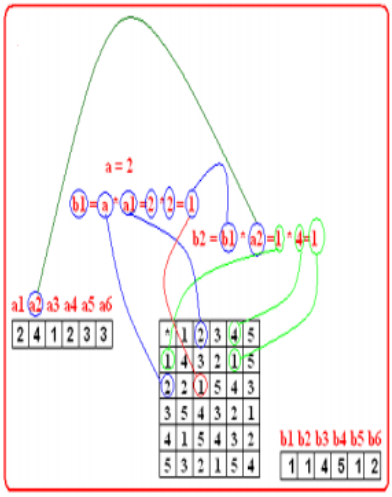

Figure 5-2.1.2.2: Quasigroup data encryption

2.2 Steganography

Steganography method (Steganography 2013), (Cummis, Diskin, Lau & Parlett 2004) is a technique used to conceal the data (plain text) such that the message is invisible from people except the sender and receiver. Digital Steganography is a technique which the information is hidden in the least significant bit of the image pixels.

There is some drawbacks for steganography compared to cryptography whereby it requires a lot of overhead to hide a relatively few bits of information, although using a scheme that proposed in the preceding paragraph may make it more effective. Also, once the system (the knowledge of message existence) is discovered, it becomes virtually worthless. This problem can be overcome if the insertion method depends on some sort of key (cryptography).

The advantage of Steganography is that the message does not attract attention to itself as the message is hidden from naked eyes.

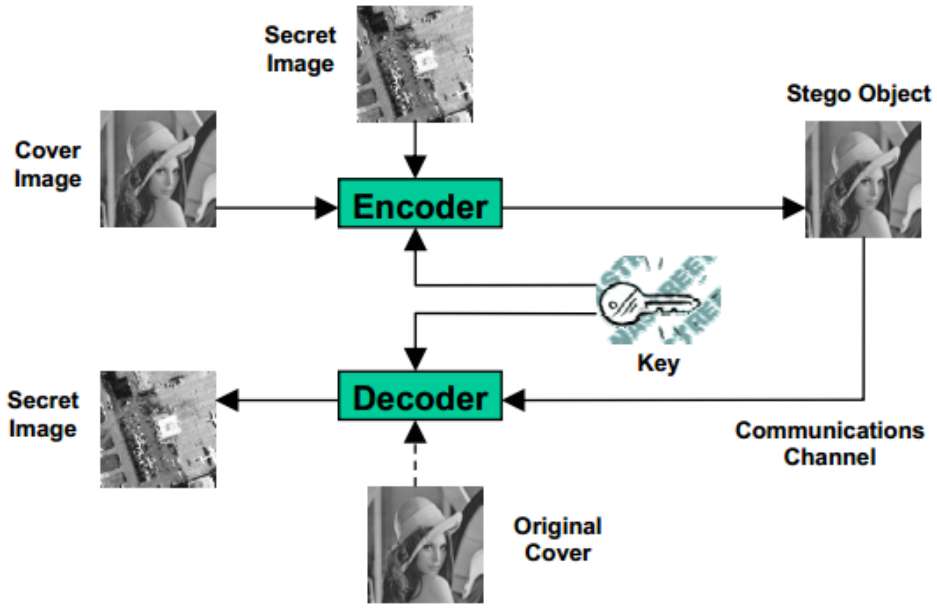

Figure 6-2.2: General Steganography process of encoding and decoding (Cummis, Diskin, Lau & Parlett 2004)

2.2.1 The method developed by scholars

The method developed by scholars on Steganography is as follows:

2.2.1.1 Watermarking

Watermarking (Cummis, Diskin, Lau & Parlett 2004), (Digital watermarking 2013), (Cao, Li & Lv 2008) is commonly used in the industries to recognize their original works when the LAWS OF MALAYSIA ACT 332 COPYRIGHT ACT 1987 (LAWS OF MALAYSIA 2000) were implemented. Digital watermarking is a technology that embeds a symbol of the copyright owner (watermark information) in the data carrier. Watermarking can be embedded in a compressed image by adding the DCT coefficients of a watermark to the quantized DCT coefficients of the compressed host signal followed by re-encoding of the watermarks by selectively discarding high-frequency DCT coefficient in certain regions of the image. The Figure 6-2.2.1.1 below clearly shows the flow of how the watermark is embedded.

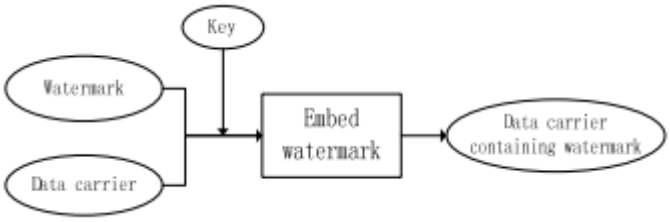

Figure 7-2.2.1.1: General embedding model of digital watermarking

2.2.1.2 Fingerprinting

Digital fingerprinting (Potdar, Han & Chang 2012), (Fingerprint (computing) 2013), (Mahmoud, Al-Hulaibah, Al-Naeem, Al-Qhatani, Al-Dawood, Al-Nassar & Al-Salman 2010) is a technique used to track unauthorized redistribution of multimedia by embedding a unique identifiable trademark into the original copy. The embedded fingerprint can later be extracted and used to trace the original distributor of the unauthorized copy. Fingerprinting must be unique in order to preserve the originality of particular work. Two works with the same fingerprinting must be avoided as it will violate the original purpose as fingerprinting is very precise about the originality. According to (Fingerprint (computing) 2013) at least 64-bit is needed for fingerprinting to guarantee a virtual uniqueness in a large file system.

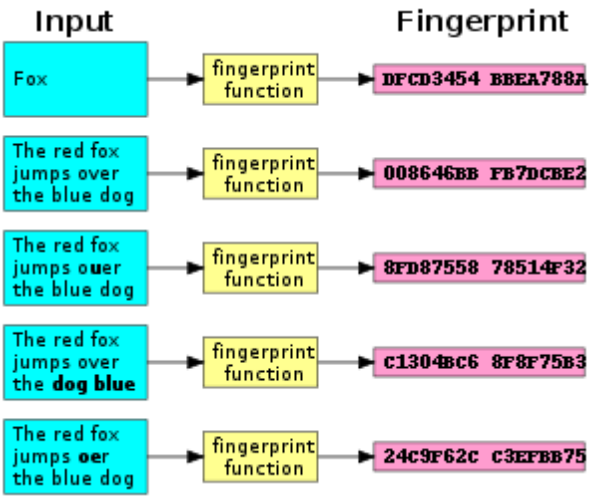

Figure 8-2.2.1.2: Sample fingerprinting results

2.3 Cryptic Steganography Method

A plain text message can be hidden either using cryptography or steganography method. The cryptography methods render the message unintelligible to outsiders by transforming the text for numerous of times using different forms of functions, whereas the methods of Steganography conceal the existence of the message. Cryptic Steganography (Sarmah & Bajpai 2009), (Rao, Kumar, Rao & Nagu 2012) is a method with double protection as it is the combination of cryptography method and Steganography method. With the combination of these two methods the unauthorized data access is reduced and the security of the message is increased.

Chapter 3: METHODOLOGY AND TOOLS

3.1 Methods/Technology Involved

The Android message security system is developed by using Eclipse IDE and Android SDK Tools with Java programming language. Eclipse is a particularly popular for Java development and commonly used with Android SDK Tools when comes to developing an Android application due to abilities to extend its capabilities by installing Android SDK plug-ins written for the Eclipse Platform. On the other way round Android SDK officially supported IDE is Eclipse using the ADT Plugin. Android code is written with Java syntax, and the core Android libraries include most of the features from the core Java APIs. The Android SDK includes all the android libraries, full documentation, excellent sample applications and also tools to help with writing and debugging applications, like the Android emulator to run projects and DDMS for debugging.

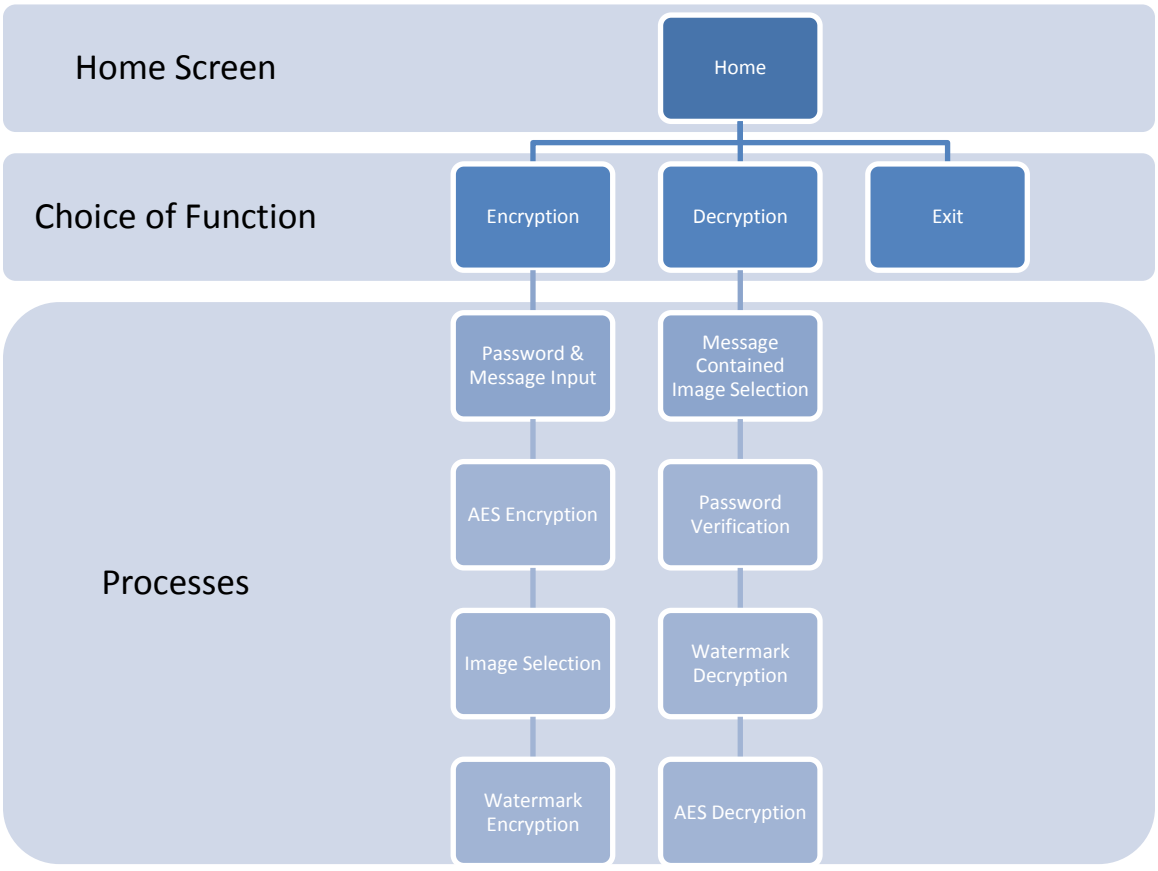

Figure 9-3.1: Application Flow Chart

3.1.1 Encryption Process

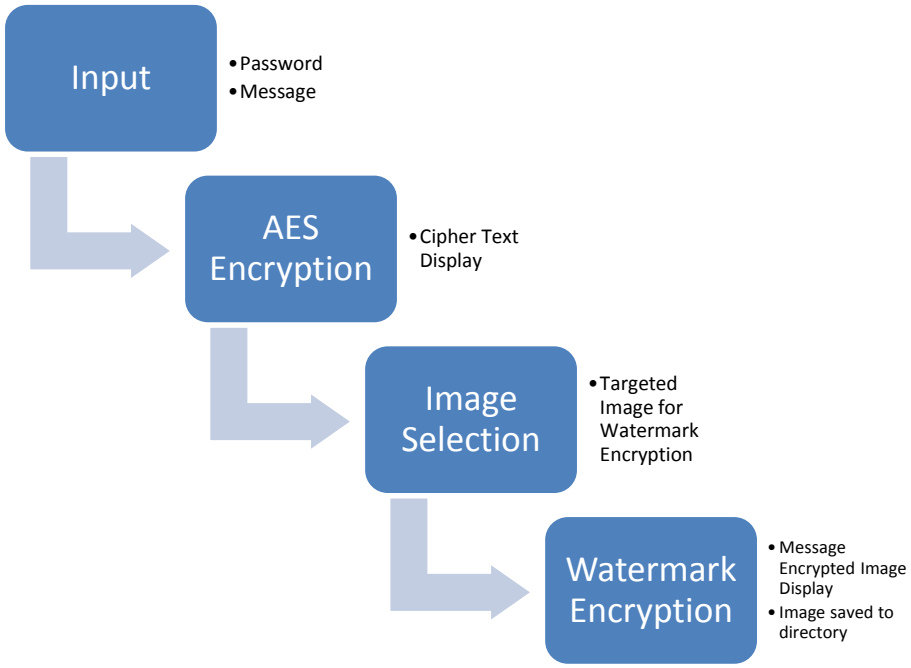

Figure 10-3.1.1: Encryption Flow

The process to develop application will be as follows:

Step1: Input

Two input will be acquired from the user. First is the secret message (plain text) and second is the password (key) used to scramble the secret message. Both input will be used in AES Encryption to produce the cipher text.

Step2: AES Encryption

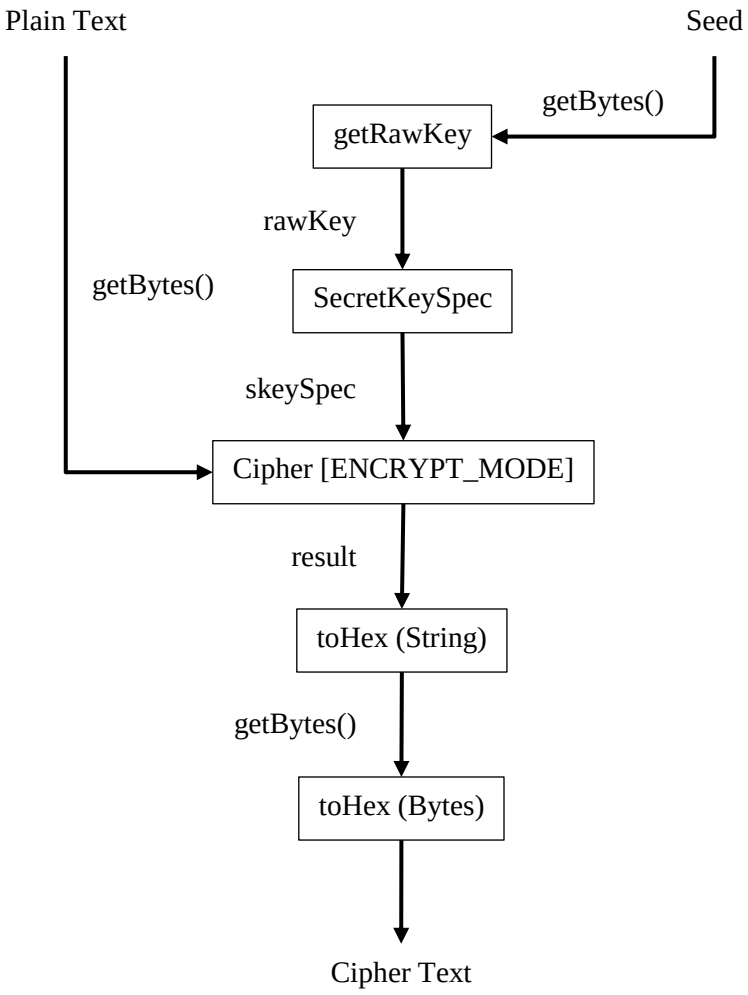

Figure 11-3.1.1: AES Encryption Flow Chart

As mentioned in step 2 a string of plain text (Message) and a string of seed (Password) is obtained from the user. The seed in the form of string is converted into bytes and passed to the function getRawKey to produce a sets of raw key.

In function getRawKey two new variables is created kgen of KeyGenerator variable type and sr of SecureRandom variable type. The seed is set into sr variable before used to initialize kgen with 128 bits type of encryption. The key is then generated using kgen and saved in SecretKey variables skey. Finally the raw bytes of the key is obtained by using the function getEncoded ().

Next, the raw key will be used in SecretKeySpec () function to create the secret key in AES. Before the encryption a variable of Cipher cipher is created with AES instance. The cipher is set to ENCRYPT_MODE and skeySpec is used to initialize the cipher.

At last, the cipher text is created by scramble the plain text using the key. However the cipher text is in the form of byte so further process using the function toHex is needed to convert the cipher text into hexadecimal string.

Figure 3-2.1.2.1 Shows the overall structure of the AES encryption process. A 16 bytes of plain text which is mentioned in step 1 is rearranged in matrix form of 4x4. Initial transformation with the round key (AddRoundKey) being carry out and a state after the initial transformation is a 16bytes 4x4 matrix form that has undergone a bitwise XOR with round key. Next is Round 1 which is comprised of 4 stage substitute bytes, shift rows, mix columns and Add round key. This will continue with Round 2, Round 3 and so on depending on the key length. As an example for 16bytes key length 10 rounds are needed before the cipher text is produced. However the round is the same for Round 1 until Round 9 and Round 10 with an exemption of Mix columns. After all 10 rounds of encryption and scrambling a 16 bytes of cipher text is produced

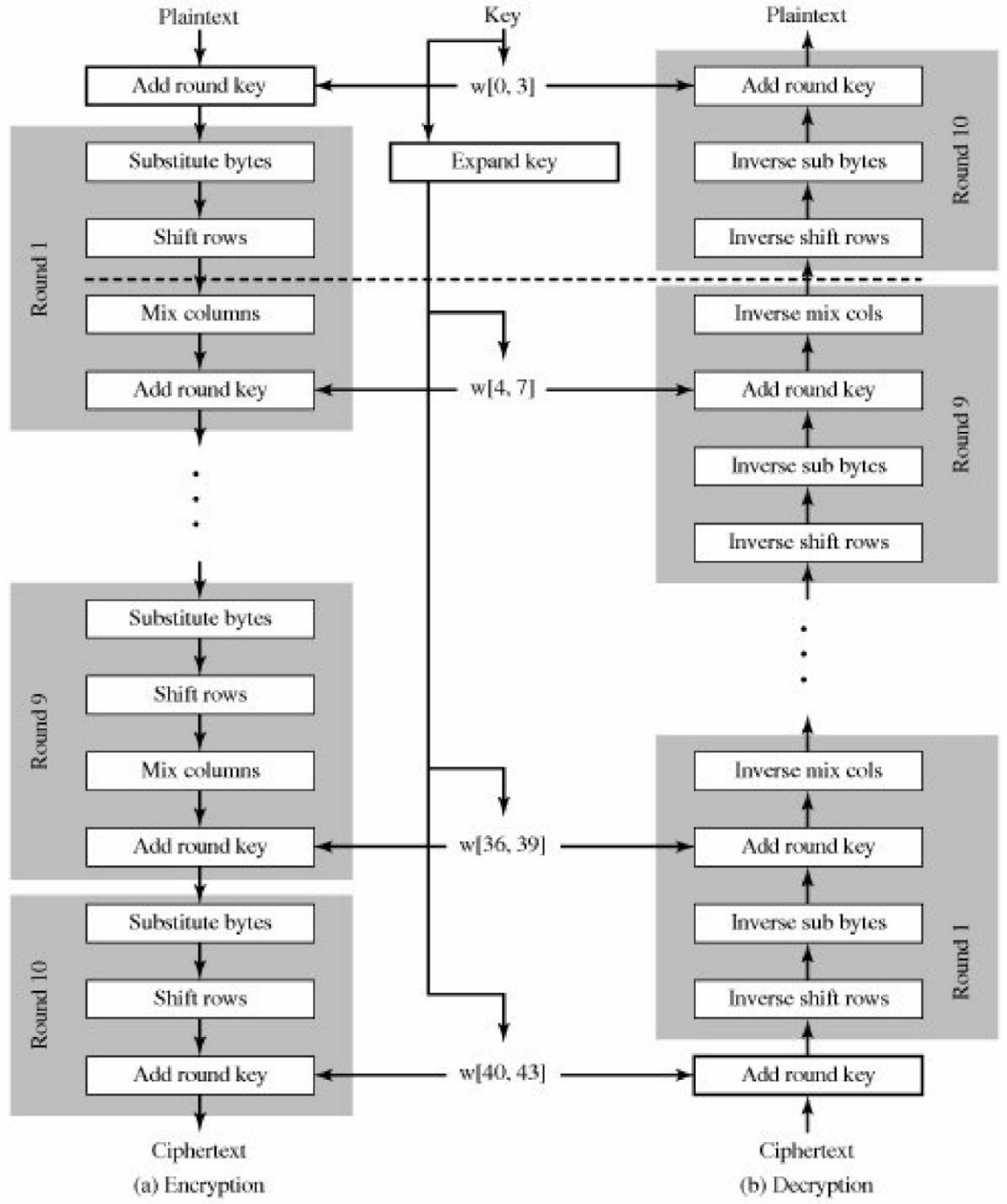

Figure 12-3.1.1: AES Encryption and Decryption process

Figure 13-3.1.1 shows the process of encryption and decryption with different stages and rounds. 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 would add no security.

Every stage of the encryption can be easily being reversed. The substitute Byte, shiftRows, and MixColumns stages can be reversed by applying the inverse function. The inverse function is applied during the decryption process. The method of XORing the round key with the block is used to inverse the ADDROUNDKEY stage.

For most of the block ciphers, reverse order of expanded key is used in the decryption algorithm. However, the decryption algorithm is not similar to the encryption algorithm. This is due to the structure of AES encryption. When it is established all of the four stages will be reversed and it will be easily be determined if the decryption has recovered the plaintext or not.

The final round of both encryption and decryption consists of only three stages. Again, this is a consequence of the particular structure of AES and is required to make the cipher reversible.

Step 3: Image Selection

A series of image obtained from the folder of mobile devices will be shown to the user in gallery and allow user to select the suitable image to be use in watermark embedding.

Step 4: Watermark Embedding

The cipher text is obtained from the AES encryption process in step 2 and the host image is obtained from the image selection mentioned in step 3. In watermark embedding process, the blue component of the image is chosen to hide the cipher text because it is less sensitive to human eyes in compare to R and G component. Each bit of the cipher text is embedded into the image pixels by using quantization method.

The cipher text getting from the AES Encryption process is originally in string. Hence, the cipher text has to be first convert from string to integer and only to binary. On the same time, the pixels of host image is obtained.

The blue component pixels is modified according to the quantizer. The pixels intensity value (f) is quantized using the equation below:

$$Q(f) = \begin{cases} 0, if\ r \times qi \leq f \leq (r+1)\,qi - 1, f\ or\ r = 0, 2, 4, 6, \ldots \\ 1, if\ r \times qi \leq f \leq (r+1)\,qi - 1, f\ or\ r = 1, 3, 5, 7, \ldots \end{cases} \quad qi = 20$$

Q(f) = quantization of pixel intensity value (f)

qi = quantization interval

r	qi	r*qi	(r+1)*qi	(r+1)*qi - 1	f ’
0	20	0	20	19	10
2	20	40	60	59	50
4	20	80	100	99	90
6	20	120	140	139	130
8	20	160	180	179	170
10	20	200	220	219	210
12	20	240	260	259	250
Q(f) = 0					

Table 2-3.1.1: Quantization Table for Q(f) = 0

r	qi	r*qi	(r+1)*qi	(r+1)*qi - 1	f ’
1	20	20	40	39	30
3	20	60	80	79	70
5	20	100	120	119	110
7	20	140	160	159	150
9	20	180	200	219	190
11	20	220	240	239	230
Q(f) = 1					

Table 3-3.1.1: Quantization Table for Q(f) = 1

After quantization, each bit of cipher text is embedded using the equation below:

$$f' = \begin{cases} \Delta f_{even} + 0.5 \times qi, if\ Q(f) = c' \\ \Delta f_{even} - 0.5 \times qi, if\ Q(f)\ \neq c' \end{cases}$$

$$f' = \begin{cases} \Delta f_{odd} \quad\quad , if\ Q(f) = c' \\ \Delta f_{odd} + qi, if\ Q(f)\ \neq c' \end{cases}$$

Where $\Delta f = \frac{f}{10} \times 10$

c' = cipher text bit

f ' = new pixel intensity value after embedding encoded watermark bit c'

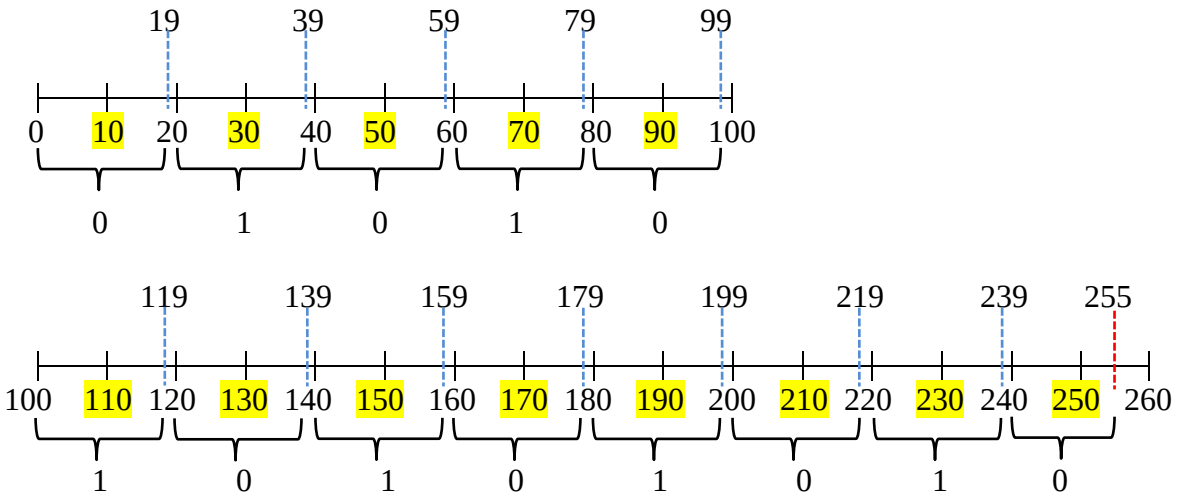

Figure 14-3.1.1: Quantization function range

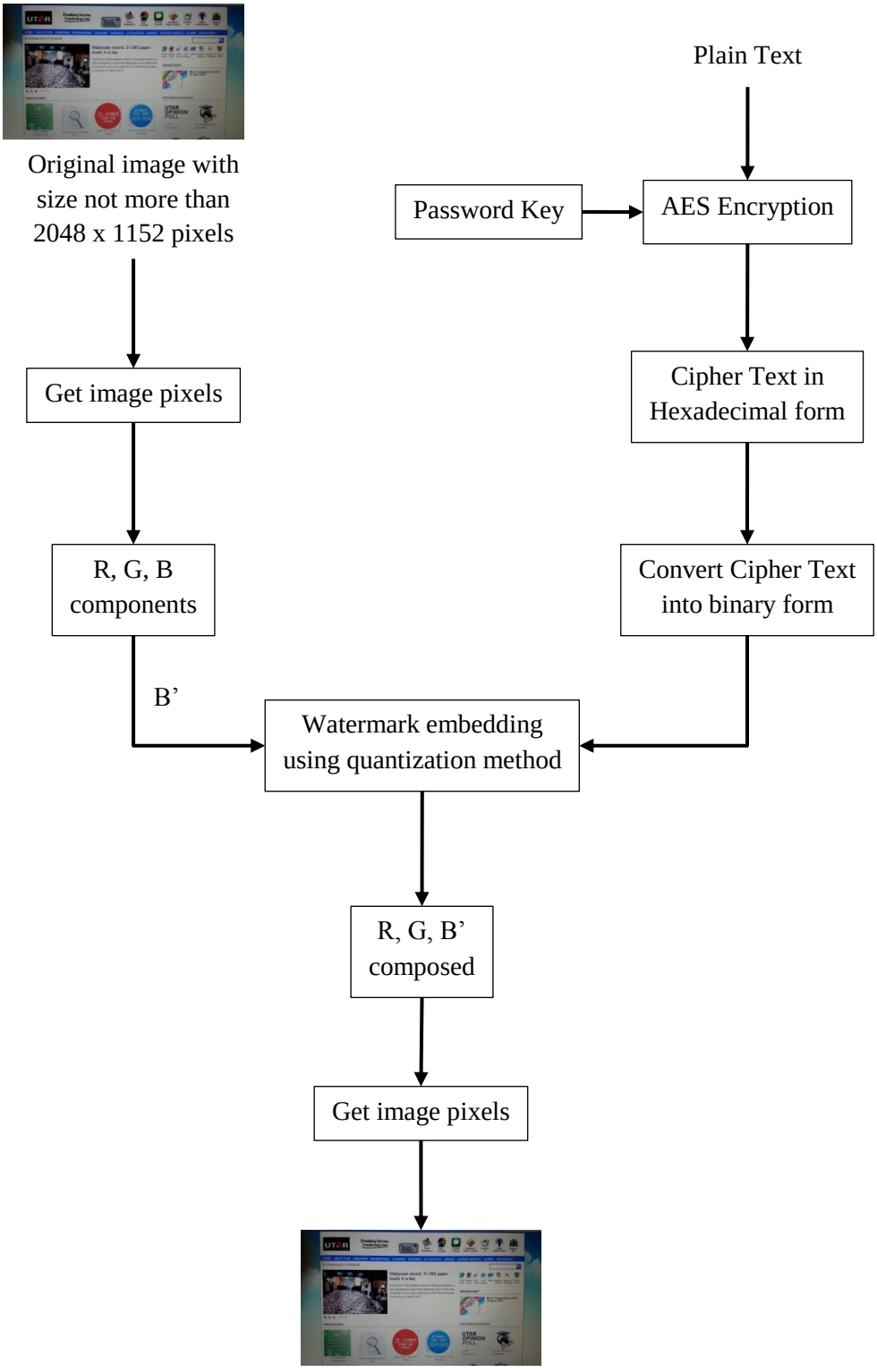

Figure 15-3.1.1: Digital Watermarking Embedding Flow

Digital Watermarking Embedding Flow

Step 8: Message Embedded Image

Lastly, a message embedded image will be obtained and ready to be sent to the receiver.

3.1.2 Decryption Process

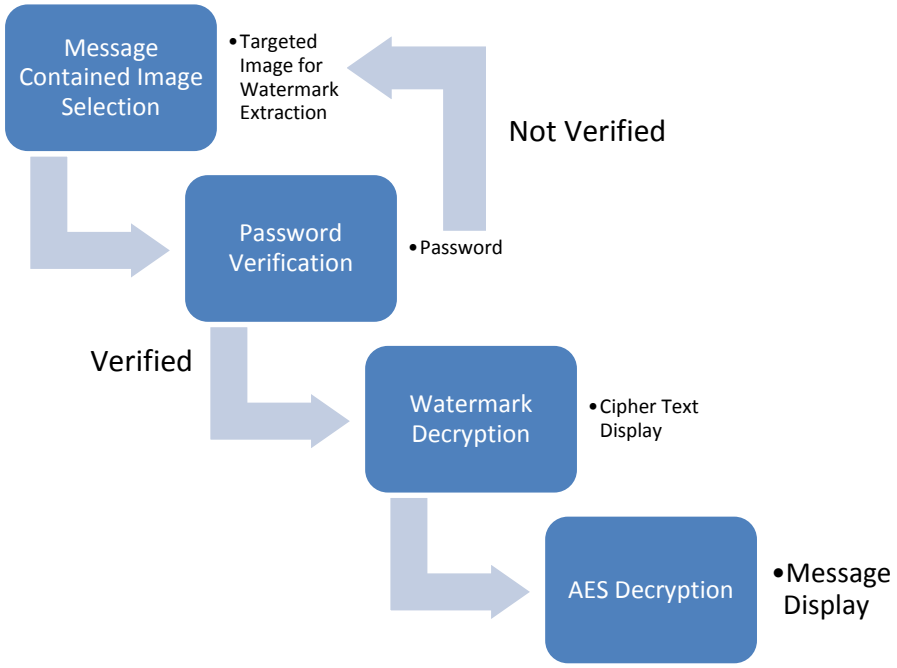

Figure 16-3.1.2: Decryption Flow

Step 1: Message Encrypted Image Selection

A successful transferred Message Encrypted Image is selected as the targeted image for watermark extraction.

Step 2: Password Verification

A password is requested and verified to determine whether to decrypt the message or not.

Step 3: Watermark Decryption (Hsu & Wu 1999), (Morkel, Eloff & Olivier n.d.)
In watermark extraction process, the cipher text is retrieved according to the quantization range shown in Figure 17-3.1.1: Quantization function range. At first the pixels of the image will be obtained from the message encrypted image mentioned in step 1 above and compare using the quantization range to determine the bit value of the cipher text.

The first 10 bits will be recognized as the total number of bits embedded into the message (not include the 10 bits). This is to prevent the application from over reading the pixels

after the range and recognized it as parts of the cipher text. Once all the cipher text bits is retrieved it will be converted back to hexadecimal form of string. The bits of cipher text is transferred back to hexadecimal 7bits by 7bits. Each of the 7 bits comprise of one hexadecimal character. Hence, after the transformation the full cipher text will be reviewed.

Digital Watermarking Extraction Flow

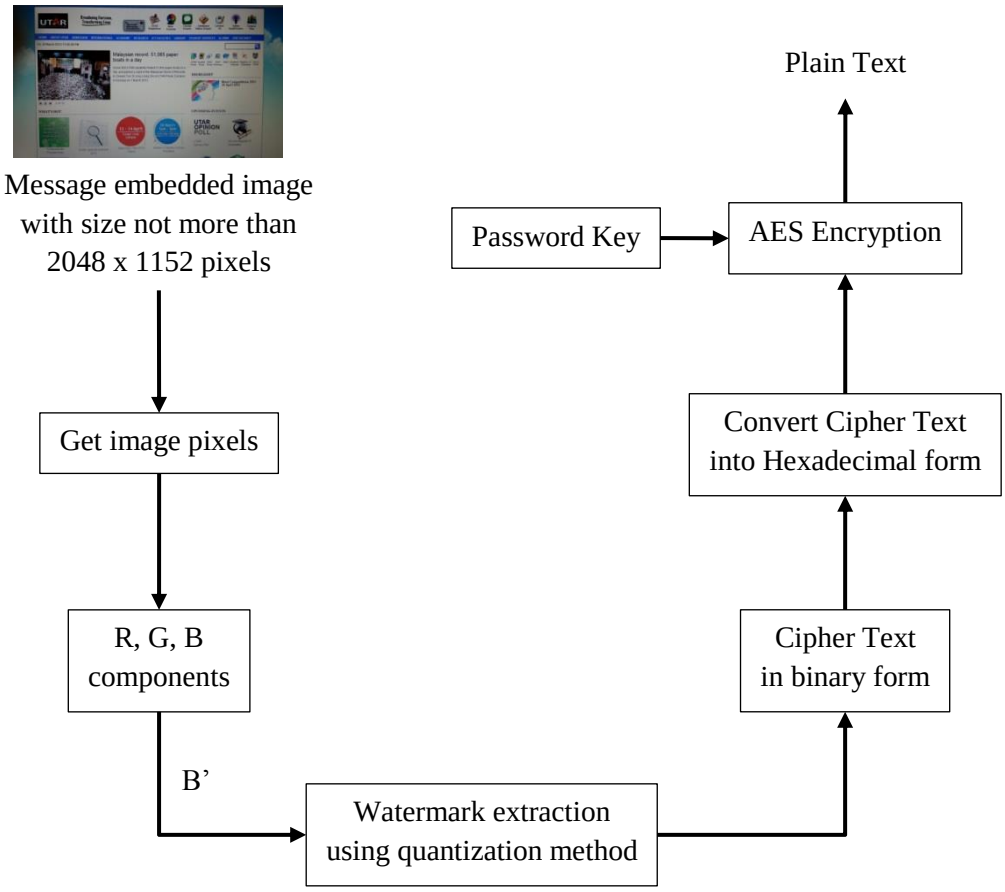

Figure 18-3.1.2: Digital Watermarking Extraction Flow

Step 5: Cipher Text

The cipher text is obtained from the watermark extraction and will be decrypted using AES Decryption to get the real message.

Step 6: AES Decryption (Rouse 2012), (Kamali, Hedayati, Shakerian & Rahmani 2010), (jamesedwardtracy 2010), (Computer Security Division 2001)

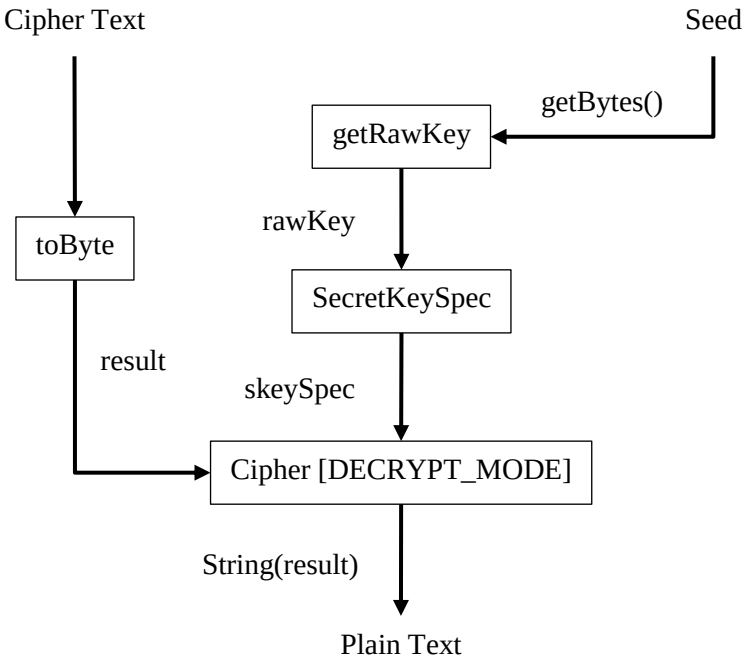

Figure 19-3.1.2: AES Decryption Flow

Most stages of Decryption are the same as Encryption but with addition of function toByte to convert the cipher text from string to byte because Cipher [DECRYPT_MODE] will process the cipher text in byte instead of string.

Besides, two function in encryption toHex (String) and toHex (Byte) are removed from the decryption flow due to the result is in the readable form of text so no further transformation needed to convert the plain text to hexadecimal. Hence, the Decryption flow chart is different from Encryption flow chart as shown in Figure 10.3.1.

Step 7: Plain Text

Final stage whereby the real message is encrypted from the cipher text and is in the form of understandable character.

3.2 Estimated Timeline to develop the project

Activities	Week												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Familiarize with eclipse IDE and learn how to master Android SDK													
Collect and formulate ideas for the selected algorithm													
Develop algorithms using eclipse IDE													
Benchmark of every algorithms													
Documentation and final report compilation													
Final adjustment													
Oral Presentation and Product Demonstration													

Table 4-3.2: Estimated Timeline to develop the project

Chapter 4: Simulations and Results

This project will be developed in windows platform using Android SDk and eclipse in Java Development language and the targeted platform for this project implementation will be Android . Hence, the simulation and result were obtained by using an Android mobile devices. Android apps usually developed using Java development language. However Android apps also can be developed in native-code languages using Android Native Development Kit but it will not benefit most of the apps.

Due to that a mobile devices running on the Android platform is necessary in this implementation. Since UTAR authority is providing Samsung galaxy W android mobile device for the final year project student the Android application created will be developed based on this model. Due to that the Android application will be running best using this model of mobile device.

The application is implemented using intent and total of eight activity were created including the home screen. An Android application can contain zero or more activities. However, when an application has more than one activity, navigation from one another is needed. In android navigation between activities is done through what is known as intent.

In the simulation process, application is installed directly into the mobile devices using eclipse. After installation the application icon will appear on the applications list. The application icon is touched so that the application execute and launch in the mobile devices. The home screen will appear on the mobile devices apparently after the touch.

4.1 Limitatations

- Mobile devices model supported

The application only run in Android based devices and run best in Samsung galaxy w Smartphone with Android Gingerbread operating system. There will be a problem on Second Activity gallery image orientation when running using other mobile device model which is shown below:

Figure 20-4.1: Gallery Image Orientation Problem Screenshot

- Size of image

The application work best for image size ranging from 0.3 megapixels (640 x 480) to 2.4 megapixels (2048 x 1152). For the application, to work on higher resolution image is only possible for mobile devices that have higher processing speed else the application will stop unexpectedly.

- Size of cropped image

The application can also work on any image that have being cropped before. However the smallest cropped image can be use should have total number of pixels around 7000 pixels. Image smaller than that will cause overflow during encryption which will lead to information lost and failure to decrypt the message using correct password.

- Message length restriction

The application only allow user to enter message not longer than 480 characters which is three pages of normal text message. This is so to support small cropped image.

- Attack and changes done to message embedded image

The application only function to protect or secure the message but not from any attack. Most of the attack and changes to message embedded image will destroy the message unless the changes does not affect the important component of the message in the image. The user can choose to send the message embedded image to few different source of the recipient to avoid this problem.

4.2 Graphical User Interface (GUI)

4.2.1 Graphic User Interface (GUI) Layout Definition

First Activity (Home Screen)

The layout of each activity is designed to have a blue colored background and with a logo before the activity title. All GUI is build according to the layout definition.

Figure 21-4.2.1: First Activity Screenshot

Layout Definition:

TextView

Id = textView1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Top = true

Layout Center Horizontal = true

Layout Margin Top = 20dp

Layout Margin Left = 20dp

Layout Margin Right = 20dp

Gravity = center horizontal

Text = "Development and Analysis of Message Embedding System for Embedded OS Using Spatial Watermarking Technique"

Text Appearance = ?android:attr/Text AppearanceLarge

ImageView

Layout Width = match parent

Layout Height = match parent

Layout Above = button1

Layout Below = textView1

Layout Margin = 30dp

Layout Center Horizontal = true

Layout Center Vertical = true

Source = ic_launcher

Button

Id = button1

Layout Width = wrap content

Layout Height = wrap content

Layout Above = button2

Layout Center Horizontal = true

On Click = onClick

Text = "Encryption"

Width = 155dp

Button

Id = button2

Layout Width = wrap content

Layout Height = wrap content

Layout Above = button3

Layout Center Horizontal = true

Layout Margin Top = 5dp

On Click = onClick

Text = "Decryption"

Width = 155dp

Button

Id = button3

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Bottom = true

Layout Center Horizontal = true

Layout Margin Top = 5dp

Layout Margin Bottom = 50dp

On Click = onClick

Text = "Exit"

Width = 155dp

Second Activity (Encryption)

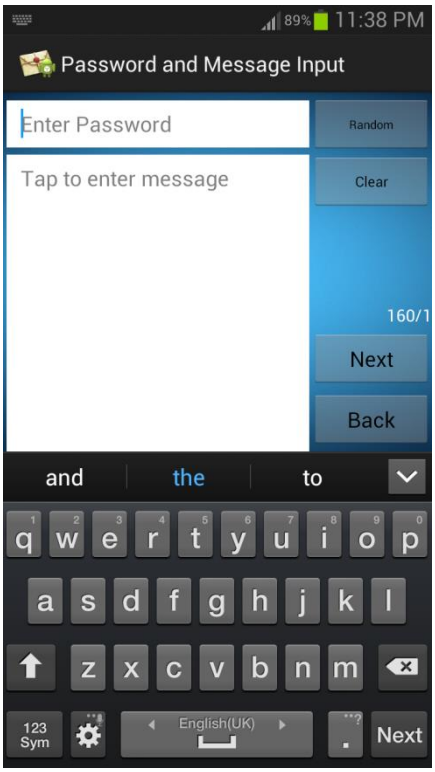

Figure 22-4.2.1: Second Activity Screenshot

Layout Definition:

EditText

Id = editText1

Layout Width = match parent

Layout Height = wrap content

Layout Align Parent Left = true

Layout Align Parent Top = true

Layout Margin Left = 3dp

Layout Margin Right = 103dp

Layout Margin Top = 7dp

ems = 10

Hint = "Enter Password"

Single Line = true

On Click = onClick

EditText

Id = editText2

Layout Width = match parent

Layout Height = match parent

Layout Align Left = editText1

Layout Align Right = editText1

Layout Below = editText1

Layout Margin Top = 6dp

Capitalize = sentences

ems = 10

Gravity = top

Hint = "Tap to enter message"

Input Type = textCapSentences|textMultiLine

Max Length = 480

Button

Id = button1

Layout Width = match parent

Layout Height = wrap content

Layout Align Left = button2

Layout Align Parent Top = true

Layout Margin Top = 3dp

On Click = onClick

Text = "Random"

Text Size = 10dp

Width = 75dp

Button

Id = button2

Layout Width = match parent

Layout Height = wrap content

Layout Align Left = button3

Layout Below = button1

On Click = onClick

Text = "Clear"

Text Size = 12dp

Width = 75dp

TextView

Id = textView1

Layout Width = wrap content

Layout Height = wrap content

Layout Above = button3

Layout Align Right = button3

Layout Margin Top = 100dp

Layout Margin Bottom = 3dp

Text = "160/1"

Button

Id = button3

Layout Width = match parent

Layout Height = wrap content

Layout Above = button4

Layout Align Left = button4

Layout Margin Bottom = 3dp

On Click = onClick

Text = "Next"

Width = 75dp

Button

Id = button4

Layout Width = 102dp

Layout Height = wrap content

Layout Align Parent Bottom = true

Layout Align Parent Right = true

Layout Margin Bottom = 3dp

Layout Margin Left = 257dp

On Click = onClick

Text = "Back"

Third Activity (Encryption)

Layout Definition:

TextView

Id = textView1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Top = true

Layout Center Horizontal = true

Layout Margin Top = 20dp

Text = "Cipher Text"

Text Appearance = ?android:attr/Text AppearanceLarge

Figure 23-4.2.1: Third Activity Screenshot

EditText

Id = editText1

Layout Width = match parent

Layout Height = match parent

Layout Below = textView1

Layout Above = button1

Layout Center Horizontal = true

Layout Margin = 15dp

Clickable = false

Cursor Visible = false

ems = 10

Focusable = false

Focusable In Touch Mode = false

Gravity = top

Long Clickable = false

Button

Id = button1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Bottom = true

Layout Align Parent Left = true

Layout Margin Bottom = 20dp

Layout Margin Left = 20dp

On Click = onClick

Text = "Back"

Width = 110dp

Button

Id = button2

Layout Width = wrap content

Layout Height = wrap content

Layout Align Bottom = button1

Layout Align Parent Right = true

Layout Margin Right = 20dp

On Click = onClick

Text = "Next"

Width = 110dp

Fourth Activity (Encryption)

Layout Definition:

TextView

Id = textView1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Top = true

Layout Center Horizontal = true

Layout Margin Top = 15dp

Layout Margin Bottom = 10dp

Text Appearance = ?android:attr/Text AppearanceLarge

Gallery

Id = gallery1

Layout Width = match parent

Layout Height = wrap content

Layout Below = textView1

Fading Edge Length = 50dp

Figure 24-4.2.1: Fourth Activity Screenshot

ImageSwitcher

Id = switcher1

Layout Width = match parent

Layout Height = match parent

Layout Above = button1

Layout Below = gallery1

Layout Margin = 15dp

Layout Center Horizontal = true

Button

Id = button1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Bottom = true

Layout Align Parent Left = true

Layout Margin Bottom = 20dp

Layout Margin Left = 20dp

On Click = onClick

Text = "Back"

Width = 110dp

Button

Id = button2

Layout Width = wrap content

Layout Height = wrap content

Layout Align Bottom = button1

Layout Align Parent Right = true

Layout Margin Right = 20dp

On Click = onClick

Text = "Next"

Width = 110dp

Fifth Activity (Encryption)

Layout Definition

TextView

Id = textView1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Top = true

Layout Center Horizontal = true

Layout Margin Top = 20dp

Gravity = center horizontal

Text = "Message Encrypted Image (Output)"

Text Appearance = ?android:attr/Text AppearanceLarge

Figure 25-4.2.1: Fifth Activity Screenshot

ImageView

Id = imageView1

Layout Width = match parent

Layout Height = match parent

Layout Above = button1

Layout Below = textView1

Layout Margin = 15dp

Layout Center Horizontal = true

Layout Center Vertical = true

Button

Id = button1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Bottom = true

Layout Align Parent Left = true

Layout Margin Bottom = 20dp

Layout Margin Left = 20dp

On Click = onClick

Text = "Back"

Width = 110dp

Button

Id = button2

Layout Width = wrap content

Layout Height = wrap content

Layout Align Bottom = button1

Layout Align Parent Right = true

Layout Margin Right = 20dp

On Click = onClick

Text = "Home"

Width = 110dp

Sixth Activity (Decryption)

Layout Definition:

TextView

Id = textView1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Top = true

Layout Center Horizontal = true

Layout Margin Top = 15dp

Layout Margin Bottom = 10dp

Text = "Image Selection"

Text Appearance = ?android:attr/Text AppearanceLarge

Gallery

Id = gallery1

Layout Width = match parent

Layout Height = wrap content

Layout Below = textView1

Fading Edge Length = 50dp

Figure 26-4.2.1: Sixth Activity Screenshot

ImageSwitcher

Id = switcher1

Layout Width = match parent

Layout Height = match parent

Layout Above = button1

Layout Below = gallery1

Layout Margin = 15dp

Layout Center Horizontal = true

Button

Id = button1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Bottom = true

Layout Align Parent Left = true

Layout Margin Bottom = 20dp

Layout Margin Left = 20dp

On Click = onClick

Text = "Back"

Width = 110dp

Button

Id = button2

Layout Width = wrap content

Layout Height = wrap content

Layout Align Bottom = button1

Layout Align Parent Right = true

Layout Margin Right = 20dp

On Click = onClick

Text = "Next"

Width = 110dp

Seventh Activity (Decryption)

Layout Definition:

TextView

Id = textView1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Top = true

Layout Center Horizontal = true

Layout Margin Top = 20dp

Text = "Cipher Text"

Text Appearance = ?android:attr/Text AppearanceLarge

Figure 27-4.2.1: Seventh Activity Screenshot

EditText

Id = editText1

Layout Width = match parent

Layout Height = match parent

Layout Below = textView1

Layout Above = button1

Layout Center Horizontal = true

Layout Margin = 15dp

Clickable = false

Cursor Visible = false

ems = 10

Focusable = false

Focusable In Touch Mode = false

Gravity = top

Long Clickable = false

Button

Id = button1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Bottom = true

Layout Align Parent Left = true

Layout Margin Bottom = 20dp

Layout Margin Left = 20dp

On Click = onClick

Text = "Back"

Width = 110dp

Button

Id = button2

Layout Width = wrap content

Layout Height = wrap content

Layout Align Bottom = button1

Layout Align Parent Right = true

Layout Margin Right = 20dp

On Click = onClick

Text = "Next"

Width = 110dp

Third Activity (Decryption)

Layout Definition:

TextView

Id = textView1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Top = true

Layout Center Horizontal = true

Layout Margin Top = 20dp

Text = "Hidden Message"

Text Appearance = ?android:attr/Text AppearanceLarge

Figure 28-4.2.1: Eighth Activity Screenshot

EditText

Id = editText1

Layout Width = match parent

Layout Height = match parent

Layout Below = textView1

Layout Above = button1

Layout Center Horizontal = true

Layout Margin = 15dp

Clickable = false

Cursor Visible = false

ems = 10

Focusable = false

Focusable In Touch Mode = false

Gravity = top

Long Clickable = false

Button

Id = button1

Layout Width = wrap content

Layout Height = wrap content

Layout Align Parent Bottom = true

Layout Align Parent Left = true

Layout Margin Bottom = 20dp

Layout Margin Left = 20dp

On Click = onClick

Text = "Back"

Width = 110dp

Button

Id = button2

Layout Width = wrap content

Layout Height = wrap content

Layout Align Bottom = button1

Layout Align Parent Right = true

Layout Margin Right = 20dp

On Click = onClick

Text = "Home"

Width = 110dp

4.2.2 Graphic User Interface (GUI) Definition

First Activity (Home Screen)

The First Activity is the Home screen of the application it will navigate to Encryption, Decryption and Exit if touched on the button.

Second Activity (Encryption)

The Second activity is the first activity that will appear when user want to encrypt a message. This activity is used to gather the information on password (user can choose to used random generated password to enhance the security) and also allowed user to enter the message to be encrypted.

The password and message of the user will eventually passed to Third activity when completed. Third activity will process the password and message into cipher text using AES encryption.

Third Activity (Encryption)

The third activity is designed to perform cryptography using AES encryption method. The successful created cipher text displaying on the text pane will be show to the user how successful the cryptography in the process.

The cipher text is then passed to the next activity to be further processed.

Fourth Activity (Encryption)

The fourth activity is to prompt the user with all the images contained in the mobile devices. User is allowed to slide through the images to select a suitable image for the encryption.

The selected image path will be passed to Fifth activity so that the activity can obtain the image information and perform watermark encryption.

Fifth Activity (Encryption)

The fifth activity will perform steganography using watermarking method. The information of the image is obtained and cipher text will be converted into more process efficient form which is in binary state and embedded into the image pixels by pixels.

The successful message embedded image will be save into the image directory automatically so that user can send it to the recipient to be decrypted.

After Sucessfully encrypt the message into the image user can choose to go back to the First Activity which would be the Home screen by touching the “Home” button.

Sixth Activity (Decryption)

The sixth Activity is visually the same with Fourth Activity and it will also prompt user with the image contained in the mobile devices. However this time user need to choose the image which have message embedded inside.

After choosing the right image and before user gets into the next activity a toast will pops up which required user to enter the password to encrypt the message (note that user will only be allowed to proceed if the password is correctly entered).

When correct password is entered and "OK" button is touched user will be allowed to proceed to the Seventh Activity.

Seventh Activity (Decryption)

The Seventh Activity is the same with Third Activity as it will also show user the cipher text. However the cipher text is obtained from the encrypted image.

To view the real message user need to touch on the "Next" button to proceed to Eighth Activity.

Eighth Activity (Decryption)

The Eight Activity is to show user the real message hidden behind the image.

After reading the message user can choose to go back to the First Activity which would be the Home screen by touching the "Home" button.

4.3 Simulations

4.3.1 Simulation Set 1: Multilanguage Support

For this set of simulation the ability of the application to support different kind of input. Input such as numbers, alphabets, symbols, Chinese characters and Korean characters is tested. The simulation set is carried out to obtain the correct output after the process of encryption and decryption. The test case is listed in the table shown below:

Password	Message
1234	0904872
abcd	English
(:-)	@#$/
一二三四	华语
뭊ㅇ	한국어
1234abcd(:-)一二三四뭊ㅇ	0904872 English @#$/ 华语 한국어

Table 5-4.3.1: Table of Simulation Set 1 Test Case

4.3.2 Simulation Set 2: Message Maximum & Minimum Length Approach

Meanwhile, a longest message which consist of 480 characters entered to test the ability of the application in handling the maximum. To be fair a random password is used when testing for the maximum. As for the minimum message and password editable is leaved blank to test the ability of the application in handling the minimum.

Password	Message	Test Component
		Minimum
Random Password: jduckzo0kimgz5ij	The main perseverance of this Final Year project is in development and analysis message embedding system for embedded OS Using Spatial Watermarking Technique. There is plenty of Android application are available currently are mainly for entertainment purposes thus users Android phone is lack of security protection in terms of privacy, indeed they does not know how important to have an application to secure their message which may be very important. This is the longest message can be typed.	Maximum

Table 6-4.3.2: Table of Simulation Set 2 Test Case

4.3.3 Simulation Set 3: Wrong Password Handling

A series of password which is different from the real password is entered to test the accuracy of the application in determine the correct password. The test case is listed in the table shown below:

Correct password: UTAR Kampar (拉曼大學)

Series of Wrong Password entered	Test Component
UTAr Kampar (拉曼大學)	Case Sensitivity
UTAR Kampar {拉曼大學)	Similar Symbol
UTAR Kampar (拉曼太學)	Chinese Character Sensitivity
UTAR Kampar(拉曼大學)	Lack of spacing
UTAR Kampar (拉曼大學)	Addition spacing
UTAR Kampar (拉曼 大學)	Addition spacing between Chinese Character
(拉曼大學) UTAR Kampar	Password Orientation
UTAR Kampar (拉曼大學)	Original Password

Table 7-4.3.3: Table of Simulation Set 3 Test Case

4.4 Simulation Result

4.4.1 Simulation Set 1 Result: Multilanguage Support

Password	Message
1234	0904872

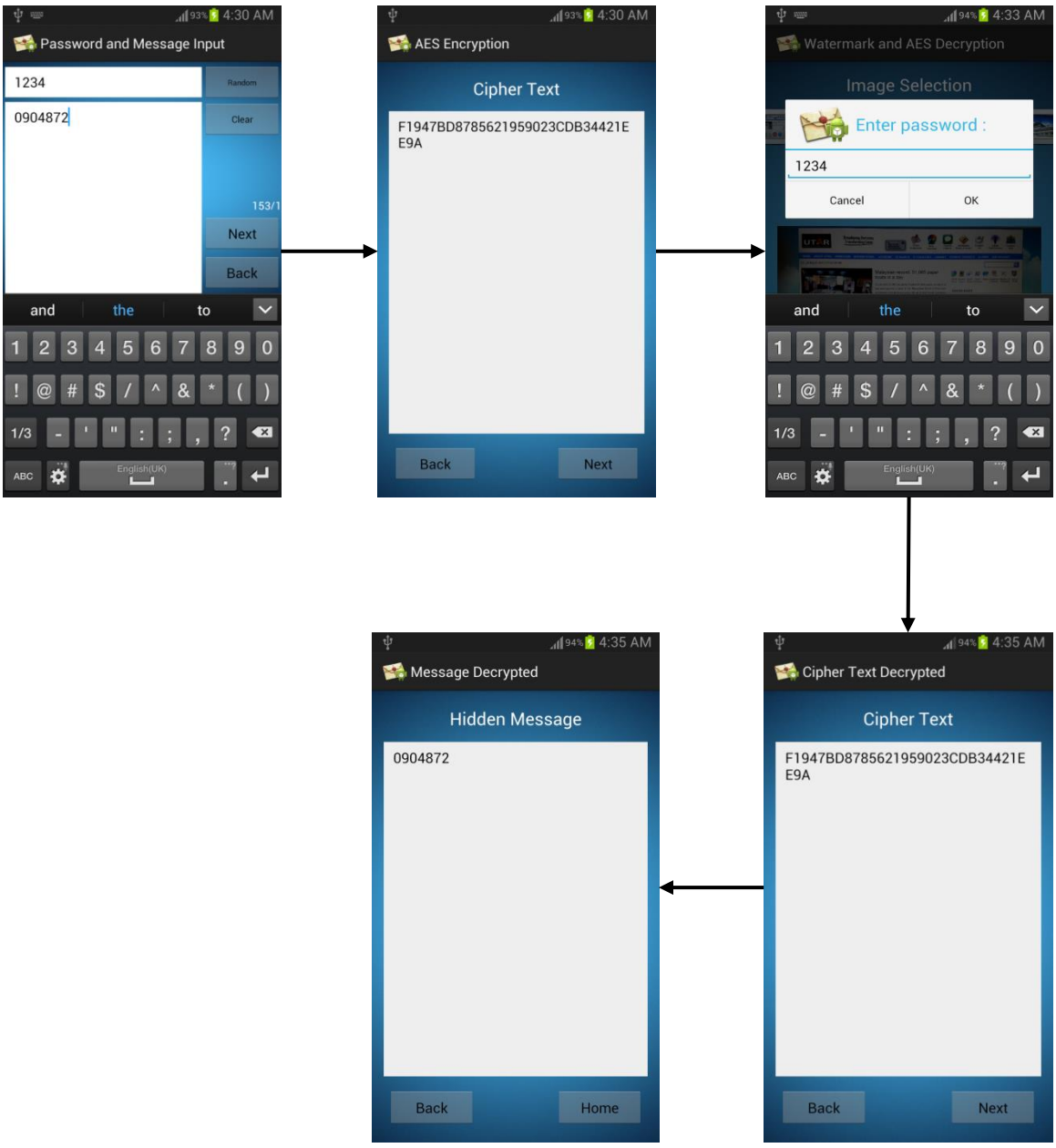

Figure 29-4.4.1: Multilanguage Support (Number) Screenshot Flow

Password	Message
abcd	English

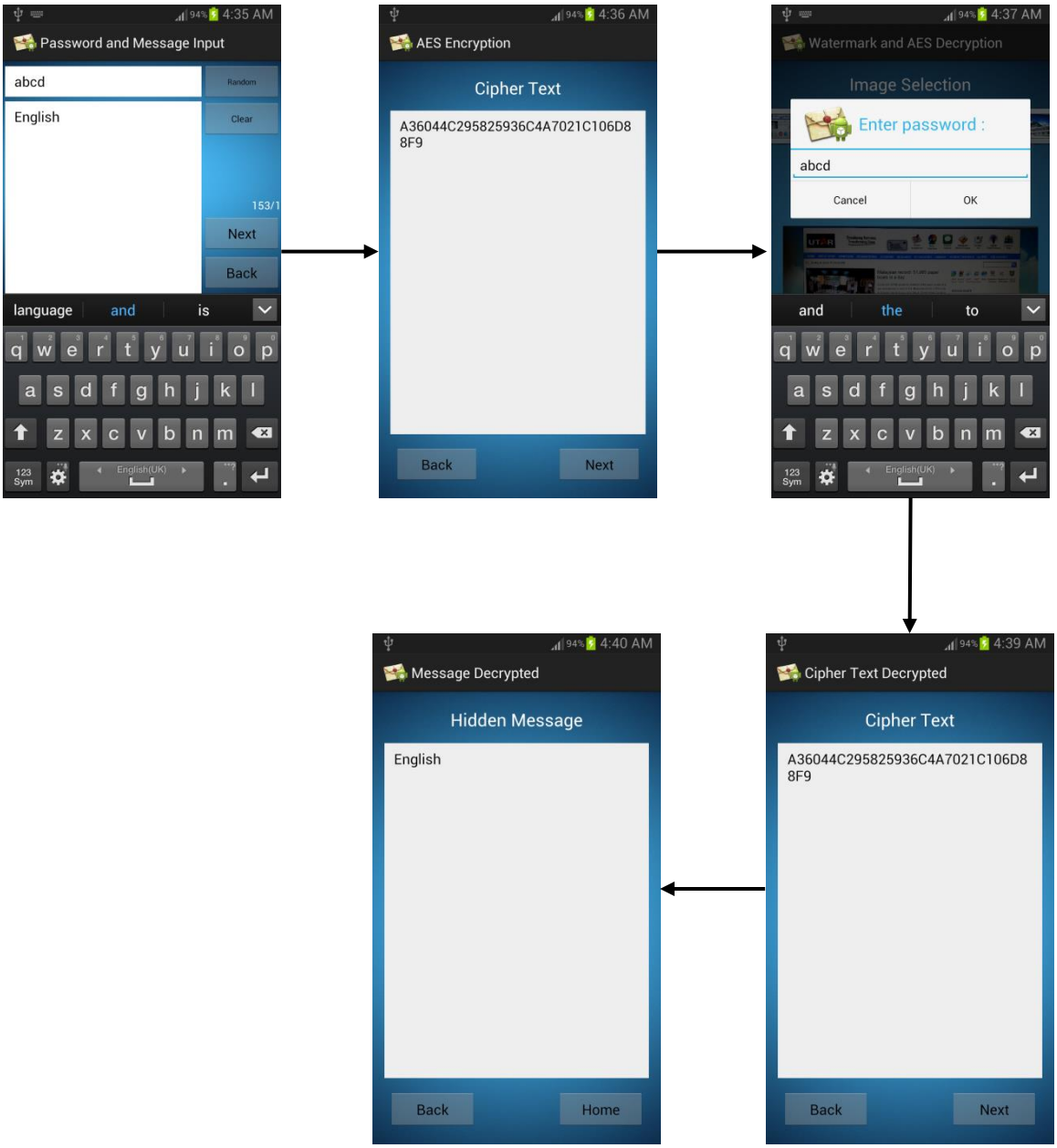

Figure 30-4.4.1: Multilanguage Support (Alphabet) Screenshot Flow

Password	Message
(:-)	@#$/

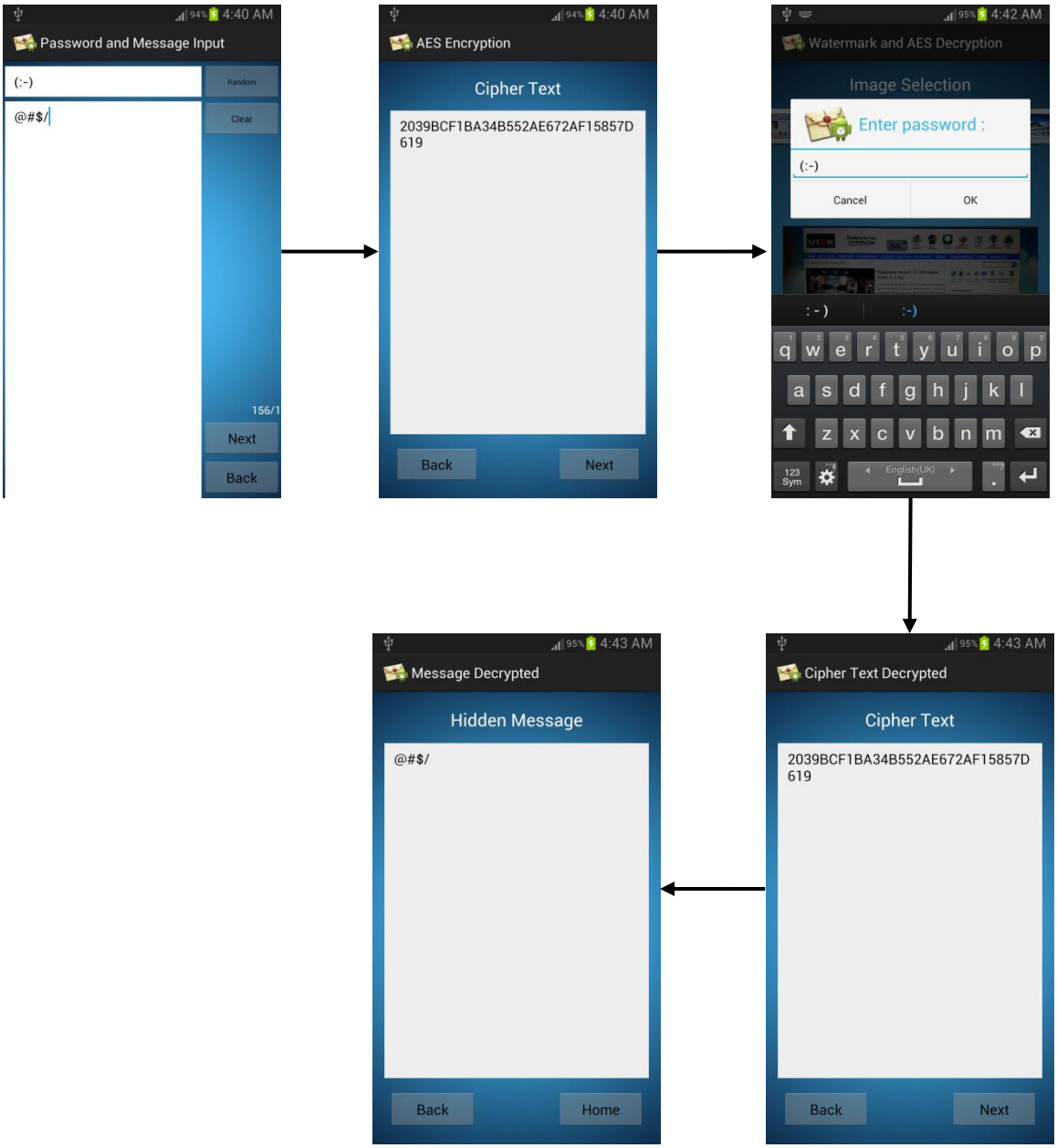

Figure 31-4.4.1: Multilanguage Support (Symbols) Screenshot Flow

Password	Message
一二三四	华语

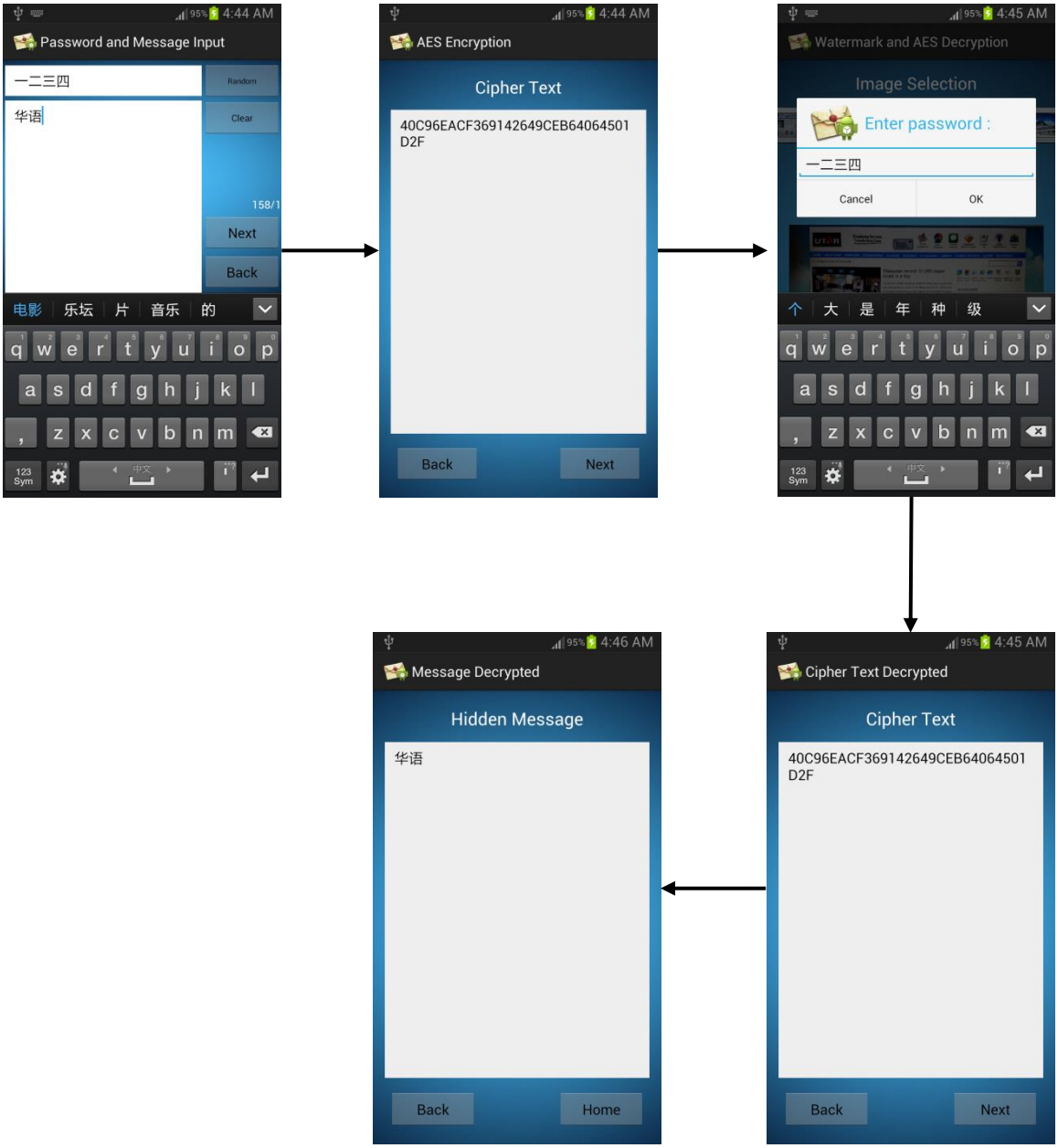

Figure 32-4.4.1: Multilanguage Support (Chinese character) Screenshot Flow

Password	Message
뭊ㅇ	한국어

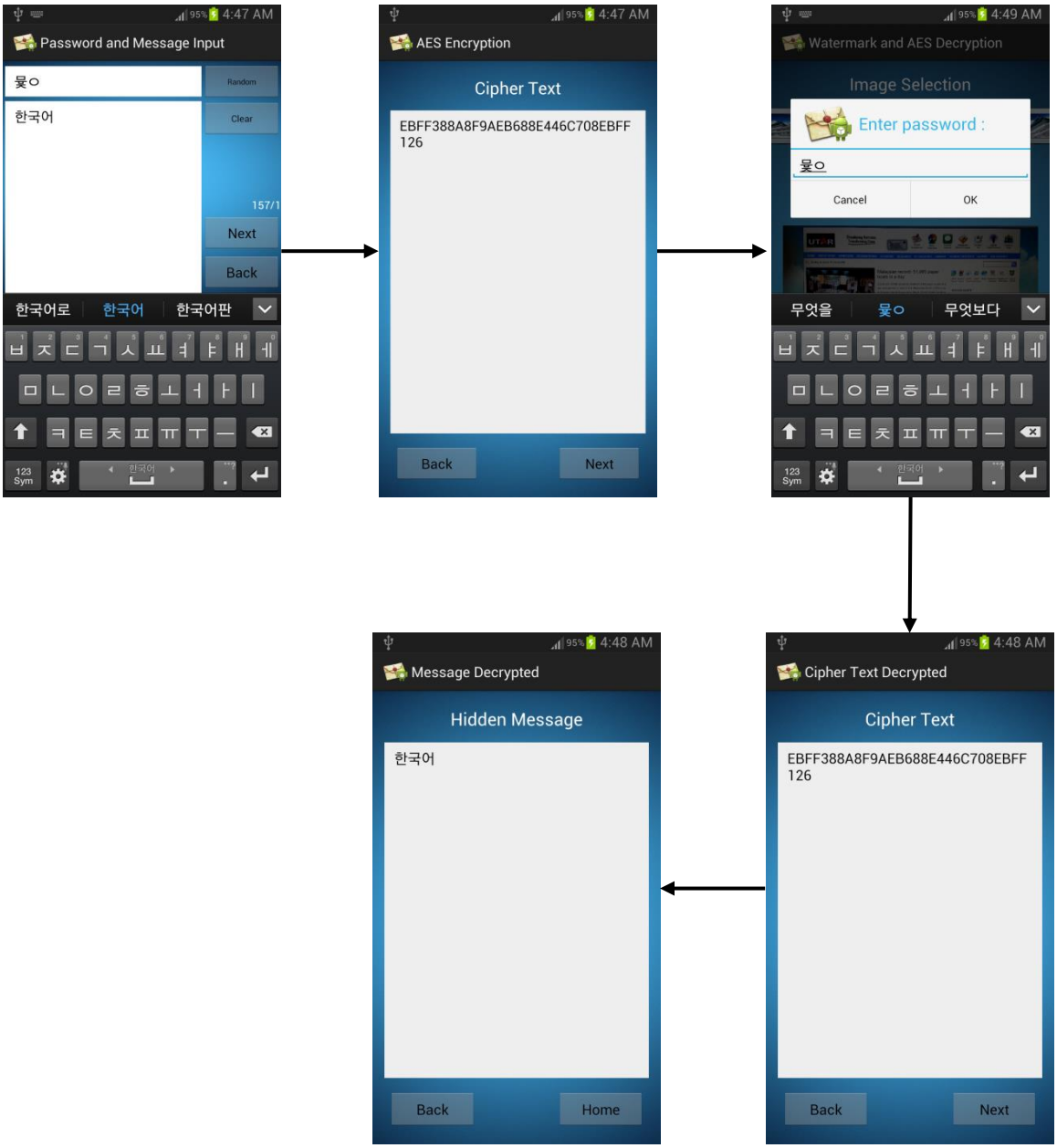

Figure 33-4.4.1: Multilanguage Support (Korean character) Screenshot Flow

Password	Message
1234abcd(:-)一二三四뮺ㅇ	0904872 English @#$/ 华语 한국어

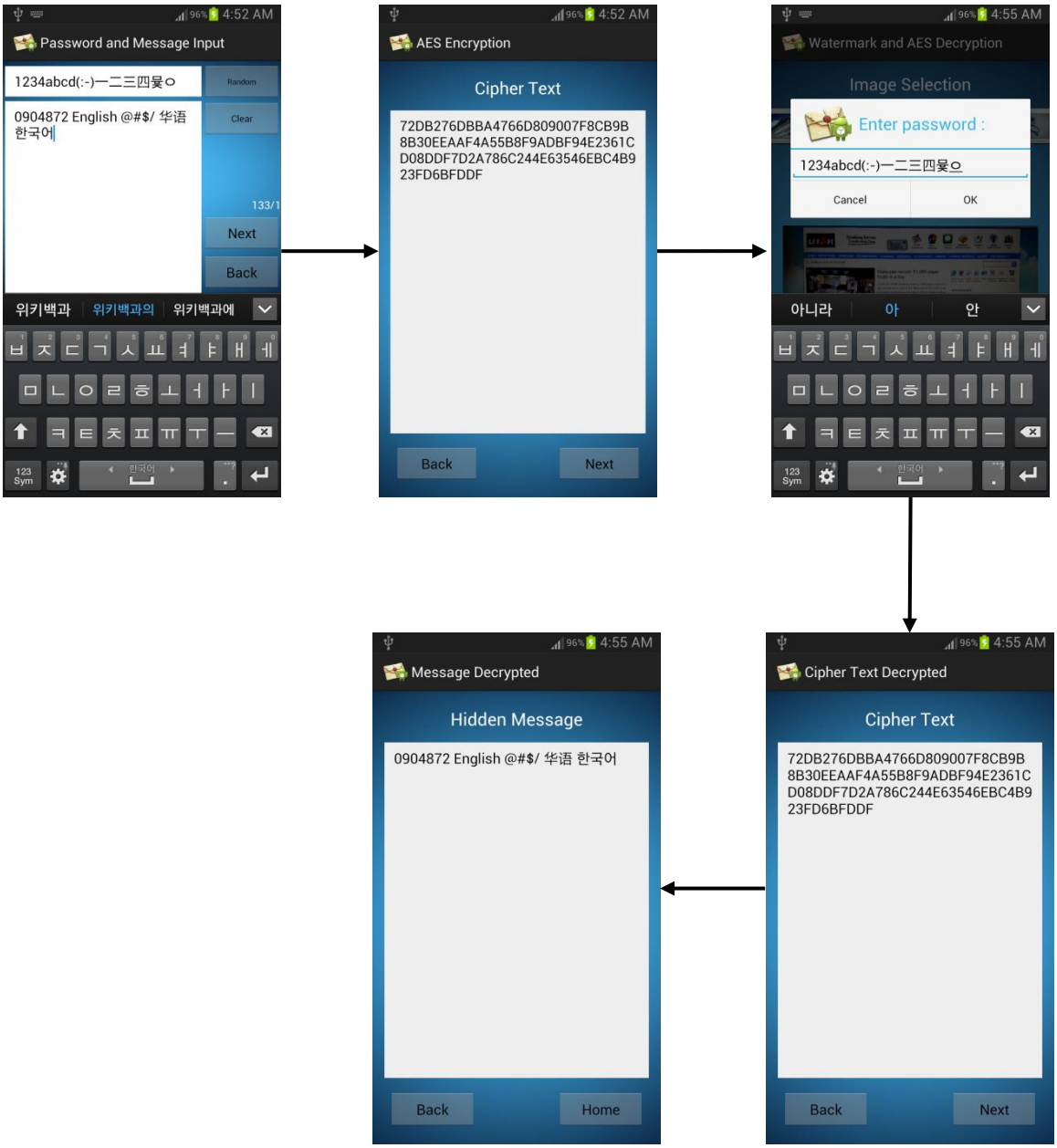

Figure 34-4.4.1: Multilanguage Support (Mixed) Screenshot Flow

4.4.2 Simulation Set 2 Result: Message Maximum & Minimum Length Approach

Password	Message	Test Component
		Minimum

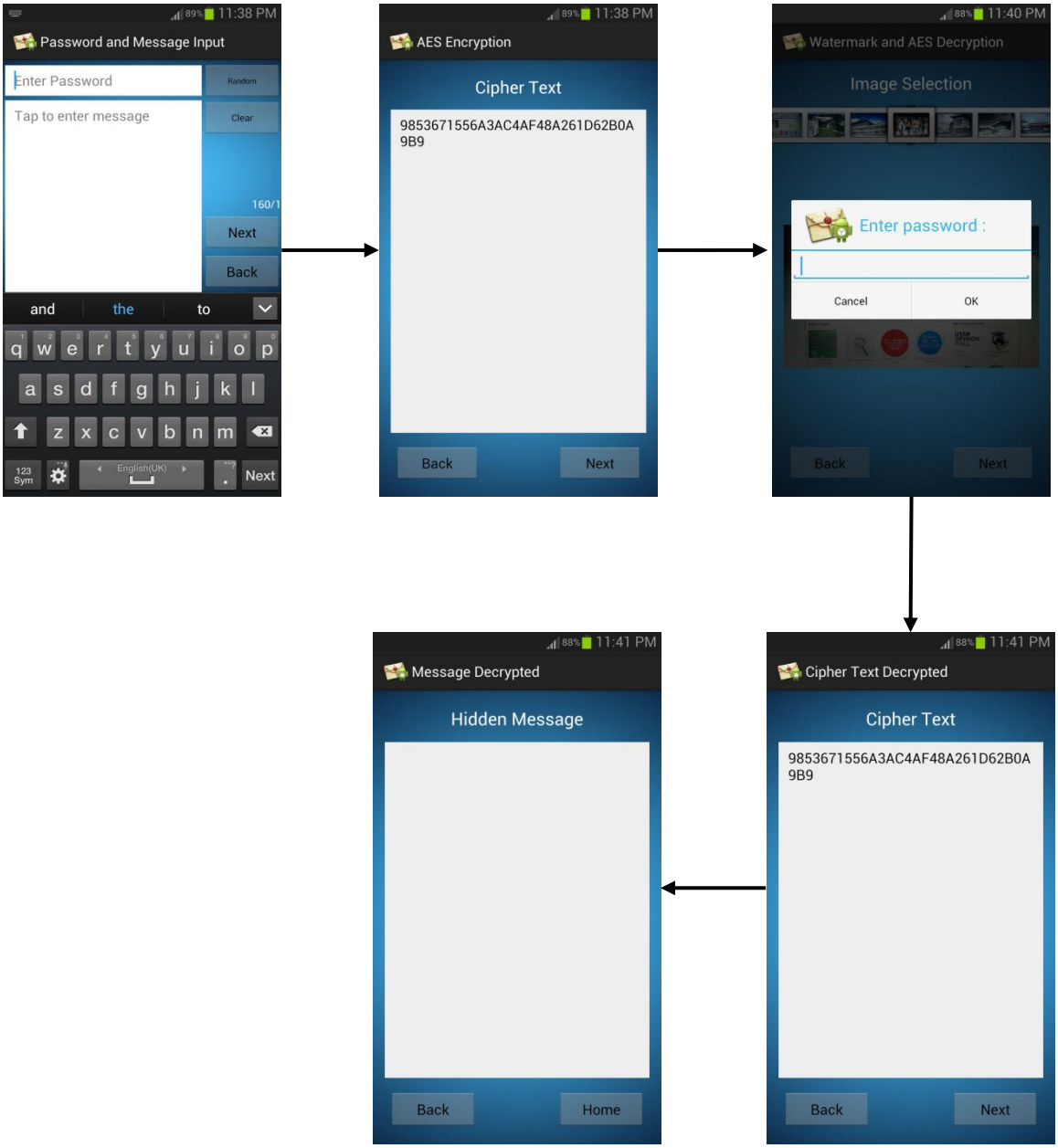

Figure 35-4.4.2: Message Minimum Length Approach Screenshot Flow

Password	Message	Test Component
Random Password: jduckzo0kimgz5ij	The main perseverance of this Final Year project is in development and analysis message embedding system for embedded OS Using Spatial Watermarking Technique. There is plenty of Android application are available currently are mainly for entertainment purposes thus users Android phone is lack of security protection in terms of privacy, indeed they does not know how important to have an application to secure their message which may be very important. This is the longest message can be typed.	Maximum

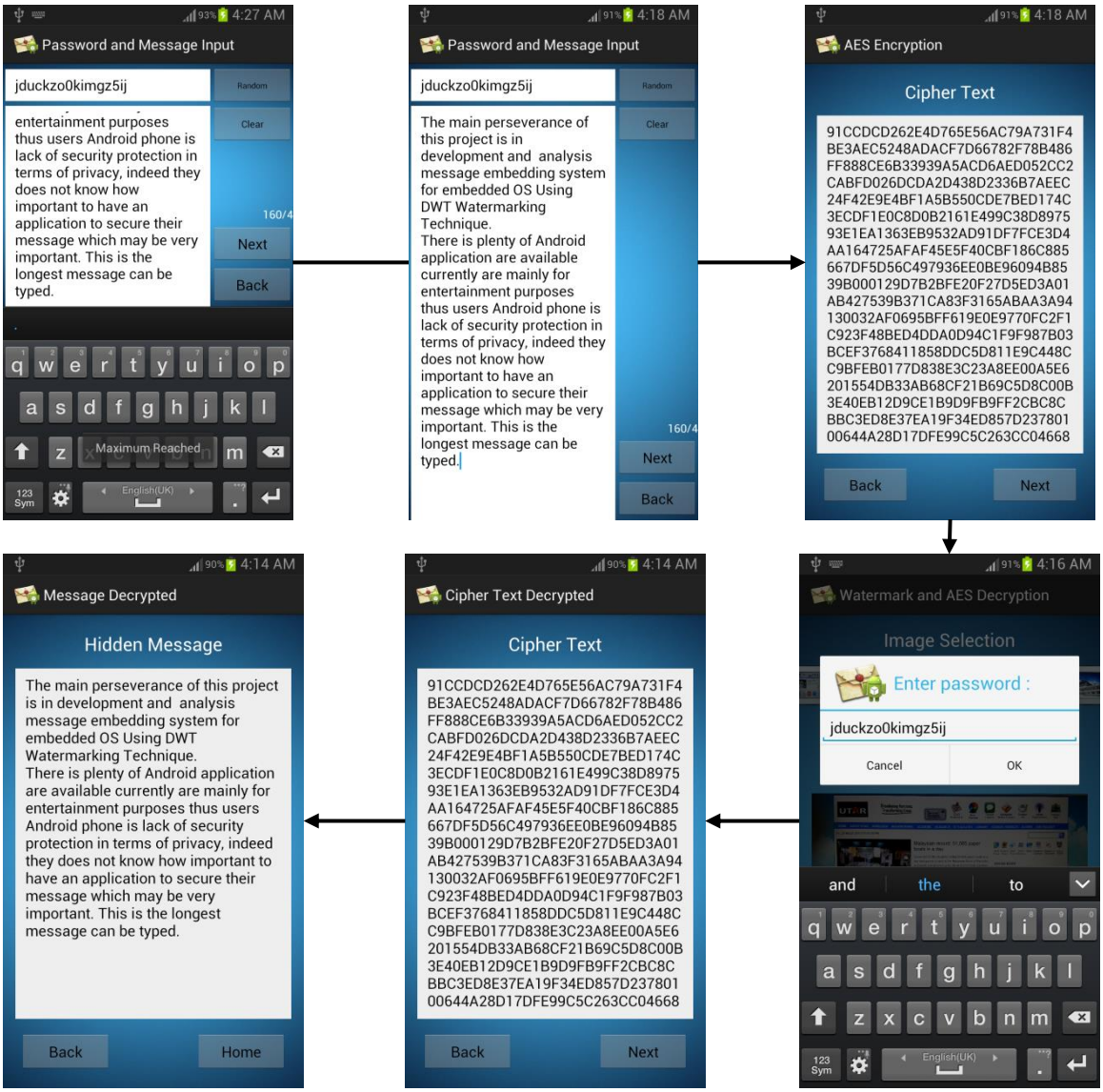

Figure 36-4.4.2: Message Maximum Length Approach Screenshot Flow

4.4.3 Simulation Set 3 Result: Wrong Password Handling

Series of Wrong Password entered	Test Component
UTAr Kampar (拉曼大學)	Case Sensitivity

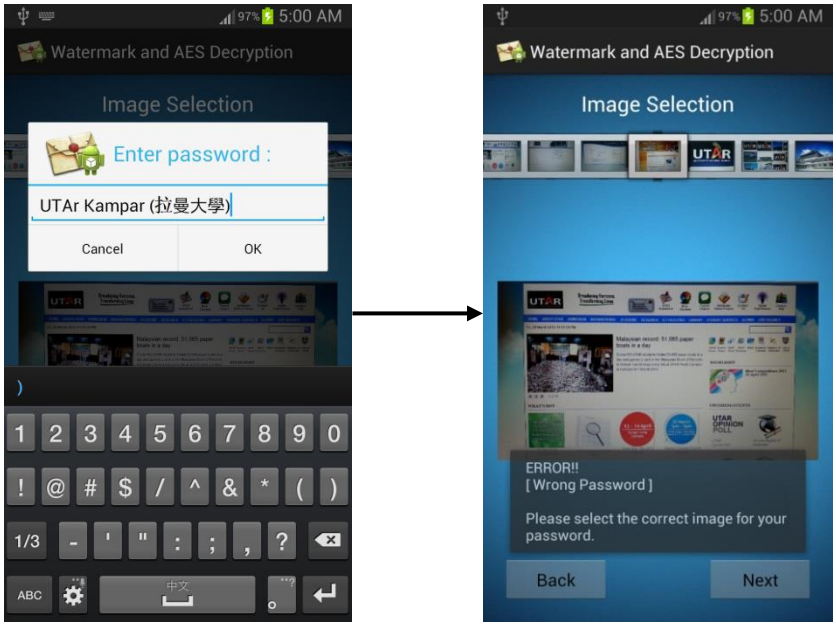

Figure 37-4.4.3: Wrong Password Handling (Case Sensitivity) Screenshot Flow

Series of Wrong Password entered	Test Component
UTAR Kampar {拉曼大學)	Similar Symbol

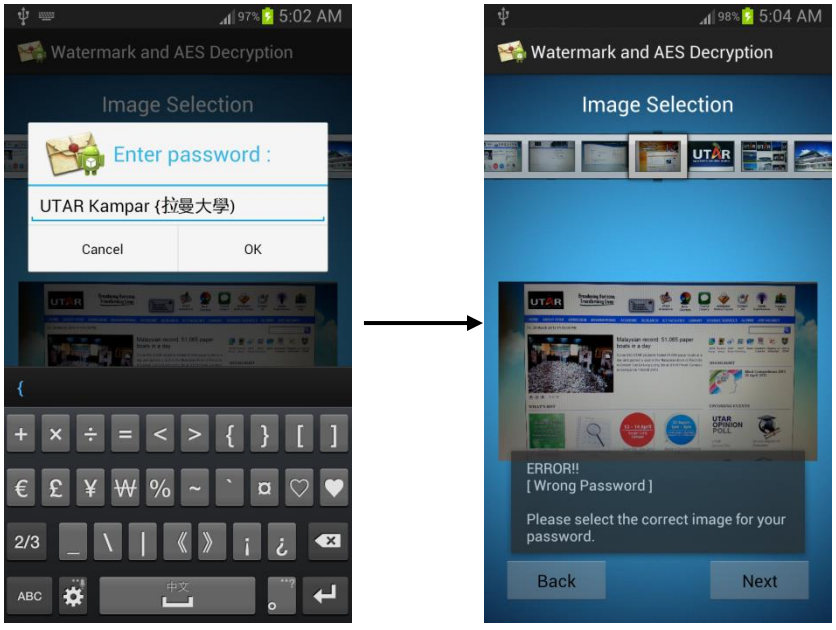

Figure 38-4.4.3: Wrong Password Handling (Similar Symbol) Screenshot Flow

Series of Wrong Password entered	Test Component
UTAR Kampar (拉曼太學)	Chinese Character Sensitivity

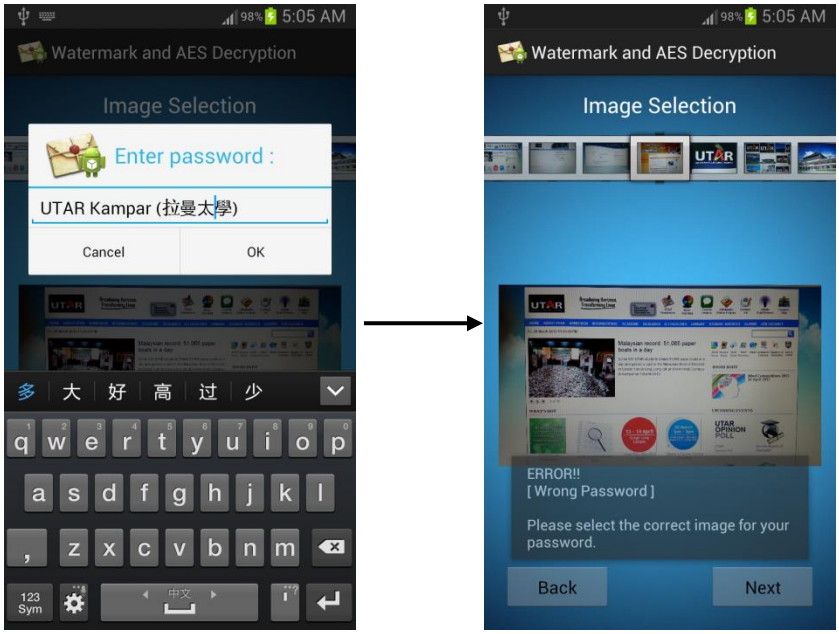

Figure 39-4.4.3: Wrong Password Handling (Chinese Character Sensitivity) Screenshot Flow

Series of Wrong Password entered	Test Component
UTAR Kampar(拉曼大學)	Lack of spacing

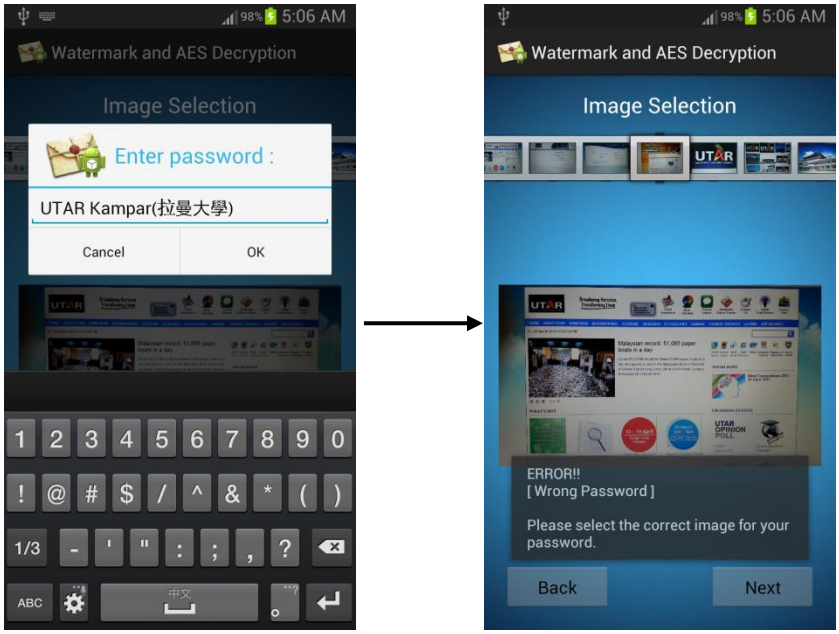

Figure 40-4.4.3: Wrong Password Handling (Lack of Spacing) Screenshot Flow

Series of Wrong Password entered	Test Component
UTAR Kampar (拉曼大學)	Addition spacing

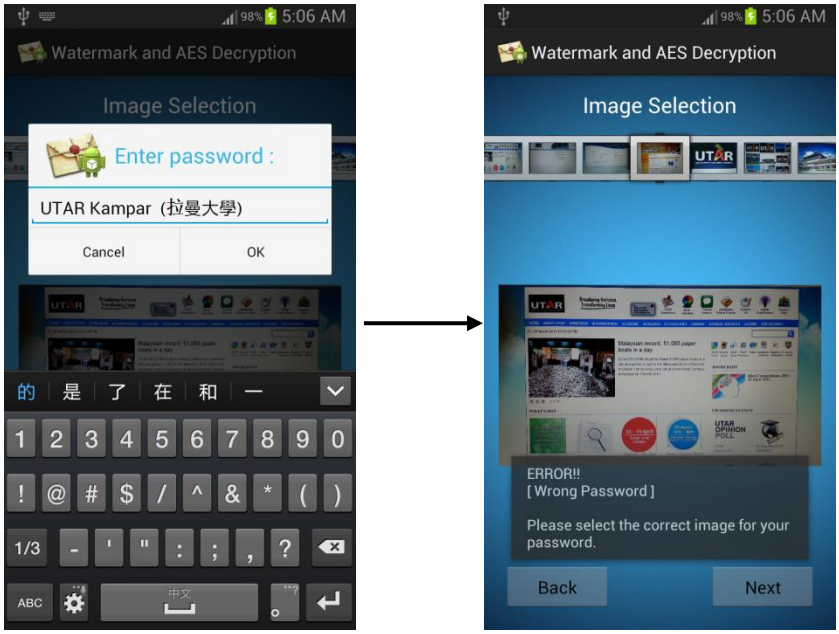

Figure 41-4.4.3: Wrong Password Handling (Addition Spacing) Screenshot Flow

Series of Wrong Password entered	Test Component
UTAR Kampar (拉曼 大學)	Addition spacing between Chinese Character

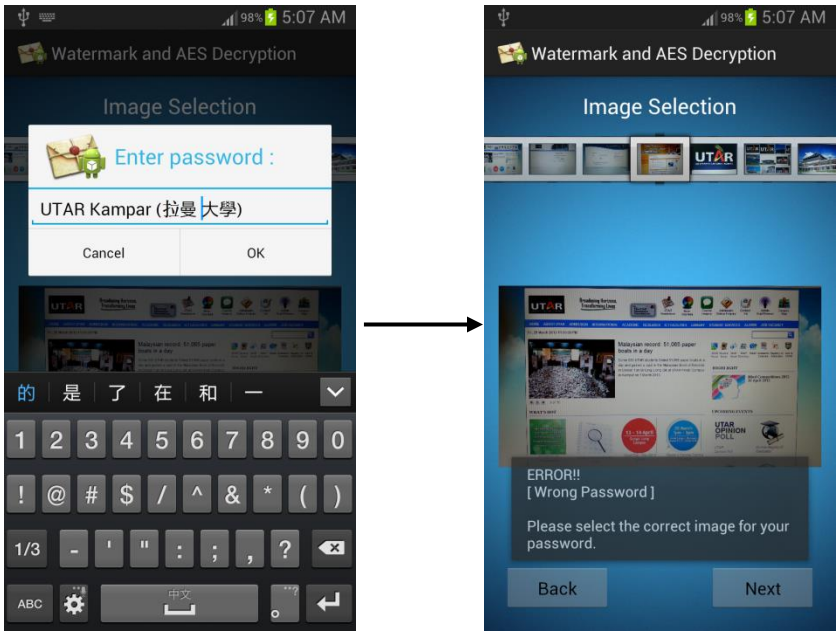

Figure 42-4.4.3: Wrong Password Handling (Addition Spacing between Chinese Character) Screenshot Flow

Series of Wrong Password entered	Test Component
(拉曼大學) UTAR Kampar	Password Orientation

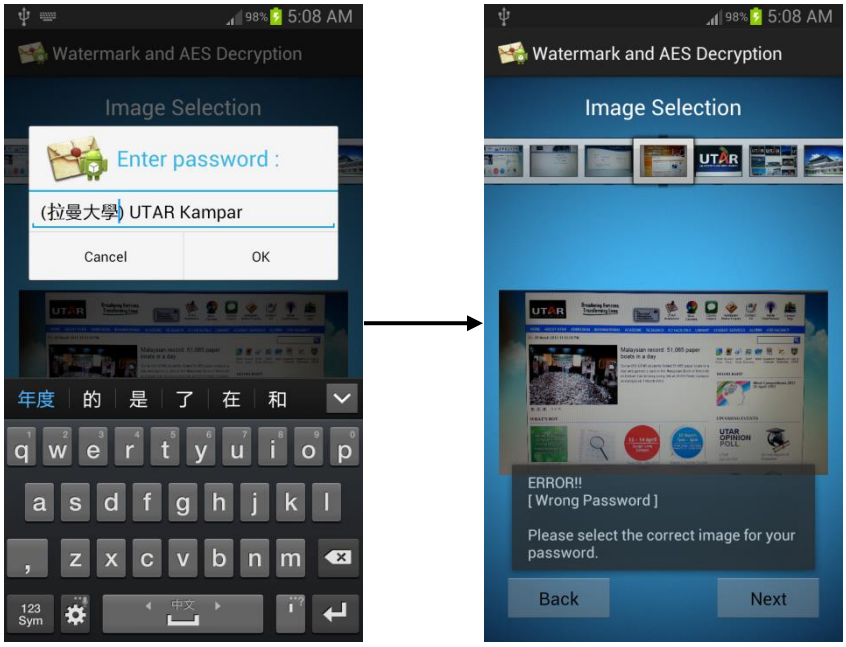

Figure 43-4.4.3: Wrong Password Handling (Password Orientation) Screenshot Flow

Series of Wrong Password entered	Test Component
UTAR Kampar (拉曼大學)	Original Password

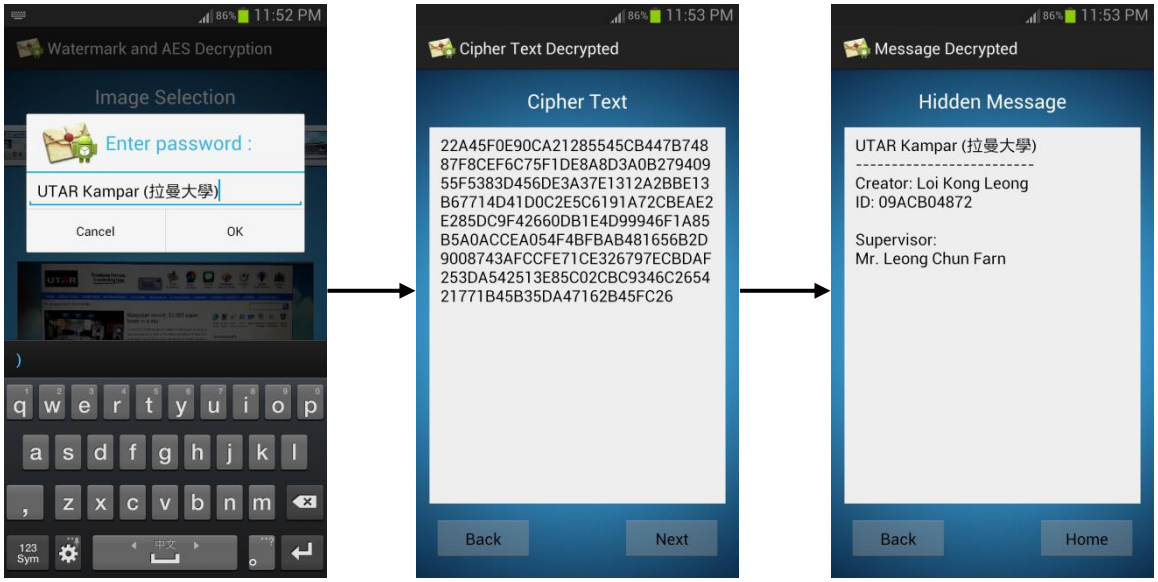

Figure 44-4.4.3: Wrong Password Handling (Original Password) Screenshot Flow

4.5 Discussion

4.5.1 Simulation set 1

Simulation set 1 is meant to test the Multilanguage support and the encryption of different kind of characters. According to the results, all the test case successfully being encrypted and decrypted without any errors or information lost. Character number of the generated cipher text has no significant difference for the message with numbers, alphabets, symbols, Chinese characters and Korean characters. All kinds of character has being threated the same by the AES encryption method implemented.

However, the character number of the generated cipher text increases when the character number of message increased. More data to be encrypted longer cipher text is used to secure the message but the character number of cipher text is not directly proportional with the character number of the message.

To conclude, there is no limitation for the application in recognizing different type of character and languages. Data is encrypted correctly using the AES encryption method and can successfully embedded into the image. Data can be retrieved through decryption process without any faulty or lose of information due to encryption and embedding.

4.5.2 Simulation set 2

Simulation set 2 is to test how the application handle the message maximum and minimum length. For the minimum message length, the application does not produce error message or empty cipher text but to output a series of 32 hexadecimal characters which represents the null password and null message. The 32 hexadecimal characters cipher text successfully embedded into the image and the null message is successfully decrypted from the image by using null as password (by leaving the password editText empty).

As for the maximum message length, the application produced a total length of 992 cipher text. The cipher text encrypted and embedded into the image successfully and decrypted with no problem occurs.

From the simulation done it is proven that the application does not restrict any kind of input as well as the ways of input.

However, there is restriction set for the user to not entering more than 480 characters of message. This is to avoid overflow during embedding whereby it occurs when a small sized image that has limited number of pixels is used in embedding. When the data is longer than the number of pixels overflow will occur and this will cause the information to be lost. Information lost causes incomplete cipher text encrypted into the image.

Hence, the message no longer can be decrypted using the original password. This will lead to failure in information transfer to the recipient.

With the restriction of number of characters allowed for the message can avoid the problem when user tends to crop the image into smaller size. This restriction allowed user to crop the image around 7000 pixels. For example 700 x 10 resolution and 100 x 70 resolution.

4.5.3 Simulation set 3

Simulation set 3 is to test the password verification and wrong password handling of the application.

Series of Wrong Password entered	Test Component	Results
UTAr Kampar (拉曼大學)	Case Sensitivity	Fail
UTAR Kampar {拉曼大學)	Similar Symbol	Fail
UTAR Kampar (拉曼太學)	Chinese Character Sensitivity	Fail
UTAR Kampar(拉曼大學)	Lack of spacing	Fail
UTAR Kampar (拉曼大學)	Addition spacing	Fail
UTAR Kampar (拉曼 大學)	Addition spacing between Chinese Character	Fail
(拉曼大學) UTAR Kampar	Password Orientation	Fail
UTAR Kampar (拉曼大學)	Original Password	Success

Table 8-4.5.3: Table of Simulation Set 3 Result

The results show that the application only verify password that is exactly the same with the original password used to encrypt the message. Any misrepresentation of the password will not be tolerated by the application.

In AES encryption the key used to decrypt the message is unique that means the key used in decryption has to be the same with the key used in encryption . For any key that does not belongs to the cipher text used to decrypt the message will produce a null output (output with no string). This indicate the decryption is unsuccessful.

Hence, the application will produce an error message through toast if the password is not exactly the same with the password used in encryption.

Chapter 5: Conclusion and Future Work

This paper presented a Development and Analysis of Message Embedding System for Embedded OS Using Spatial Watermarking Technique. Due to many intelligent intruders or more likely to be known as hackers that able to acquire any information from the network because of the vulnerability of the network security. This application can increase a normal security of message at certain satisfactory level. This technique is very reliable as it's encryption level is more than usual encryption. Not one method is implemented but two with password enabled surely a better design and it is suitable to be used by anyone with android mobile devices.

However, for professional hacker exist until today it is still possible for them to hack this application. More and more encryption are needed to be done to increase the security of the message or to disrupt the hackers from obtaining the real message.

In the future, more advanced android message security application can be created by applying additional encryption and more advance encryption method before a fully secured method of encryption method were introduced. Additional encryption can increase time for hackers to decrypt the message as different type of encryption need different ways to decrypt. Advanced encryption method can provides more hard time for hackers to decrypt the message as a unique way is required to decrypt the message. In addition, the application can be further improved to run in more different mobile devices as well as faster respond time.

Bibliography/References

Smith, A . 2011, *How Americans Use Text Messaging*. 19 September 2011. Aaron Smith: *Pew Internet & American Life Project* . Available from: <http://pewinternet.org/Reports/2011/Cell-Phone-Texting-2011/Main-Report.aspx/> [19 June 2012].

Kelly, T. 2012 '*Internet firms can access your texts, emails and pictures by spying through smartphone apps'* Dailymail.co.uk 27 February 2012. Available from: <http://www.dailymail.co.uk/sciencetech/article-2106627/Internet-firms-access-texts-emails-pictures-spying-smartphone-apps.html/> [18 June 2012].

SecretSMSReplicator: Spy On Text Messages (Android), 30 October 2010. Available from: <http://www.makeuseof.com/dir/secretsmsreplicator-secretly-receive-sms-messages-android-phone/> [18 June 2012].

Stallings, W 2011, *Cryptography and Network Security*. 5th (ed.) , Prentice Hall, United States of America, pp.5.

Steganography - Wikipedia, the free encyclopedia 2013. Available from: <http://en.wikipedia.org/wiki/Steganography/> [28 March 2013].

Cryptography - Wikipedia, the free encyclopedia 2013. Available from: <http://en.wikipedia.org/wiki/Cryptography#cite_note-26/> [28 March 2013].

Globusonline.org (n.d.), *Security Primer*. Available from: <https://www.globusonline.org/media/publications/The_Resource_Providers_Guide_to_Globus_Online/security.html/> [2 July 2012].

Rouse, M. 2012, *What is Advanced Encryption Standard (AES)? - Definition from WhatIs.com*. Available from: <http://searchsecurity.techtarget.com/definition/Advanced-Encryption-Standard/> [25 June 2012].

Kamali, SH, Hedayati, M, Shakerian R & Rahmani, M 2010, '*A New Modified Version of Advanced Encryption Standard Based Algorithm for Image Encryption'*. Available from: <http://ieeexplore.ieee.org.libezp.utar.edu.my/stamp/stamp.jsp?tp=&arnumber=5559902/> [25 June 2012].

jamesedwardtracy 2010, 'Advanced Encryption Standard ', 15 January 2010 at 11:28, Available from: <http://www.jamesedwardtracy.com/WWW/AES/AES-WIKI.pdf/> [25 June 2012].

Computer Security Division (CSD) Computer Security Resource Center (CSC) 2001, *Announcing the ADVANCED ENCRYPTION STANDARD (AES)* . Available from: <http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf/> [25 June 2012].

Feistel cipher - Wikipedia, the free encyclopedia 2013. Available from: <http://en.wikipedia.org/wiki/Feistel_cipher/> [25 March 2013].

Satti, MVK 2007, '*Quasi Group Based Crypto-System'*. Available from: <http://etd.lsu.edu/docs/available/etd-10182007-182636/unrestricted/Thesisupdated.pdf/> [26 June 2012].

Brute-force attack - Wikipedia, the free encyclopedia 2013. Available from: <http://en.wikipedia.org/wiki/Brute-force_attack/> [28March 2013].

Cummis, J, Diskin, P, Lau, S & Parlett, R 2004, '*Steganography And Digital Watermarking'* . Available from: <http://www.cs.bham.ac.uk/~mdr/teaching/modules03/security/students/SS5/Steganography.pdf/> [25 June 2012].

Digital watermarking - Wikipedia, the free encyclopedia 2013. Available from: <http://en.wikipedia.org/wiki/Digital_watermarking/> [26 March 2013].

Cao, J, Li, A & Lv, G 2008, '*Study on Multiple Watermarking Scheme for GIS Vector Data 1'*. Available from: <http://ieeexplore.ieee.org.libezp.utar.edu.my/stamp/stamp.jsp?tp=&arnumber=5568201/> [9 July 2012].

LAWS OF MALAYSIA 2000, '*COPYRIGHT ACT 1987'*. Available from: <http://portal.psz.utm.my/psz/images/stories/2012/copyright_act_1987.pdf/> [9 July 2012].

Potdar, VM, Han, S & Chang, E 2012, '*Fingerprinted Secret Sharing Steganography for Robustness against Image Cropping Attacks'*. Available from: <http://debii.edu.au/~vidy/publications/INDIN_2005_Fingerprinted%20Secret%20Sharing%20Steganography%20for%20Robustness%20against%20Image%20Cropping%20Attacks.pdf/> [14 July 2012].

Fingerprint (computing) - Wikipedia, the free encyclopedia 2013. Available from: <http://en.wikipedia.org/wiki/Fingerprint_(computing)/> [26 March 2013].

Mahmoud, H, Al-Hulaibah, HS, Al-Naeem, SA, Al-Qhatani, SA, Al-Dawood, A, Al-Nassar, B & Al-Salman, DY 2010, '*Novel Technique for Steganography in Fingerprints Images: Design and Implementation'*. Available from: <http://ieeexplore.ieee.org.libezp.utar.edu.my/stamp/stamp.jsp?tp=&arnumber=5604078/> [26June 2012].

Sarmah, DK & Bajpai, N 2009, '*Proposed System for data hiding using Cryptography and Steganography'*. Available from: <http://arxiv.org/ftp/arxiv/papers/1009/1009.2826.pdf/> [30 June 2012].

Rao, BR, Kumar, PA, Rao, KRM & Nagu, M 2012, '*A Novel Information Security Scheme using Cryptic Steganography'* . Available from: <http://www.ijcse.com/docs/IJCSE10-01-04-37.pdf/> [26 June 2012].

GSM - Wikipedia, the free encyclopedia 2013. Available from: <http://en.wikipedia.org/wiki/GSM [30 March 2013].

Encryption - Wikipedia, the free encyclopedia 2013. Available from: <http://en.wikipedia.org/wiki/Encryption/> [28 March 2013].

Hsu, CT & Wu, JL 1999, 'Hidden Digital Watermarks in Images'. Available from: <http://ieeexplore.ieee.org.libezp.utar.edu.my/stamp/stamp.jsp?tp=&arnumber=736686/> [26 August 2012].

Morkel, T, Eloff, JHP & Olivier, MS (n.d.), 'AN OVERVIEW OF IMAGE STEGANOGRAPHY'. Available from: <http://icsa.cs.up.ac.za/issa/2005/Proceedings/Full/098_Article.pdf/> [26 August 2012].

APPENDIX A: BIWEEKLY REPORT

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no:** 1
Student Name & ID: Loi Kong Leong 09ACB04872	
Supervisor: Mr. Leong Chun Farn	
Project Title: Design and Analysis of Message Embedding System for Mobile Operating System Using Spatial Watermarking Technique	

1. WORK DONE

Learned on Android versions, its feature set and installed the eclipse and SDK for developing Android applications. Have successfully develop a the first simple application .

2. WORK TO BE DONE

Write and learn more application that maybe needed to build the project apps.

3. PROBLEMS ENCOUNTERED

Still not so familiar with writing a .xml

4. SELF EVALUATION OF THE PROGRESS

Up to pace, completed project flow chart with extra information from internet added.

X

Mr. Leong Chun Farn
Supervisor"s signature

X

Loi Kong Leong
Student"s signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no:** 2
Student Name & ID: Loi Kong Leong 09ACB04872	
Supervisor: Mr. Leong Chun Farn	
Project Title: Design and Analysis of Message Embedding System for Mobile Operating System Using Spatial Watermarking Technique	

1. WORK DONE

Completed an application which is very useful to the project called "Intents". This application capable in calling activities and able to navigate from one activity to another. This is important as the project use more than one activities.

2. WORK TO BE DONE

Try to link the activities with each others such as passing data to the target and obtaining the result from other activities. Study on the algorithms of DWT watermark encryption method.

3. PROBLEMS ENCOUNTERED

Encountered problems in making the intent activity linked.

4. SELF EVALUATION OF THE PROGRESS

Ought to read more about Android apps, lack of knowledge on how to apply the functions implementation.

X

Mr. Leong Chun Farn
Supervisor"s signature

X

Loi Kong Leong
Student"s signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no:** 3
Student Name & ID: Loi Kong Leong 09ACB04872	
Supervisor: Mr. Leong Chun Farn	
Project Title: Design and Analysis of Message Embedding System for Mobile Operating System Using Spatial Watermarking Technique	

1. WORK DONE

Successfully link the activities with each others by passing data to the target and obtaining the result from other activities. Learned on fragments but it is not very useful to the project.

2. WORK TO BE DONE

Get to know the Android user interface such as xml layout on button, textView, editText and imageView. Learn how to create the user interface via code for the layout.

3. PROBLEMS ENCOUNTERED

Not quite understand the algorithms in the scholars thesis although have read it for several times already.

4. SELF EVALUATION OF THE PROGRESS

Satisfactory, at least many resources were found and read at the mean time.

X

Mr. Leong Chun Farn
Supervisor"s signature

X

Loi Kong Leong
Student"s signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no:** 4
Student Name & ID: Loi Kong Leong 09ACB04872	
Supervisor: Mr. Leong Chun Farn	
Project Title: Design and Analysis of Message Embedding System for Mobile Operating System Using Spatial Watermarking Technique	

1. WORK DONE

Learned most of the Android user interface such as xml layout on button, textView, editText, imageView, imageButton, checkbox, toggleButton, radioButton and radioGroup. Able to create the user interface via code for the layout and handling view events.

2. WORK TO BE DONE

Get to know the auto complete textView.

3. PROBLEMS ENCOUNTERED

Struggle while trying to understand the algorithms of DWT watermarking method and don't know how to implement it in Android instead of MATLAB

4. SELF EVALUATION OF THE PROGRESS

Progress is good, first two activities of the project is completed.

X

Mr. Leong Chun Farn
Supervisor"s signature

X

Loi Kong Leong
Student"s signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no:** 6
Student Name & ID: Loi Kong Leong 09ACB04872	
Supervisor: Mr. Leong Chun Farn	
Project Title: Design and Analysis of Message Embedding System for Mobile Operating System Using Spatial Watermarking Technique	

1. WORK DONE

Two main useful component of the project is created. First is the dialog fragment which can be used to request user to enter password for the image in the project apps. Second is using the gallery view such that the image can be displayed in the form of gallery and user is allowed to browse through the image easily to select the image for encryption and decryption.

2. WORK TO BE DONE

Work on AES encryption and DWT watermark embedding.

3. PROBLEMS ENCOUNTERED

Nil.

4. SELF EVALUATION OF THE PROGRESS

The schedule works smoothly

X

Mr. Leong Chun Farn
Supervisor"s signature

X

Loi Kong Leong
Student"s signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no:** 7
Student Name & ID: Loi Kong Leong 09ACB04872	
Supervisor: Mr. Leong Chun Farn	
Project Title: Design and Analysis of Message Embedding System for Mobile Operating System Using Spatial Watermarking Technique	

1. WORK DONE

The gallery view is upgraded to image switcher view where the selected image is now will be shown in larger size below the gallery and the AES encryption is successfully created.

2. WORK TO BE DONE

Combine the fragments of apps done previously to assemble the project apps and try to work on DCT watermarking method.

3. PROBLEMS ENCOUNTERED

Failed to implement DWT watermarking in Android due to lack of function and import.

4. SELF EVALUATION OF THE PROGRESS

Progress is good, the image can be obtained from the mobile devices and show on the gallery.

X

Mr. Leong Chun Farn
Supervisor"s signature

X

Loi Kong Leong
Student"s signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no:** 8
Student Name & ID: Loi Kong Leong 09ACB04872	
Supervisor: Mr. Leong Chun Farn	
Project Title: Design and Analysis of Message Embedding System for Mobile Operating System Using Spatial Watermarking Technique	

1. WORK DONE

After seeking guidance from Mr. Leong, realized that DWT and DCT watermarking is a high level processing for Android implementation.

2. WORK TO BE DONE

Combine all the previous work into single application using intent method.

3. PROBLEMS ENCOUNTERED

Fail to complete the watermarking using DCT watermarking method.

4. SELF EVALUATION OF THE PROGRESS

Progress is good, but there is a problem in developing the DCT using android.

X

Mr. Leong Chun Farn
Supervisor"s signature

X

Loi Kong Leong
Student"s signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no:** 9
Student Name & ID: Loi Kong Leong 09ACB04872	
Supervisor: Mr. Leong Chun Farn	
Project Title: Design and Analysis of Message Embedding System for Mobile Operating System Using Spatial Watermarking Technique	

1. WORK DONE

All the previous work is successfully combined and linked. The application in now can be run with no watermarking.

2. WORK TO BE DONE

Complete the watermarking part of the application

3. PROBLEMS ENCOUNTERED

Nil.

4. SELF EVALUATION OF THE PROGRESS

Progress is good, the application is runnable.

X

Mr. Leong Chun Farn
Supervisor"s signature

X

Loi Kong Leong
Student"s signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Trimester 3 Year 3	**Study week no:** 10
Student Name & ID: Loi Kong Leong 09ACB04872	
Supervisor: Mr. Leong Chun Farn	
Project Title: Design and Analysis of Message Embedding System for Mobile Operating System Using Spatial Watermarking Technique	

1. WORK DONE

After seeking guidance from Mr. Leong for another time, a quantization mehod is introduced by Mr. Leong and knowing that it is possible to implemented it in Android.

2. WORK TO BE DONE

Complete the watermarking by using quantization method.

3. PROBLEMS ENCOUNTERED

Nil.

4. SELF EVALUATION OF THE PROGRESS

Progress is good, some of the application bugs and error are fixed.

X

Mr. Leong Chun Farn
Supervisor"s signature

X

Loi Kong Leong
Student"s signature