

REVERSIBLE IMAGE STEGANOGRAPHY
USING ROI & RONI

LIM JEE CHAO

BACHELOR OF COMPUTER SCIENCE

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's Full Name : Lim Jee Chao

Date of Birth : 12/07/1995

Title : Reversible Image Steganography Using ROI & RONI

Academic Session : Semester I 2018/2019

I declare that this thesis is classified as:

- CONFIDENTIAL (Contains confidential information under the Official Secret Act 1997)*
- RESTRICTED (Contains restricted information as specified by the organization where research was done)*
- OPEN ACCESS I agree that my thesis to be published as online open access (Full Text)

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:



(Student's Signature)



(Supervisor's Signature)

950712-08-6340
New IC/Passport Number
Date: 03/01/2019

Dr. Liew Siau Chuin
Name of Supervisor
Date: 03/01/2019

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.



SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Bachelor Computer Science (Software Engineering)

A handwritten signature in black ink, appearing to be 'LSC', is written above a horizontal line.

(Supervisor's Signature)

Full Name : Dr. Liew Siau Chin
Position : Senior Lecturer
Date : 03/01/2019



STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

A handwritten signature in black ink, appearing to be 'JC' or similar initials, written in a cursive style.

(Student's Signature)

Full Name : Lim Jee Chao
ID Number : CB15010
Date : 03/01/2019

REVERSIBLE IMAGE STEGANOGRAPHY
USING ROI & RONI

LIM JEE CHAO

Thesis submitted in fulfillment of the requirements
for the award of the degree of
Bachelor of Computer Science (Software Engineering)

Faculty of Systems Computer & Software Engineering
UNIVERSITI MALAYSIA PAHANG

JANUARY 2019

ACKNOWLEDGEMENTS

First of all, I am grateful and would like to thank my supervisor, Dr. Liew Siau Chuin for his continuous encouragement, outstanding advice and guidance in conducting this research. I am appreciated for his patience in spending a lot of time to guide me and give a lot of useful suggestions during this period.

Next, I would like to appreciate to my friends who supported and gave their knowledge to me. I would like to express my high appreciation to all lecturers and friends that have guided me. Lastly, I am very grateful to my family for their endless support.

ABSTRAK

Reversible steganography dapat memulihkan imej asal tanpa ada penyelewengan apabila mesej rahsia tertanam telah diekstrak. Kajian ini telah diuji dengan empat imej yang berbeza dan satu storan imej. Imej sampel dibahagikan kepada tiga jenis rantau iaitu *Region of Interest (ROI)*, *Region of Non-Interest (RONI)* dan rantau yang tak tersentuh. Penyelidikan ini menggunakan teknik mencari *ROI* dan *RONI* untuk mencari kedudukan untuk membenamkan mesej dan memulihkan imej asal. Semasa proses pembenaman, *bit RONI* disimpan ke dalam imej storan yang dikenali sebagai *sample_image*. Seterusnya, *bit ROI* disimpan ke *RONI* supaya ia dapat pulih semasa proses pengekstrakan. Pengirim memilih koordinat x dan koordinat y untuk membenamkan maklumat rahsia ke dalam *ROI2*. Pengirim juga perlu membenamkan kunci rahsia dalam imej *RONI2* yang boleh membantu untuk mendapatkan maklumat rahsia. Selepas itu, *stego-imej* dihasilkan selepas *ROI* dan *RONI* tertanam. Dalam proses pengekstrakan, penerima perlu mengekstrak kunci rahsia untuk menyahsulit mesej rahsia. Untuk proses balik, *ROI* dan *RONI* dibalikkan kepada *bit* asal. *Peak Signal-to-Noise Ratio (PSNR)* digunakan untuk mengukur kualiti *stego-imej* dan kesamaan imej asal dan memulihkan imej. Nilai *PSNR* daripada empat imej sampel terpilih adalah antara 52.60dB hingga 52.62dB. Histogram imej asal, *stego-imej* dan imej pulih dijana untuk perbezaan visual antara imej asal, *stego-image* dan memulihkan imej. Kesimpulannya, kaedah yang dicadangkan ini telah membuktikan pendekatan yang lebih baik berbanding dengan kerja sebelumnya dari segi memilih kedudukan untuk membenamkan dengan menggunakan *ROI* dan *RONI*.

ABSTRACT

Reversible steganography allows to recover of original image without any distortion when the embedded secret message has extracted. This research was tested with four different image and one storage image. The sample image is divided into three type of region which are Region of Interest (ROI), Region of Non-Interest (RONI) and untouchable region. This research using the technique of finding the ROI and RONI of the cover image to find position to embed the message and recover the original image. During embedding process, the RONIs' bits are stored into the storage image known as sample_image. Next, the ROIs' bits are stored into RONI so that it can be recovered during extraction process. Sender select the x-coordinate and y-coordinate to embed the secret information into the ROI2. Sender also need embed the secret key in the RONI2 image which can help to secure the secret information. After that, stego-image was generated after ROI and RONI embedded. In extraction process, receiver need to extract the secret key to decrypt the secret message. For reversible process, ROIs and RONIs were reversed to original bits. Peak Signal-to-Noise Ratio (PSNR) value was used to measure the quality of stego-image and similarity of original image and recover image. The value of PSNR of the four selected sample image is between 52.60dB to 52.62dB. Histogram of original image, stego-image and recover image are generated for visual difference between original image, stego-image and recover image. In conclusion, this proposed method has proved a better approach compared to previous work in terms of selecting a position to embed by using ROI and RONI.

TABLE OF CONTENT

DECLARATION	
TITLE PAGE	
ACKNOWLEDGEMENTS	ii
ABSTRAK	iii
ABSTRACT	iv
TABLE OF CONTENT	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	xii
CHAPTER 1 INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objectives	2
1.4 Scope	2
1.5 Significant	3
1.6 Thesis Organization	3
CHAPTER 2 LITERATURE REVIEW	4
2.1 Introduction	4
2.2 Overview of Steganography	4
2.3 Reversible Steganography Techniques	6
2.4 Peak Signal to Noise Ratio	7

2.5	Existing Method in Reversible Steganography	8
2.5.1	Reversible Data Hiding (RDH)	8
2.5.2	High Capacity Reversible Steganography using Multilayer Embedding (CRS)	10
2.5.3	High Capacity and Adaptive Steganographic Algorithm based on Novel Image Interpolation (RAS)	12
2.6	Conclusion	14
CHAPTER 3 METHODOLOGY		15
3.1	Introduction	15
3.2	Methodology	15
3.2.1	Least Significant Bit (LSB)	15
3.2.2	Image used and Secret Information File	16
3.2.3	Secret Key Preparation	20
3.2.4	Embedding Process	21
3.2.5	Extracting Process	23
3.3	Hardware and Software	25
3.4	Gantt Chart	25
3.5	Implementation	26
3.5.1	Example of Secret Text File in Roni	26
3.5.2	Algorithm	28
CHAPTER 4 RESULT & DISCUSSION		30
4.1	Introduction	30
4.2	Process of Reversible Steganography	30
4.2.1	Image and Text Preparation	31
4.2.2	Storage Image Embedding	37

4.2.3	Text Embedding	38
4.2.4	Text Extraction	39
4.2.5	Reversible Steganography	40
4.2.6	Embedding Capacity	40
4.3	Experimental Result	41
4.3.1	Message Embedding and Extraction	45
4.3.2	Reversible Steganography	51
4.4	Discussion	60
CHAPTER 5 CONCLUSION		62
5.1	Introduction	62
5.2	Conclusion	62
5.3	Research Constraint	63
5.4	Future Work	64
5.5	Summary	64
REFERENCES		65
APPENDIX A GANTT CHART		67

LIST OF TABLES

Table 2.1 PSNR value of Wu H., J. Dugelay, and Y. Shi, 2015 method	9
Table 2.2 PSNR value of M. Tang, J. Hu. & W. Song,2014 method	11
Table 2.3: PSNR value of M. Tang, J. Hu, W. Song and S. Zeng,2015 method	13
Table 2.4 Comparison of PSNR between Lena Images in Reversible Steganography Methods	14
Table 3.1 Hardware Requirements	25
Table 3.2 Software Requirements	25
Table 4.1 Secret Text File	31
Table 4.2 Details of Secret Key	31
Table 4.3 Details of lena.bmp and divided regions	32
Table 4.4 Details of peppers.bmp and divided regions	33
Table 4.5 Details of baboon.bmp and divided regions	34
Table 4.6 Details of zelda.bmp and divided regions	35
Table 4.7 PSNR of each steganography image and average PSNR	45
Table 4.8 Experiment 1: Total bits to be embedded is less than ROI2 image	46
Table 4.9 Experiment 2: Total bits to be embedded is larger than ROI2 image	47
Table 4.10 Experiment 3: Total bits to be embedded is less than RONI2 image	48
Table 4.11 Experiment 4: Total bits to be embedded is larger than ROI2 image	49
Table 4.12 Comparison of selected pixels from ROI2	53
Table 4.13 Comparison of selected pixels from RONI1	55
Table 4.14 Comparison of selected pixels from RONI2	57

LIST OF FIGURES

Figure 2.1	Steganography scheme	5
Figure 2.2	Reversible Steganography scheme	6
Figure 2.3	Process of the RDH algorithm	8
Figure 2.4	3 x 3 blocks of interpolating image	10
Figure 2.5	3 x 3 overlapping blocks of interpolating image	12
Figure 3.1	1 st position of Least Significant Bits in pixel	15
Figure 3.2	lena.bmp	16
Figure 3.3	peppers.bmp	16
Figure 3.4	baboon.bmp	17
Figure 3.5	zelda.bmp	17
Figure 3.6	sample of ROI1 (120 x 120 pixels)	17
Figure 3.7	The layout of cover image	18
Figure 3.8	sample of cover image divide into ROI, RONI and untouchable region	18
Figure 3.9	Secret Text File (myfile.txt)	19
Figure 3.10	ASCII Code	20
Figure 3.11	Convert user input to hexadecimal and binary values	20
Figure 3.12	Flowchart of embedding process	22
Figure 3.13	Flowchart of extracting process	24
Figure 3.14	Sample of pixel value in hexadecimal number	26
Figure 3.15	Sample of pixel value in binary number	26
Figure 3.16	Sample of secret message	27
Figure 3.17	Sample of embedded message into a stego-image	27
Figure 4.1	lena.bmp	32
Figure 4.2	Lena ROI1	32
Figure 4.3	Lena ROI2	32
Figure 4.4	Lena RONI1 and RONI2	32
Figure 4.5	pepper.bmp	33
Figure 4.6	Peppers ROI1	33
Figure 4.7	Peppers ROI2	33
Figure 4.8	Peppers RONI1 and RONI2	33
Figure 4.9	Baboon.bmp	34
Figure 4.10	Baboon ROI1	34

Figure 4.11	Baboon ROI2	34
Figure 4.12	Baboon RONI1 and RONI2	34
Figure 4.13	zelda.bmp	35
Figure 4.14	Zelda ROI1	35
Figure 4.15	Zelda ROI2	35
Figure 4.16	Zelda RONI1 and RONI 2	35
Figure 4.17	Storage image to keep original bits of RONI1 and RONI 2 (1300x1300 pixels)	36
Figure 4.18	Illustration of image embedded	37
Figure 4.19	Illustration of message and secret key embedded	38
Figure 4.20	Illustration of message and secret key extraction	39
Figure 4.21	Illustration of reversible steganography	40
Figure 4.22	Stego-image of lena.bmp, PSNR = 52.6054dB	41
Figure 4.23	Stego-image of peppers.bmp, PSNR = 52.6207dB	42
Figure 4.24	Stego-image of baboon.bmp, PSNR = 52.6174dB	42
Figure 4.25	Stego-image of zelda.bmp, PSNR = 52.6143dB	42
Figure 4.26	Histogram of Original Lena and Encrypted Stego-image Lena	43
Figure 4.27	Histogram of Original Peppers and Encrypted Stego-image Peppers	43
Figure 4.28	Histogram of Original Baboon and Encrypted Stego-image Baboon	44
Figure 4.29	Histogram of Original Zelda and Encrypted Stego-image Zelda	44
Figure 4.30	The message file to embed in ROI2 image (348 characters)	49
Figure 4.31	The message that extract in encrypt file.	49
Figure 4.32	The message file to embed in ROI2 image (630 characters)	50
Figure 4.33	The message that extract first 2601pixels in decrypt file.	50
Figure 4.34	User key in the secret key that extract from RONI2.	50
Figure 4.35	User key in the correct key will get the secret message	50
Figure 4.36	User input wrong password will get the error message	50
Figure 4.37	Selected ROI2 region of Lena image	52
Figure 4.38	Selected ROI2 region of Lena image after embedded the secret message	52
Figure 4.39	Selected ROI2 region of Lena image after reversible	53
Figure 4.40	Selected RONI1 region of Lena image	54
Figure 4.41	Selected RONI1 region of Lena image after embedded the original bit of ROI1	54
Figure 4.42	Selected RONI1 region of Lena image after reversible	55
Figure 4.43	Selected RONI2 region of Lena image	56

Figure 4.44	Selected RONI2 region of Lena image after embedded the secret key	56
Figure 4.45	Selected RONI2 region of Lena image after reversible	57
Figure 4.46	Histogram of Original Lena and Recover Lena	58
Figure 4.47	Histogram of Original Peppers and Recover Peppers	58
Figure 4.48	Histogram of Original Baboon and Recover Baboon	59
Figure 4.49	Histogram of Original Zelda and Recover Zelda	59
Figure 4.50	PSNR and MSE of reversed image of Lena	61

LIST OF ABBREVIATIONS

ASCII	American Standard Code for Information Interchange
bmp	Bitmap format file
dB	Decibels
DWT	Discrete Wavelet Transform
LSB	Least Significant Bit
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
ROI	Region of Interest
RONI	Region of Non-Interest

CHAPTER 1

INTRODUCTION

1.1 Introduction

Steganography is a type of technique of hiding some information into a media such as video, text, audio and image. Steganography technique can let us use secret information communication to others. This technique allows us send hided an encrypted message inside another file to receiver so that can avoid detected, stolen, or destroyed by third party. Reversible steganography allows to recover of original image without any distortion when the embedded secret message has extracted. There are two types of reversible steganographic techniques which spatial domain and transform domain. In spatial domain, the Least Significant Bits (LSB) is the most common and simple approach for embedding message in a cover image. It is used the least significant bit of every pixel value in the image. Least Significant Bit is the simplest and easiest way of hiding information. In transform domain, it is a complex way to hide the information into the cover image compare with spatial domain. It transferred the cover image into another transformation and apply data hiding technique on it. There have two type methods in transform domain which Discrete Cosine Transformation (DCT) and Discrete Wavelet Transformation (DWT). DCT embeds the information by altering the transformed DCT co-efficient. DWT work by talking many wavelets to encode the image.

Steganography technique allow hiding information in the image. After embedded secret message into the image, we cannot recover the original image. Hence, we use reversible steganography technique which allows to extract hidden data and recover the original image.

1.2 Problem Statement

When using steganography technique, it will replace the stego-image to the original image. If delete the secret message from stego-image, it still cannot get back the original image. Therefore, steganography method needs to make sure are reversible and the original image can be used repeatedly.

Hence, technique on finding, hide secret information and recover original image on Region of Interest and Region of Non-Interest is proposed. This method prevents data hiding method are irreversible and user can recover the original image.

1.3 Objectives

- i. To study current reversible image steganography technique.
- ii. To propose reversible image steganography using LSB technique.
- iii. To test the reversibility steganography technique to recover the original image.

1.4 Scope

- i. Reversible Image Steganography technique only focuses on grayscale image.
- ii. Demonstrate to ROI and RONI technique to recover original image.
- iii. Test the reversibility steganography technique.

1.5 Significant

- i. Sender and receiver can recover the original image on reversible image steganography technique.
- ii. This research project help to improve the quality of original image.
- iii. To help hiding some information into an image.

1.6 Thesis Organization

This project consists of five chapters. Chapter 1 is research introduction. Chapter 2 is a literature review. Chapter 3 is the methodology of this research. Chapter 4 is implementation, testing and result in discussion. Chapter 5 is the conclusion of this research.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter illustrates details in steganography and comparison of those reversible steganography techniques. Section 2.2 describe on the steganography background. Section 2.3 is discussed on the reversible steganography techniques. Section 2.4 is described about peak signal-to-noise. Section 2.5 is discussed method that used in reversible steganography. Section 2.6 is about the comparison of those reversible steganography techniques.

2.2 Overview of Steganography

Steganography is hide secret messages into digital media. Steganography relate with high security and capacity. There have four elements in steganography such as original image as a cover image, secret key, message and stego-image. Cover image is the input use for embedding with secret message. Secret key is to ensure the security of secret message. We cannot recover the secret information without same secret key. Message is any kind of information that the sender had to send. stego-image is an image that have a secret message. After embedding a secret message in the cover image, stego-image not easily detectable.

Steganography technique has two types domains, which are spatial domain and transform domain. In spatial domain, it used the least significant bit of every pixel value in the image. For example, Least Significant Bit (LSB) is the simplest way of hiding information. In transform domain, it transferred the cover image into another transformation and apply data hiding technique on it. For example, Discrete Cosine Transformation (DCT) and Discrete Wavelet Transformation (DWT) change the value of coefficients quantization.

The general steganography process is illustrated in Figure 2.1.

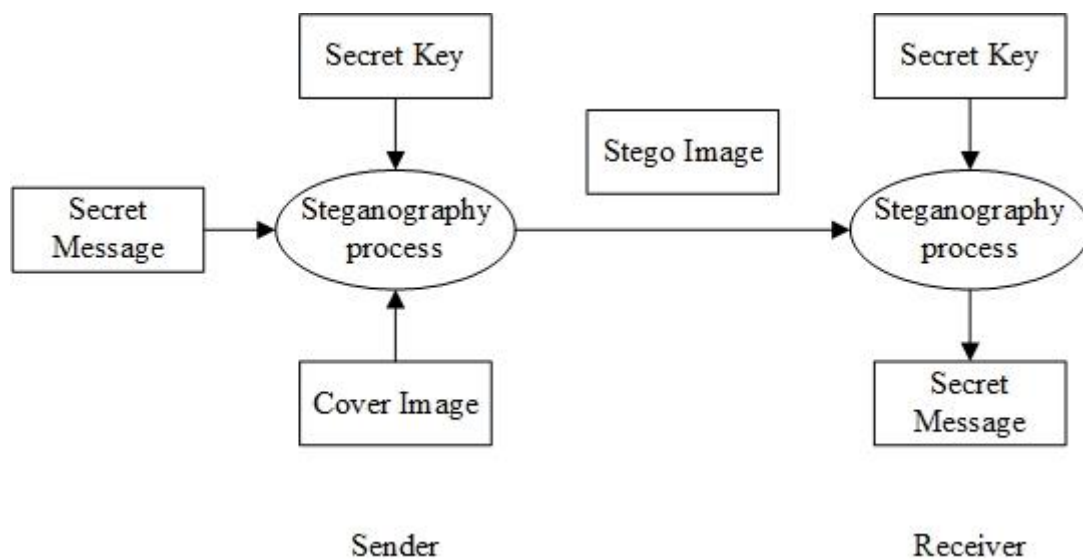


Figure 2.1 Steganography scheme

2.3 Reversible Steganography Techniques

Steganography will destroy to the cover image and resulted in the exact recovery of original image may be impossible. To restore the original image, the steganography method must be reversible, so the original image can used repeatedly. From this issue, reversible steganography is the solution to handle the issue. Reversible steganography is a way that can recover the original image after the data have been extracted. Figure 2.2 shows the reversible steganography process.

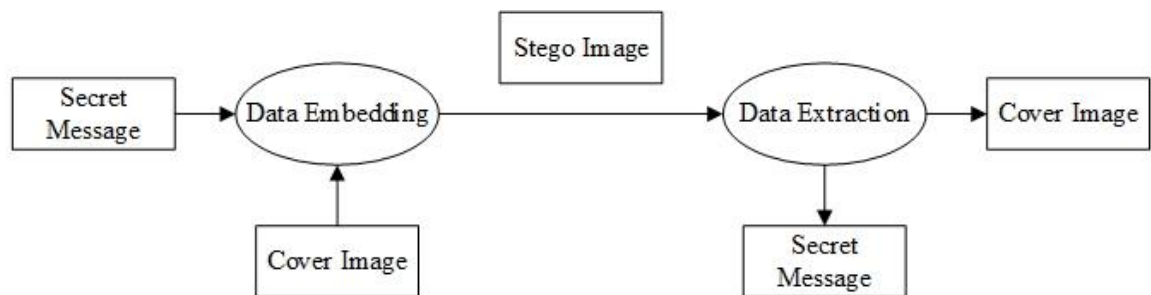


Figure 2.2 Reversible Steganography scheme

2.4 Peak Signal to Noise Ratio

Peak Signal-to-Noise Ratio (PSNR) and Means Square Error (MSE) are used to measure the image quality of original and stego-image. PSNR used to measure quality metric between two images. MSE used to calculate the average of pixels squares value between two images. The higher the PSNR, the higher quality of the stego-image. PSNR and MSE equations are defined as follows:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right)$$

- MAX_I^2 = maximum pixel value of the image

$$MSE = \frac{1}{M \times N} \sum_{i=1}^N \sum_{j=1}^M [I(i, j) - I'(i, j)]^2$$

- M = number of rows of the image
- N = number of columns of the image
- i = index of row
- j = index of column
- I = data of original image
- I' = data of stego-image

2.5 Existing Method in Reversible Steganography

2.5.1 Reversible Data Hiding (RDH)

Wu, Dugelay, and Shi (2015) proposed a reversible image data hiding with Contrast Enhancement. To perform the PSNR value high, the proposed algorithm enhances the contrast of a stego-image to improve its visual quality. For data embedding process, they applied the highest two bins in the histogram, so the same process can be repeated by split each of two peaks into two adjacent bins. The length of the compressed location map, the value of L , the LSBs of the 16 excluded pixels, and the previous peak values are embedded in the last two peaks. The last two peaks are split to replace the LSBs of the 16 excluded pixels. Figure 2.3 shows the produce of the RDH algorithm.

For the extraction process, the LSBs of the 16 excluded pixels are restored because need to know the values of the last two split peaks. The last two split peaks with data embedded are extracted, then the length of the compressed location map, the value of L , the LSBs of the 16 excluded pixels, and the previous peak values are known. The recovery original image is performs by the extraction and recovery operations are repeated until the split peaks retrieved and the data embedded was extracted.

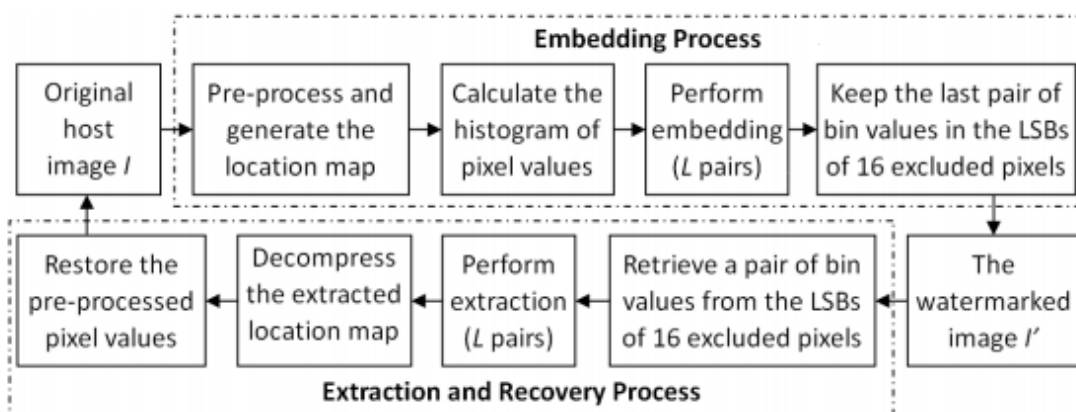


Figure 2.3 Process of the RDH algorithm

The result is stated that the PSNR of the Lena image is 29.10dB and the cover image used is 512x512 8bit grayscale.

Table 2.1 PSNR value of Wu H., J. Dugelay, and Y. Shi, 2015 method

Method	Lena (512x512 with 8bit)
Wu H., J. Dugelay, and Y. Shi (2015)	29.10dB

Source: Wu, Dugelay, and Shi (2015)

2.5.2 High Capacity Reversible Steganography using Multilayer Embedding (CRS)

Tang, Hu, and Song (2014) proposed a reversible steganography method which improve the embedding capacity and keep quality of image. The CRS algorithm was same with the properties of neighboring pixels difference value. CRS reproduced a $R \times L$ cover image, C which to be used in the next phase of secret information embedded. If the secret message is embedded completely, they will show the stego-image S with $R \times L$.

For the embedding phase, CRS requires maximum and minimum values and pixels differences values from 3×3 same block in cover image. When get the differences values, they can define bits or length of a secret message that can be hide in pixel that choose. They calculate the difference values to define the value of secret information to be embedded.

For the extraction phase, is also a reversible process of the embedding process. They will determine maximum and minimum values for the unchanged pixels and difference value of each pixel that chosen. Next, original image exact recover due to the four unchanged pixels of cover image same to the original image. Figure 2.4 displays the process of using 3×3 same blocks of interpolating image for embedding and extracting process.

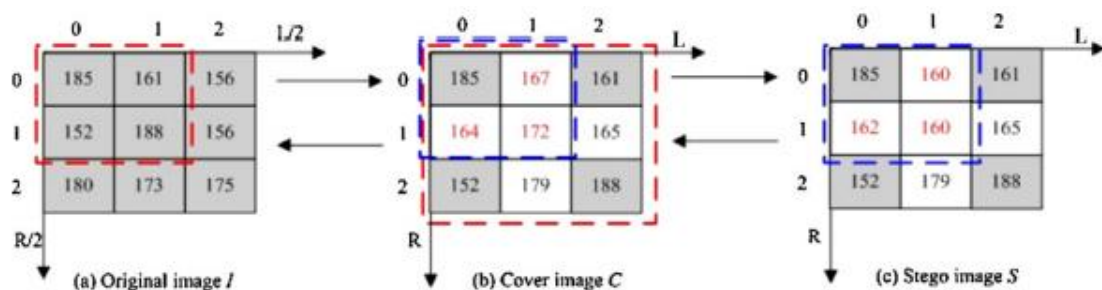


Figure 2.4 3×3 blocks of interpolating image

Table 2.2 PSNR value of M. Tang, J. Hu. & W. Song,2014 method

Method	Lena (512x512 with 8bit)
M. Tang, J. Hu & W. Song., 2014	37.13dB

Source: Tang, Hu, Song (2014)

2.5.3 High Capacity and Adaptive Steganographic Algorithm based on Novel Image Interpolation (RAS)

According to Lee and Huang (2012), they focus on increasing the payload without increased image distortion. Tang et al. (2014) improve the embedding capacity and keep quality of image. Tang, Hu, Song, and Zeng (2015) were improve Lee et al. and Tang et al.'s information hiding algorithm by create a novel image interpolating technique and increase the embed capacity. Figure 2.5 displays the process of using interpolating technique to produce a cover image C_i from the original image O_i . Stego-image S_i is produced after secret messages are embedded to the C_i .

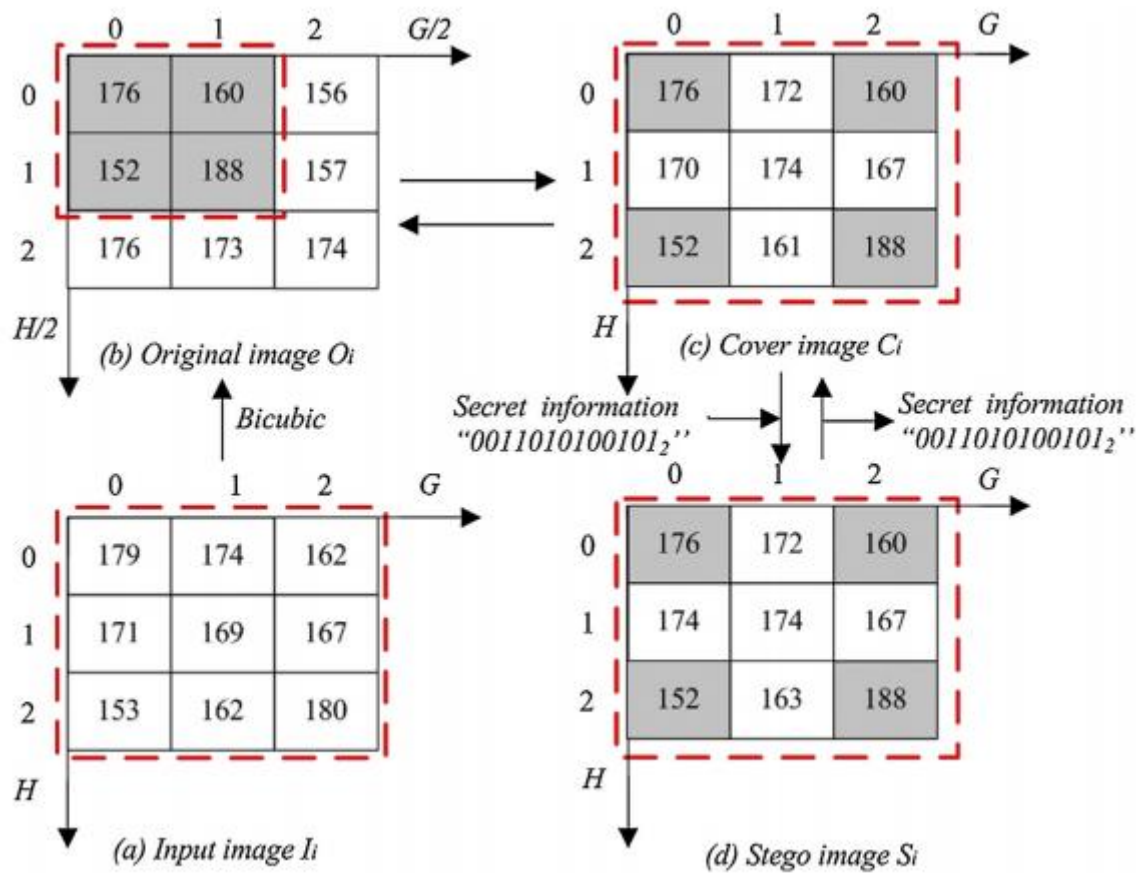


Figure 2.5 3 x 3 overlapping blocks of interpolating image

Source: Tang et al. (2015)

I_i is an input image sized of $H \times G$ and resized to be O_i 's size is $H/2 \times G/2$. RAS use of same properties of adjacent pixels in O_i to remake a $H \times G$ of C_i . For the embedding process, secret message will embed to the C_i and formed a stego-image S_i .

For the extracting process, it will separate to nine stego pixels within a same sub-block and determine the maximum value of the four unchanged pixels. The next step is to exact recover the original image O_i .

Table 2.3: PSNR value of M. Tang, J. Hu, W. Song and S. Zeng,2015 method

Method	Lena (512x512 with 8bit)
M. Tang, J. Hu, W. Song and S. Zeng, 2015	37.87dB

Source: Tang, Hu, Song, Zeng (2015)

2.6 Conclusion

To summarize, table 2.4 shows the comparison of PSNR between three grayscale reversible steganography techniques.

Table 2.4 Comparison of PSNR between Lena Images in Reversible Steganography Methods

Steganography Methods	Reversible Data Hiding with Contrast Enhancement (Wu et al., 2015)	High Capacity Reversible Steganography Using Multilayer Embedding (Tang et al., 2014)	High Capacity and Adaptive Steganographic Algorithm based on Novel Image Interpolation (Tang et al., 2015)
Average PSNR (dB)	29.10dB	37.0229dB	37.3313dB
Reversible	Yes	Yes	Yes
Domain	Spatial	Spatial	Spatial
Advantages	<ul style="list-style-type: none"> • Histogram and location map give the easy calculations • Visual quality can pre-served 	<ul style="list-style-type: none"> • Good quality image • Low complexity of computation 	<ul style="list-style-type: none"> • Low computing complexity • Reduce false effects
Disadvantages	<ul style="list-style-type: none"> • Algorithm is not robust 	<ul style="list-style-type: none"> • The hidden information can be destroyed easily by simple attacks. • Similarities between neighboring pixels 	<ul style="list-style-type: none"> • The hidden information can be destroyed easily by simple attacks.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter consists of five sections. Section 3.1 describes the method and procedures of reversible steganography during this research. Section 3.2 describe the technique used in reversible image steganography. Section 3.3 is list out the hardware and software to be used in this research. Section 3.4 list out the estimated duration of this research. Section 3.5 is about the implementation of this research.

3.2 Methodology

3.2.1 Least Significant Bit (LSB)

Steganography and reversible steganography are embedded in the spatial domain of the image by Least Significant Bit (LSB). LSB is a simple technique and embed the secret message into a cover image and embed bits of ROI and RONI into a RONI and storage image respectively. In this research, LSB focus on the grayscale image. LSB is the lowest bit in a series of numbers in binary. Figure 3.1 show the first position of LSB.

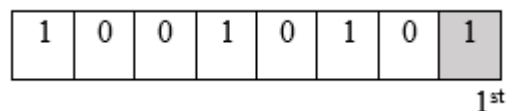


Figure 3.1 1st position of Least Significant Bits in pixel

3.2.2 Image used and Secret Information File

The sample of image that used in this research are as shown in Figure 3.2, Figure 3.3, Figure 3.4 and Figure 3.5. These images are size in 512 x 512 pixels and each pixel has 8 bits. These cover images are used to hide a secret message to produce a stego-image. The layout of the cover images are standardize as the region of interest (ROI) is located at main part of the image and the other parts to be used as the region of non-interest (RONI). For the sample of four images that used, an image with dimension of 120 x 120 pixels will be stored in RONI blocks for reversible purpose. Figure 3.6 shows sample of main part of lena.bmp to store in RONI1 blocks. Figure 3.7 and Figure 3.8 show the cover image consists of three types of region which are ROIs, RONIs and untouchable region. There have one square block of ROIs with 50 x 50 pixels, two rectangle blocks of RONI with 230 x 511 pixels and 150 x 511 pixels and other rest region that untouchable. The number of ROI bits stored in RONI1 is 115,200 bits. The embedding capacity of RONI is 1,553,440 bits. Figure 3.9 shows the sample of secret message in text file which named myfile.txt. Total pixel of this text file is 3066 pixels.



Figure 3.2 lena.bmp



Figure 3.3 peppers.bmp

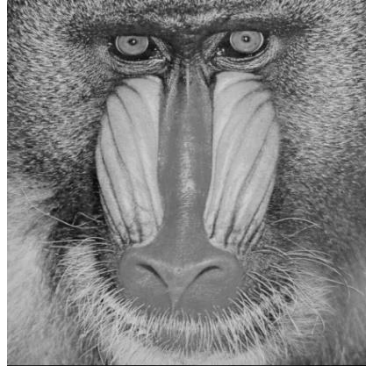


Figure 3.4 baboon.bmp



Figure 3.5 zelda.bmp

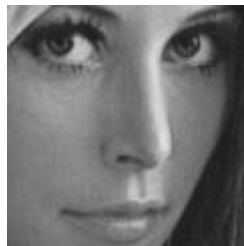


Figure 3.6 sample of ROI1 (120 x 120 pixels)



Figure 3.7 The layout of cover image



Figure 3.8 sample of cover image divide into ROI, RONI and untouchable region

myfile - Notepad

File Edit Format View Help

Reversible steganography allows to recover of original image without any distortion when the embedded secret message has extracted. There are two types of reversible steganographic techniques which spatial domain and transform domain. In spatial domain, the Least Significant Bits (LSB) is the most common and simple approach for embedding message in a cover image. It is used the least significant bit of every pixel value in the image.

Figure 3.9 Secret Text File (myfile.txt)

3.2.3 Secret Key Preparation

Sender used secret key to input secret message and recover original image. It will change the input to hexadecimal form and binary bit representation. Figure 3.10 and 3.11 show ASCII code and the process of change input to hexadecimal form and binary bit respectively.

Character	ASCII	Character	ASCII	Character	ASCII	Character	ASCII	Character	ASCII
a	97	n	110	A	65	N	78	0	48
b	98	o	111	B	66	O	79	1	49
c	99	p	112	C	67	P	80	2	50
d	100	q	113	D	68	Q	81	3	51
e	101	r	114	E	69	R	82	4	52
f	102	s	115	F	70	S	83	5	53
g	103	t	116	G	71	T	84	6	54
h	104	u	117	H	72	U	85	7	55
i	105	v	118	I	73	V	86	8	56
j	106	w	119	J	74	W	87	9	57
k	107	x	120	K	75	X	88		
l	108	y	121	L	76	Y	89		
m	109	z	122	M	77	Z	90		

Figure 3.10 ASCII Code

User Input	Hexadecimal	Binary
pwd1234	112	01110000
	87	01010111
	100	01100100
	49	00110001
	50	00110010
	51	00110011
	52	00110100

01110000010101110110010000110001001100100011001100110100

Figure 3.11 Convert user input to hexadecimal and binary values

3.2.4 Embedding Process

For this process, sender have four files which are original image, storage image, secret key and secret text file. It will be transformed to the 8-bit binary to fulfill LSB substitution method. First, sender need to select X-coordinate and Y-coordinate to generate RONI1 and RONI2 to embed in storage image for reversible purpose. After bit of RONIs embedded into storage image, sender select X-coordinate and Y-coordinate to generate ROI1. Original bits of ROI1 will embedded into bits of RONIs. ROI1 image size is 120 x 120 pixels, ROI1 was cover ROI2, so just save bits of ROI1 into RONI1. ROI2 image size is 50 x 50 pixels and to embed the secret text file. Size of RONI1 and RONI2 is 230 x 511 pixels and 150 x 511 pixels respectively. The secret key will embed into RONI2. To form a stego-image, ROI and RONI image are put to the cover image. Lastly, PSNR are calculated for measure quality of the stego-image and generate histogram to show visual difference between original image and stego-image. The embedding process flowchart is illustrated in Figure 3.12.

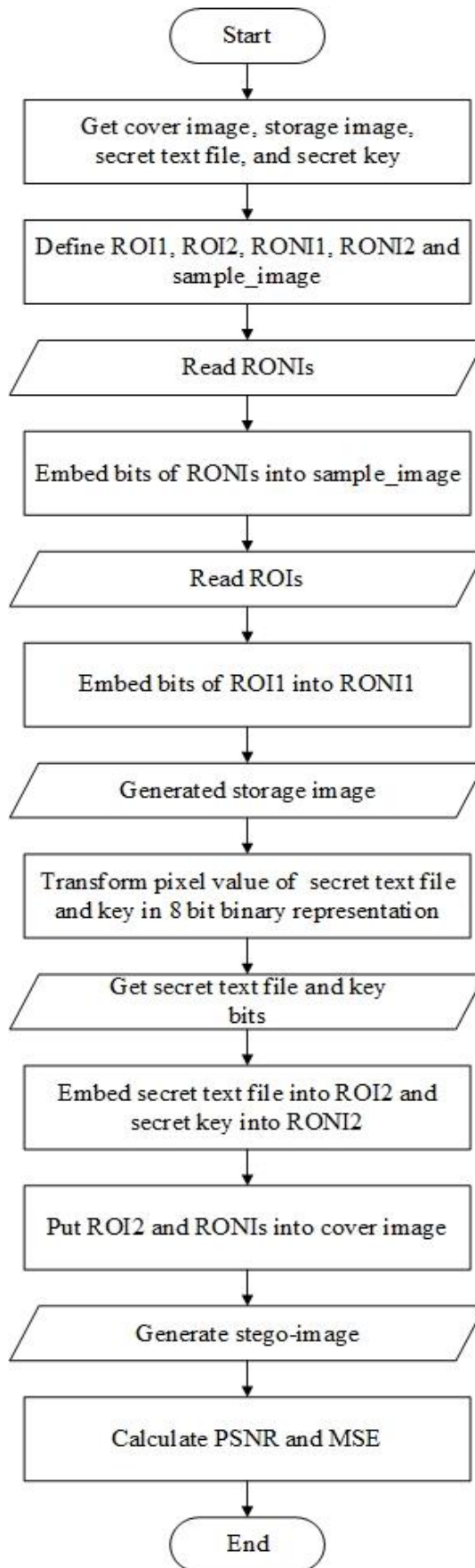


Figure 3.12 Flowchart of embedding process

3.2.5 Extracting Process

In the extraction process, the receiver received the X-coordinate and Y-coordinate which put inside the stego-image. Receiver transform the stego-image into 8-bit binary representation. Next, receiver need to extract the user input in the RONI2. Receiver need to key in the right key which provided by sender. Receiver cannot read the contents if secret key incorrect. After decrypt the message, receiver can choose to recover the original image. Receiver recover the original image by extract from RONI1 to get original bits of ROI1 and extract from sample_image to retrieve bits of RONIs. Lastly, PSNR and MSE will calculated for measure quality of recover image. The process of authentication flowchart is shown in Figure 3.13.

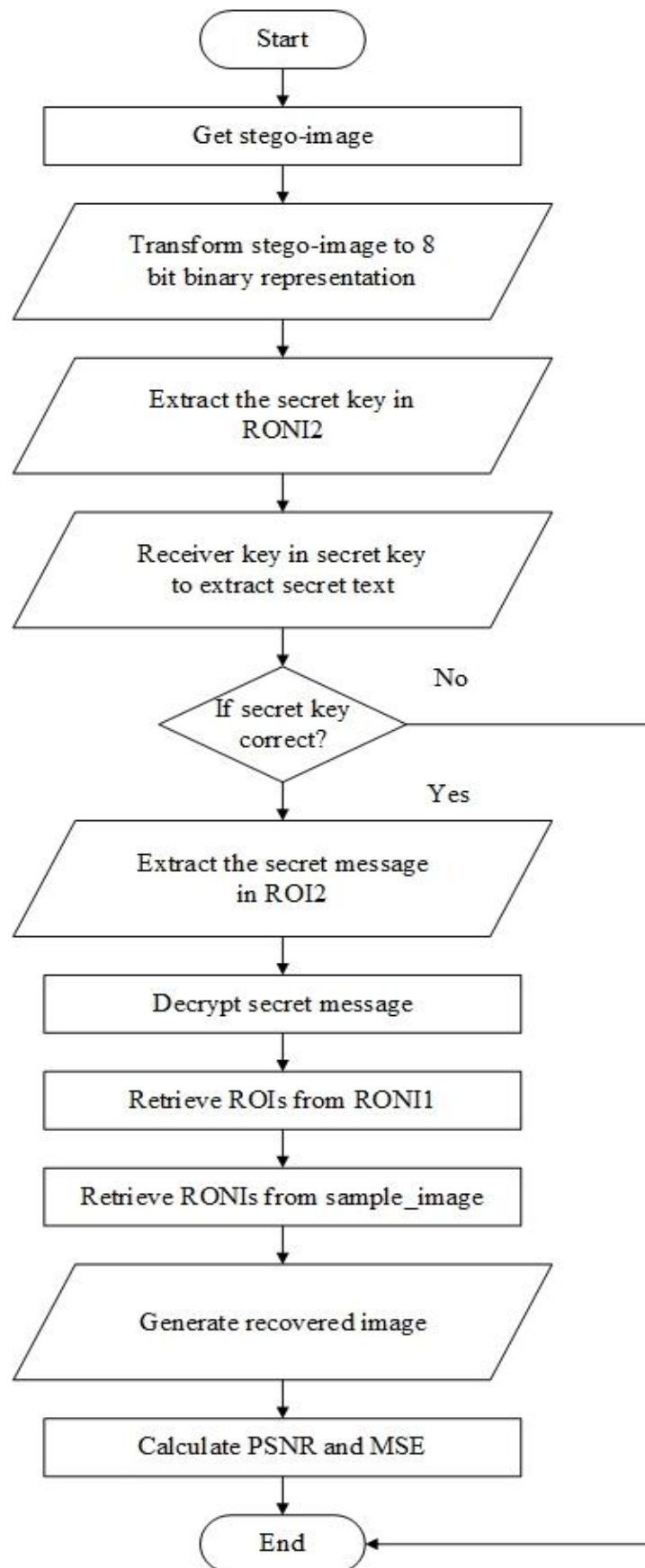


Figure 3.13 Flowchart of extracting process

3.3 Hardware and Software

There are some hardware and software requirements need to carry out this research project. Hardware requirements are displayed in Table 3.1 and software requirements are listed in Table 3.2.

Table 3.1 Hardware Requirements

	Hardware Used	Role
1	Laptop, ASUS A550L	Generate project and research documentation
2	External Hard disk, Toshiba	Data storing and back up
3	Printer, Canon E510	Print out hard copy of research document

Table 3.2 Software Requirements

	Software Used	Role
1	MATLAB R2014a	Coding of this project
2	Microsoft Word 2016	Documentation
3	Microsoft Visio 2016	Flow Chart design
4	Microsoft Project 2016	Gantt Chart design

3.4 Gantt Chart

(Refer to appendix A)

3.5 Implementation

3.5.1 Example of Secret Text File in Roni

Figure 3.14 and Figure 3.15 display the number of pixel and binary number in cover image. Each pixel value needs to convert to a binary number. For instance, 1st pixel is 70 converts to binary 01000110.

70	57	42	100	66
53	51	42	46	63
100	94	82	82	74
90	87	77	80	77
84	108	112	102	...

Figure 3.14 Sample of pixel value in hexadecimal number

01000110	00111001	00101010	01100100	01000010
00110101	00110011	00101010	00101110	00111111
01100100	01011110	01010010	01010010	01001010
01011010	01010111	01001101	01010000	01001101
01010100	01101100	01110000	01100110	...

Figure 3.15 Sample of pixel value in binary number

Figure 3.16 displays the code ASCII of message. Each value of “pwd1234” is embedded into a cover image as shown in Figure 3.14 based on LSB. The embedding message was start from top left corner to right, from top to bottom. After embedding to the cover image, the result of stego-image is generated as shown in Figure 3.17.

Message	ASCII	Binary
pwd1234	112	01110000
	87	01010111
	100	01100100
	49	00110001
	50	00110010
	51	00110011
	52	00110100

Figure 3.16 Sample of secret message

0100011 <u>0</u>	0011100 <u>1</u>	0010101 <u>1</u>	0110010 <u>1</u>	0100001 <u>0</u>
0011010 <u>0</u>	0011001 <u>0</u>	0010101 <u>0</u>	0010111 <u>0</u>	0011111 <u>1</u>
0110010 <u>0</u>	0101111 <u>1</u>	0101001 <u>0</u>	0101001 <u>1</u>	0100101 <u>1</u>
0101101 <u>1</u>	0101011 <u>0</u>	0100110 <u>1</u>	0101000 <u>1</u>	0100110 <u>0</u>
0101010 <u>0</u>	0110110 <u>1</u>	0111000 <u>0</u>	0110011 <u>0</u>	...

Figure 3.17 Sample of embedded message into a stego-image

3.5.2 Algorithm

In this research that I use to implemented as shown as below:

The embedding algorithm of this project:

Input: Cover image, I with 512x512 size, secret message file, secret key, storage image

Output: Stego-image, S

begin

Step 1: Transform pixel value of cover image, text file, and key in 8 binary bits.

Step 2: Select X-coordinate and Y-coordinate of RONIs to store bits of RONIs into storage image for reversible.

Step 3: Select X-coordinate and Y-coordinate ROI to store bits of ROI into RON1I for reversible.

Step 4: Select X-coordinate and Y-coordinate of ROI2 to embed text file.

Step 5: Select X-coordinate and Y-coordinate of RON12 to embed secret key.

Step 6: Output stego-image.

Step 7: End

end

The extracting algorithm of this project:

Input: Stego-image, X-coordinate and Y-coordinate of ROI and RONI, Storage image

Output: Recover image, R

begin

Step 1: Transform pixel value of stego-image in 8 binary bits.

Step 2: Extract the secret key in RONI.

Step 3: Key in secret key. If correct proceed to step 4, else go to step 8.

Step 4: Extract and decrypt the secret file in ROI.

Step 5: Recover original bits of ROI from RONI1.

Step 6: Retrieve original bits of RONIs from storage image.

Step 7: Output recovered image.

Step 8: End.

end

CHAPTER 4

RESULT & DISCUSSION

4.1 Introduction

This chapter discuss about the result of implementing reversible steganography which consists of the process of image and text message embedment, image and text message extraction and recover process.

4.2 Process of Reversible Steganography

There are six main processes in this research which are image and secret text message preparation, image embedding, text message embedding, text message extraction, image extraction and recover image embedding which involve the testing of applying reversible steganography technique on LSB, ROI and RONI in the cover image, testing of security key and the available embedding capacity of ROI with the text message.

4.2.1 Image and Text Preparation

The sample cover images used in this research are 512 x 512 pixels which are lena.bmp, peppers.bmp, baboon.bmp, and zelda.bmp. In the process of preparing of image, the images were divided into ROI and RONI. The details of images and the divided regions are displays in Table 4.3 until Table 4.6. Those tables also will show the storage image with dimension 120 x 120 pixels was used to store original bits of ROIs. There also have a storage image with dimension of 1300 x 1300 pixels were used to keep the original bits of RONIs. Figure 4.17 displays the sample image of 1300 x 1300 pixels.

The text message is prepared for the embedding text process. The secret message is embedded into the ROI2 image. Table 4.1 displays the detail of the secret message file. User will provide the secret key that embedded into the RONI2 image. Table 4.2 shows the details of secret key.

Table 4.1 Secret Text File

File Name	Message	Total Character	Total Bits
myfile.txt	Reversible steganography allows to recover of original image without any distortion when the embedded secret message has extracted. There are two types of reversible steganographic techniques which spatial domain and transform domain. In spatial domain, the Least Significant Bits (LSB) is the most common and simple approach for embedding message in a cover image. It is used the least significant bit of every pixel value in the image.	371	2597

Table 4.2 Details of Secret Key

User Input	Total Character	Total Bits
pwd1234	7	49

Table 4.3 Details of lena.bmp and divided regions





File Name	lena.bmp	File Size
Original Image		512 x 512 pixels
Stored Image (ROI1)		120 x 120 pixels
ROI2		50 x 50 pixels
RONI1 and RONI2		230 x 511 pixels 150 x 511 pixels

Figure 4.4 Lena RONI1 and RONI2

Table 4.4 Details of peppers.bmp and divided regions


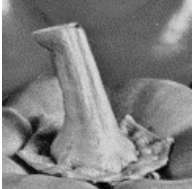
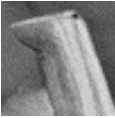

File Name	peppers.bmp	File Size
Original Image		512 x 512 pixels
Stored Image (ROI1)		120 x 120 pixels
ROI2		50 x 50 pixels
RONI1 and RONI2		230 x 511 pixels 150 x 511 pixels

Figure 4.5 pepper.bmp
Figure 4.6 Peppers ROI1
Figure 4.7 Peppers ROI2
Figure 4.8 Peppers RONI1 and RONI2

Table 4.5 Details of baboon.bmp and divided regions

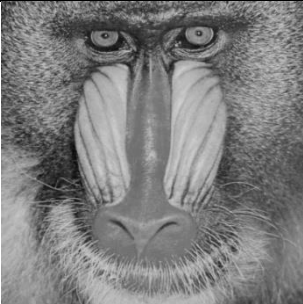


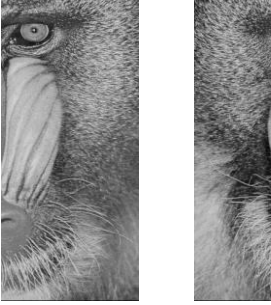
File Name	baboon.bmp	File Size
Original Image		512 x 512 pixels
Figure 4.9 Baboon.bmp		
Stored Image (ROI1)		120 x 120 pixels
Figure 4.10 Baboon ROI1		
ROI2		50 x 50 pixels
Figure 4.11 Baboon ROI2		
RONI1 and RONI2		230 x 511 pixels 150 x 511 pixels
Figure 4.12 Baboon RONI1 and RONI2		

Table 4.6 Details of zelda.bmp and divided regions





File Name	zelda.bmp	File Size
Original Image		512 x 512 pixels
Stored Image (ROI1)		120 x 120 pixels
ROI2		50 x 50 pixels
RONI1 and RONI2		230 x 511 pixels 150 x 511 pixels

Figure 4.13 zelda.bmp

Figure 4.14 Zelda ROI1

Figure 4.15 Zelda ROI2

Figure 4.16 Zelda RONI1 and RONI 2

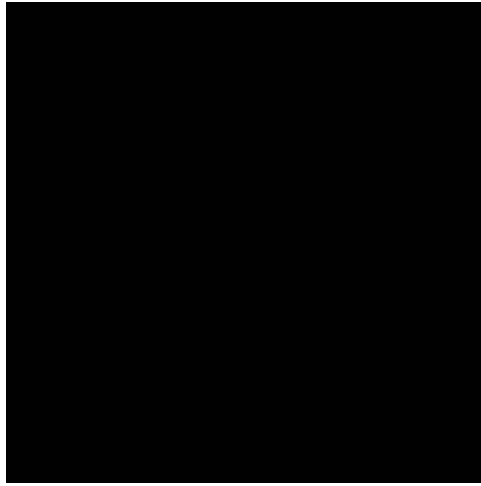


Figure 4.17 Storage image to keep original bits of RONI1 and RONI 2 (1300x1300 pixels)

4.2.2 Storage Image Embedding

In the image embedding process, the original pixels of RONIs were embedded into the first of LSBs of storage image to make RONIs reversible. Next, the first LSB of the RON1 were being changed to embed the storage image of original bits of ROI1. The purpose to store the original bits of ROI1 is to make ROI2 can reverse to the original bits of ROI2.

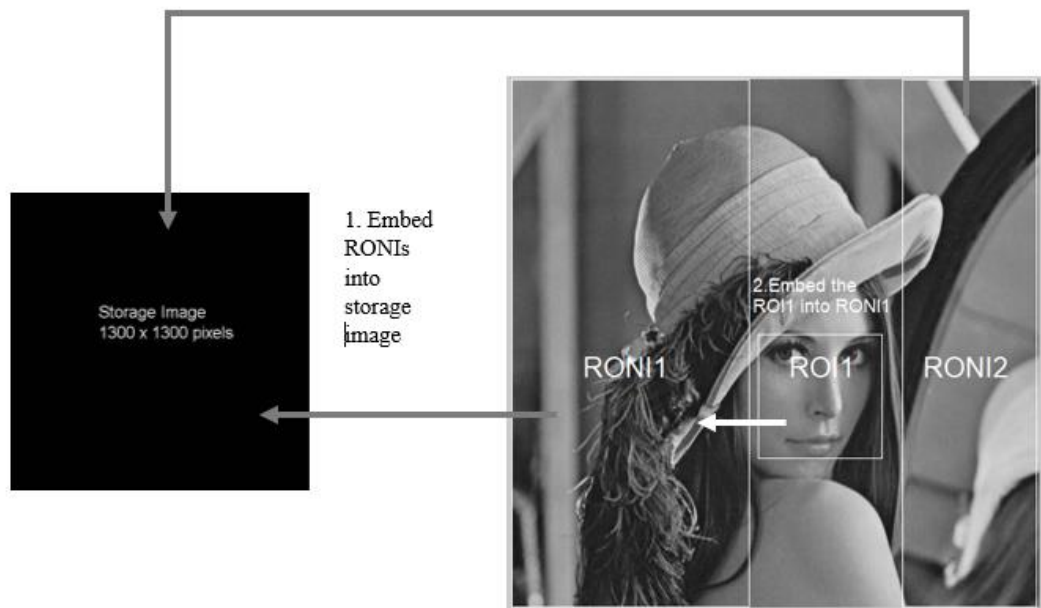


Figure 4.18 Illustration of image embedded

4.2.3 Text Embedding

The secret message will embed into the ROI2 image. The ROI2 image will be selected from the cover image with the storage image. The secret key from user input will embed into the RONI2 image. The RONI2 image also selected from the cover image with storage image.

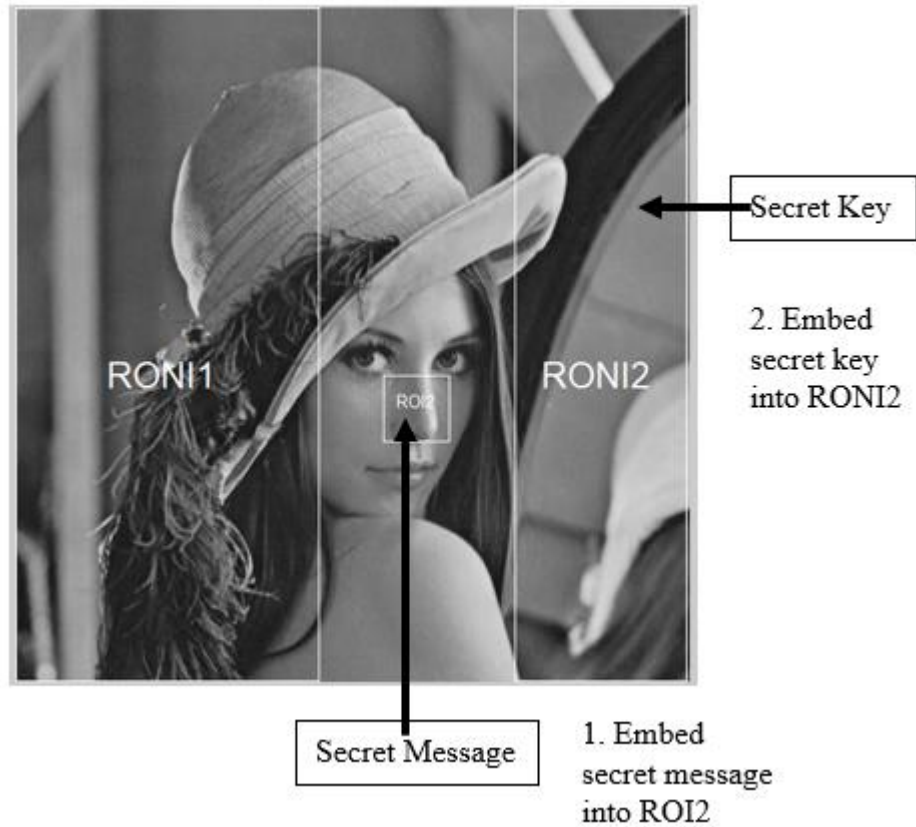


Figure 4.19 Illustration of message and secret key embedded

4.2.4 Text Extraction

For the extraction process, user needs to extract the secret key from the RONI2 of the stego-image. User can extract the secret message from the ROI2 of the stego-image when key in correct key.

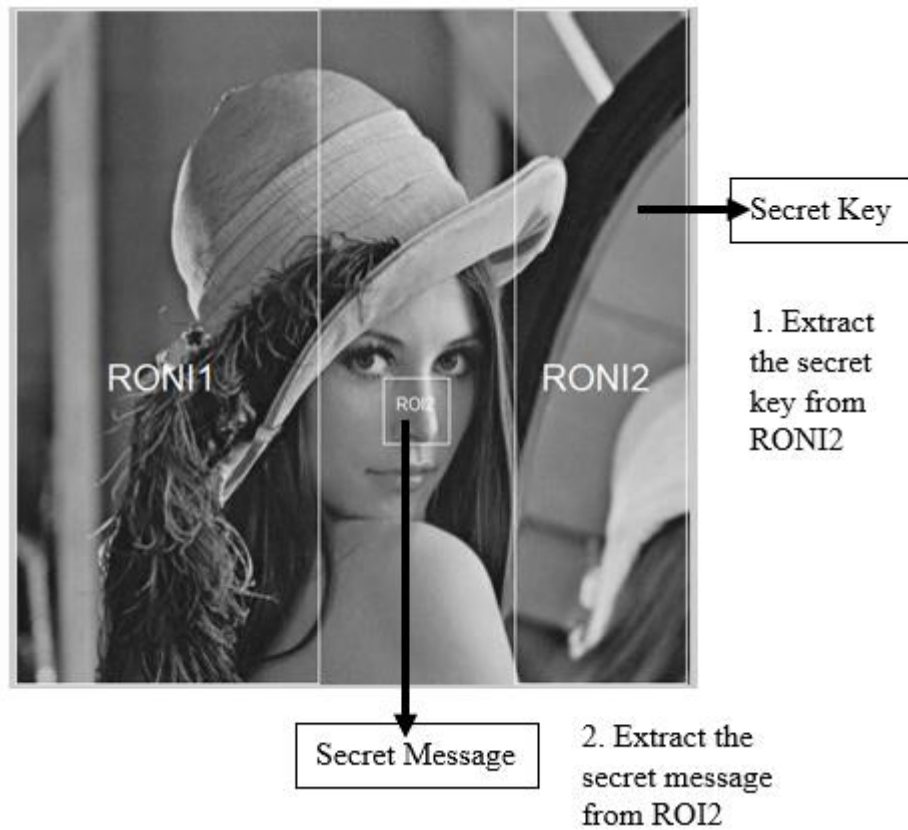


Figure 4.20 Illustration of message and secret key extraction

4.2.5 Reversible Steganography

ROI1 was reversed to original pixels by using reversible steganography through recover of original bits from RONI1. After that, RONI1 and RONI2 were change back to original pixels through retrieve of original bits from the storage image.



Figure 4.21 Illustration of reversible steganography

4.2.6 Embedding Capacity

The embedding capacity will be calculated during the process of embed the original bits of ROI1 to RONI1.

$$\text{Total of pixels of image} * 8 \text{ bits} = \text{Total bits to embed}$$

If total bits to embed is smaller or equal to the total available space of RONI1 image (940240 bits), it will allow embed the ROI1 image. It will reject to embed the ROI image when the total of bits to embed is larger than the available space.

For the text and key embedded process, the embedding capacity will be calculated into the ROI2 and RONI2 image.

$$\text{Total of character count} * 7 \text{ bits} = \text{Total bits to embed}$$

When the total pixel to embed is smaller or equal to the total available space of image (2601) for ROI2 and (77312) for RONI2, it will embed whole text message and secret key into the ROI2 image and RONI2 image respectively.

When the total pixel to embed is bigger than the total available space of image, it will just embed the first 2601 bits and 77312 bits into the ROI or RONI image respectively.

4.3 Experimental Result

Experiment were carried out by performing reversible steganography on four sample images with 512 x 512 pixels. Figure 4.22 to Figure 4.25 show the image steganography with image, text and key. Next, histograms are shown the visual difference between original image with stego-image in Figure 4.26 to Figure 4.29. Table 4.7 show the PSNR value of each steganography image and average PSNR value.



Figure 4.22 Stego-image of lena.bmp, PSNR = 52.6054dB



Figure 4.23 Stego-image of peppers.bmp, PSNR = 52.6207dB

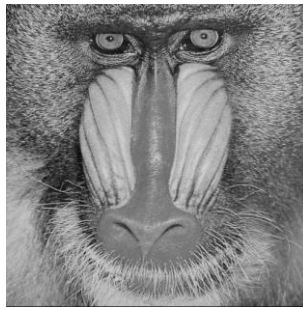


Figure 4.24 Stego-image of baboon.bmp, PSNR = 52.6174dB



Figure 4.25 Stego-image of zelda.bmp, PSNR = 52.6143dB

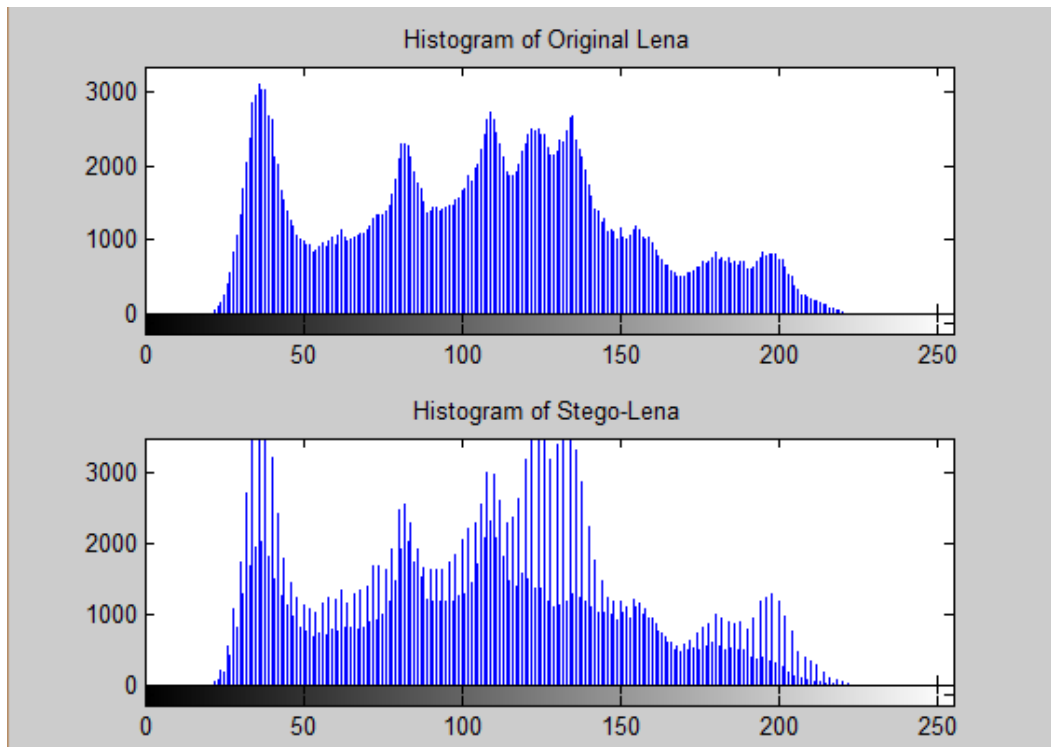


Figure 4.26 Histogram of Original Lena and Encrypted Stego-image Lena

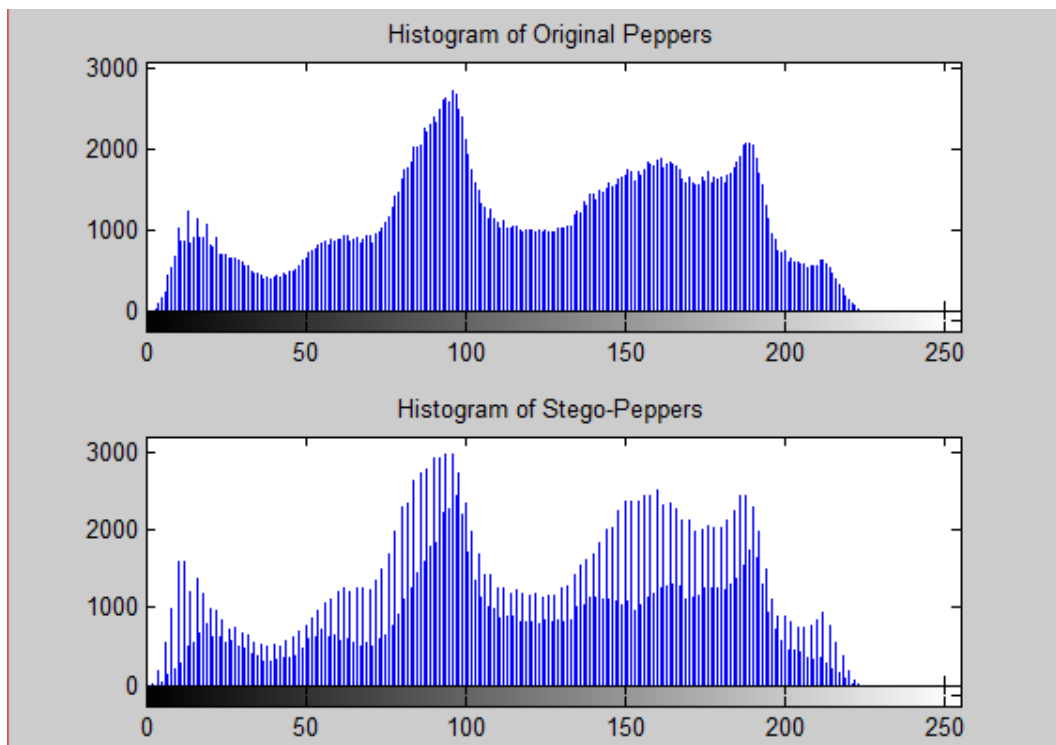


Figure 4.27 Histogram of Original Peppers and Encrypted Stego-image Peppers

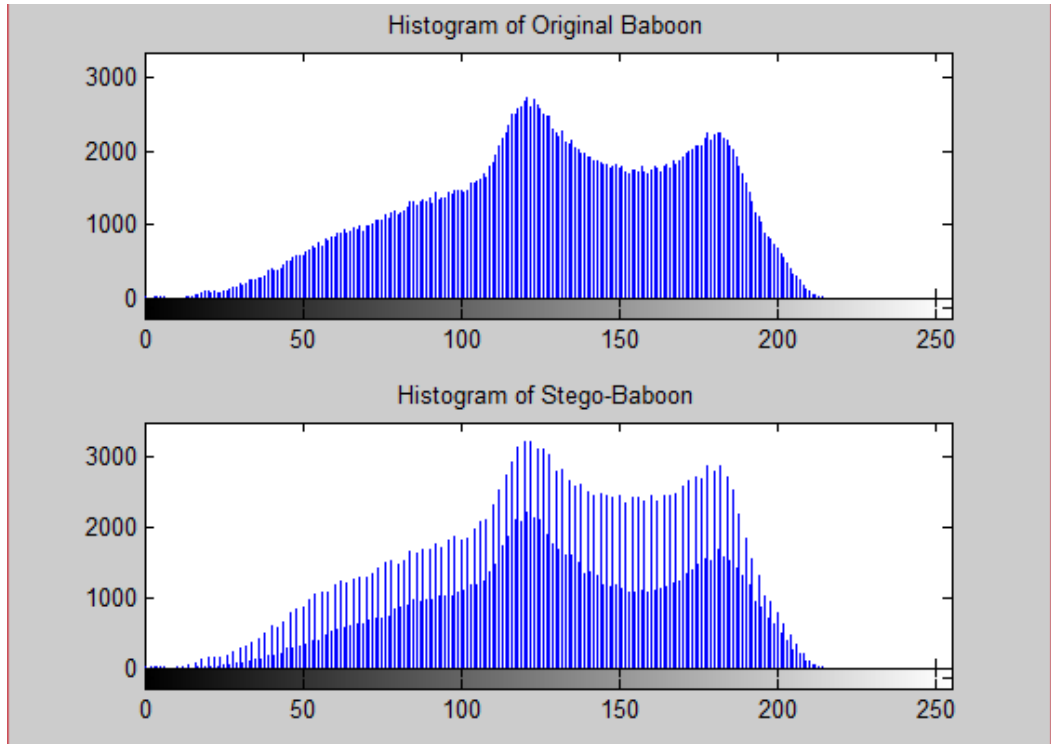


Figure 4.28 Histogram of Original Baboon and Encrypted Stego-image Baboon

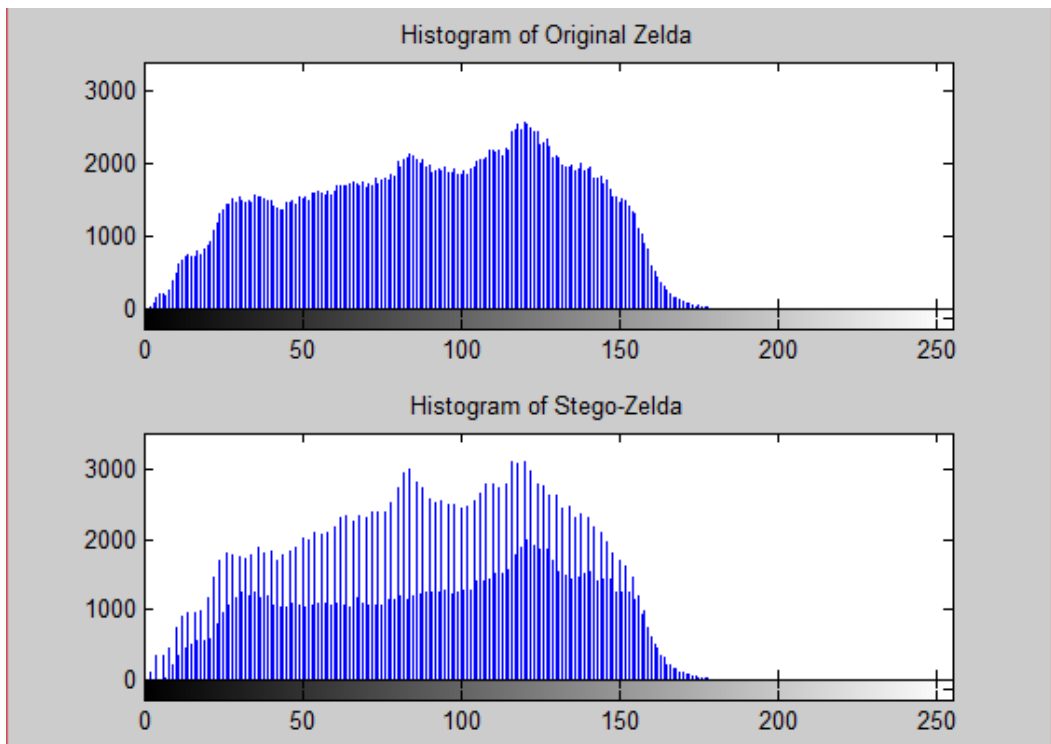


Figure 4.29 Histogram of Original Zelda and Encrypted Stego-image Zelda

Table 4.7 PSNR of each steganography image and average PSNR

Steganography Image	PSNR (dB)
lena.bmp	52.6054
peppers.bmp	52.6207
baboon.bmp	52.6174
zelda.bmp	52.6143
Average	52.6145

4.3.1 Message Embedding and Extraction

There are four experiments are focus on capability on embed the secret message and key into ROI2 and RONI2 image. The first experiment is total bits of message need to embed is less than total available space. The second experiment is total bits of message need to embed is larger than available space. Figure 4.30 and Figure 4.32 show the file that need embed to the ROI2 image. Figure 4.31 and Figure 4.33 show the result of the experiment. Figure 4.34 to Figure 4.36 display the user how input the secret key.

Table 4.8 Experiment 1: Total bits to be embedded is less than ROI2 image

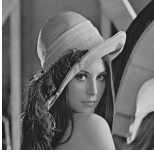



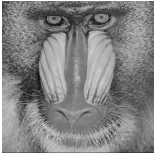



Image	ROI2	Available space (pixels)	Message File (character)	Total bits to embed (bits)	Allow to embed?
 lena.bmp		2601	348	2436	YES
 peppers.bmp		2601	348	2436	YES
 baboon.bmp		2601	348	2436	YES
 zelda.bmp		2601	348	2436	YES

Table 4.9 Experiment 2: Total bits to be embedded is larger than ROI2 image

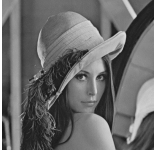



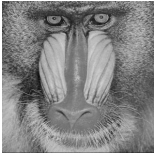



Image	ROI2	Available space (pixels)	Message File (character)	Total bits to embed (bits)	Allow to embed?
 lena.bmp		2601	630	4410	Yes, but embedded the first 2601 pixels
 peppers.bmp		2601	630	4410	Yes, but embedded the first 2601 pixels
 baboon.bmp		2601	630	4410	Yes, but embedded the first 2601 pixels YES
 zelda.bmp		2601	630	4410	Yes, but embedded the first 2601 pixels YES

Table 4.10 Experiment 3: Total bits to be embedded is less than RONI2 image

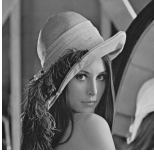


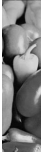
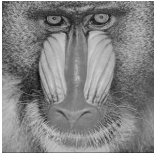



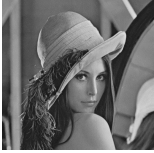


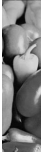
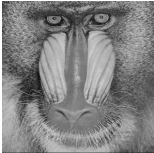



Image	RONI2	Available space (pixels)	Secret key (character)	Total bits to embed (bits)	Allow to embed?
 lena.bmp		77312	57	357	YES
 peppers.bmp		77312	57	357	YES
 baboon.bmp		77312	57	357	YES
 zelda.bmp		77312	57	357	YES

Table 4.11 Experiment 4: Total bits to be embedded is larger than ROI2 image

Image	ROI2	Available space (pixels)	Message File (character)	Total bits to embed (bits)	Allow to embed?
 lena.bmp		77312	11500	80500	Yes, but embedded the first 77312 pixels
 peppers.bmp		77312	11500	80500	Yes, but embedded the first 77312 pixels
 baboon.bmp		77312	11500	80500	Yes, but embedded the first 77312 pixels
 zelda.bmp		77312	11500	80500	Yes, but embedded the first 77312 pixels

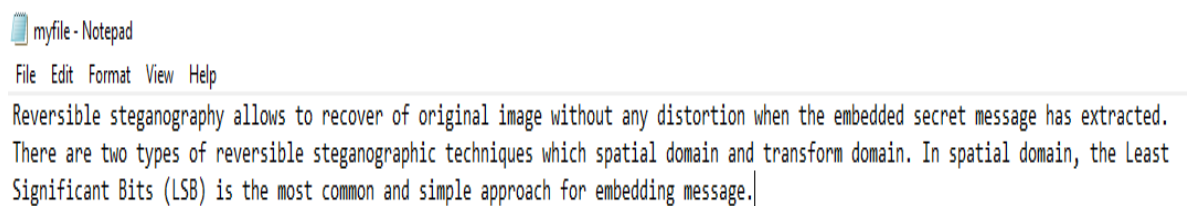


Figure 4.30 The message file to embed in ROI2 image (348 characters)

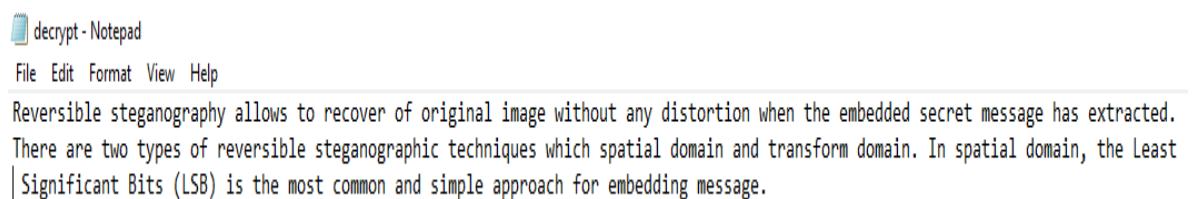


Figure 4.31 The message that extract in encrypt file.

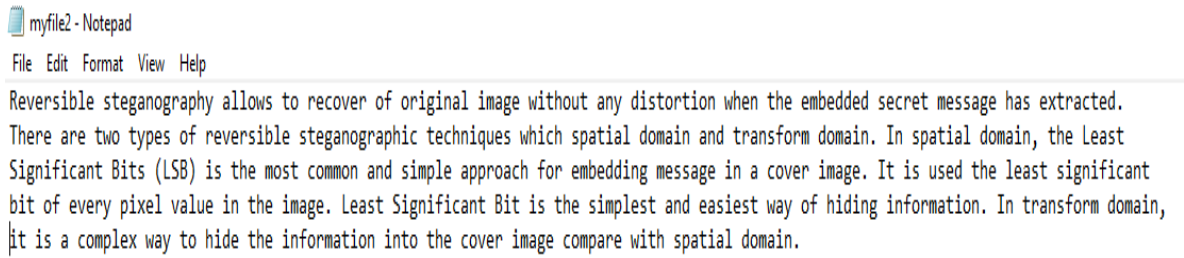


Figure 4.32 The message file to embed in ROI2 image (630 characters)

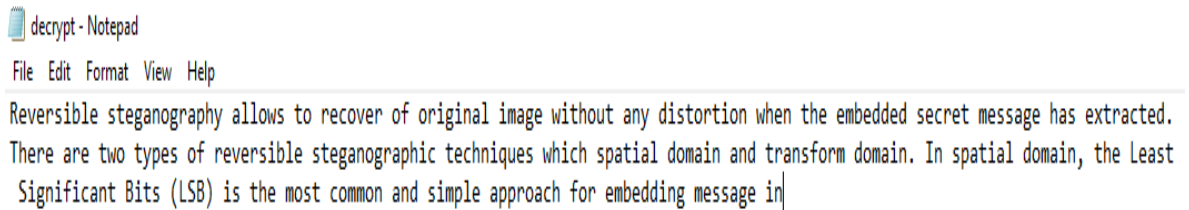


Figure 4.33 The message that extract first 2601pixels in decrypt file.

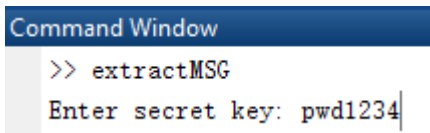


Figure 4.34 User key in the secret key that extract from RONI2.

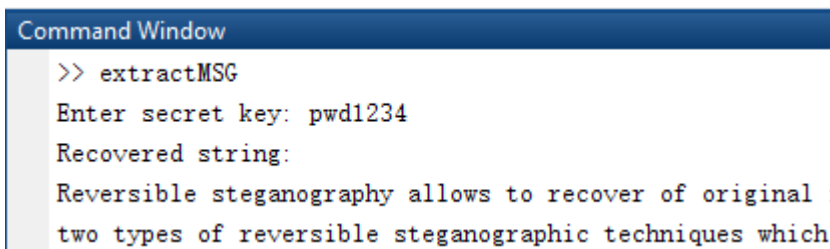


Figure 4.35 User key in the correct key will get the secret message

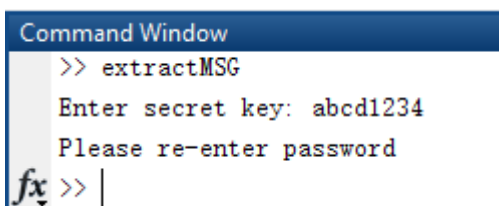


Figure 4.36 User input wrong password will get the error message

4.3.2 Reversible Steganography

The RONI1's first LSBs were being changed to embed the original bits of ROI1. The ROI2 and RONI2's first LSBs also modified due to embed message and key. Thus, the reversible steganography is performed on the ROI and RONI to reverse the original bit value after finished the process of embed message and extraction process. The efficiency of the reversible steganography scheme applies in this research is 100%. All the pixels value of ROI and RONI image were success to reverse back to their original bit values.

The screenshot of pixel values of selected ROI2 region from lena.bmp are shows in Figure 4.37. The screenshot of pixel values of selected ROI2 region from lena.bmp after embedded the secret message and reverse to the original bit values of ROI2. Table 4.12 displays the comparison of the different of selected pixels from the ROI2 region.

Figure 4.40 displays the screenshot of pixel values of selected RONI1 region from lena.bmp. Figure 4.41 displays the screenshot of pixel values of selected RONI1 region from lena.bmp after embedded the original bit of ROI1 and Figure 4.42 shows the screenshot of pixel values of selected RONI1 region after recover the storage original pixel values. Table 4.13 shows the difference of selected pixels from the RONI1 region between original image, steganography image and reversible steganography image.

Figure 4.43 shows the screenshot of pixel values of selected RONI2 region from lena.bmp. Figure 4.44 and Figure 4.45 show the screenshot of pixel values of selected RONI2 from lena.bmp after embedded the secret key and recover to the original pixels value. The comparison of selected pixels from the RONI2 region are shown in Table 4.14. Figure 4.46 to Figure 4.49 show the visual difference between original image with recover image.

Variables - originalLena									
	288	289	290	291	292	293	294	295	
298	127	131	125	126	129	124	121	121	
299	125	128	126	128	129	123	119	119	
300	125	123	123	127	129	122	118	119	
301	125	119	119	125	126	120	119	120	
302	123	117	117	122	123	119	119	121	
303	118	117	117	120	120	118	120	120	
304	116	119	117	120	119	117	119	119	
305	115	118	113	110	114	122	124	116	
306	114	116	113	113	114	118	118	114	
307	112	114	114	114	113	112	113	112	
308	113	113	113	114	114	113	111	111	
309	115	113	112	113	114	115	114	111	
310	117	112	111	112	114	115	115	111	
311	115	111	111	111	113	114	113	110	
312	113	110	110	110	110	111	110	109	

Figure 4.37 Selected ROI2 region of Lena image

Variables - StegoLena									
	288	289	290	291	292	293	294	295	
298	126	131	124	127	129	124	120	121	
299	125	128	126	128	129	122	119	119	
300	125	122	123	126	128	123	119	119	
301	125	118	119	125	127	120	119	121	
302	122	116	117	122	122	118	119	120	
303	119	116	117	120	120	119	121	120	
304	117	119	117	120	118	117	118	119	
305	115	119	112	110	115	122	124	117	
306	115	116	113	112	115	118	119	115	
307	113	114	114	115	112	113	112	113	
308	112	112	112	115	114	112	111	110	
309	115	112	113	113	115	114	115	110	
310	116	113	111	112	114	115	114	110	
311	114	111	110	111	113	115	112	110	
312	113	111	111	111	110	110	111	109	

Figure 4.38 Selected ROI2 region of Lena image after embedded the secret message

	288	289	290	291	292	293	294	295
298	127	131	125	126	129	124	121	121
299	125	128	126	128	129	123	119	119
300	125	123	122	127	129	122	118	119
301	125	119	119	125	126	120	119	120
302	123	117	117	122	123	119	119	121
303	118	117	117	120	120	118	120	120
304	116	119	117	120	119	117	119	119
305	115	118	113	110	114	122	124	116
306	114	116	113	113	114	118	118	114
307	112	114	114	114	113	112	113	112
308	113	113	113	114	114	113	111	111
309	115	113	112	113	114	115	114	111
310	117	112	111	112	114	115	115	111
311	115	111	111	111	113	114	113	110
312	113	110	110	110	110	111	110	109

Figure 4.39 Selected ROI2 region of Lena image after reversible

Table 4.12 Comparison of selected pixels from ROI2

Pixel Coordinate (x, y)	Pixel Value		
	Original Image	Steganography Image	Reversed Image
(288, 303)	118	119	118
(289, 113)	113	112	113
(290, 122)	122	123	122
(292, 114)	114	115	114
(294, 115)	115	114	115

Variables - originalLena									
originalLena × StegoLena × recoverLena ×									
512x512 uint8									
	11	12	13	14	15	16	17	18	
22	137	137	138	139	139	139	140	139	
23	138	138	139	139	139	138	140	141	
24	138	138	139	138	139	139	138	139	
25	138	140	139	139	138	138	140	139	
26	140	139	140	140	139	138	139	138	
27	140	141	141	139	139	139	139	138	
28	139	140	140	140	138	138	140	140	
29	139	140	140	139	138	137	140	139	
30	139	140	139	139	139	138	140	139	
31	141	141	140	141	140	138	141	140	
32	142	143	143	142	141	141	143	141	
33	139	140	141	141	140	139	141	141	
34	140	138	138	140	141	142	142	142	
35	140	139	140	141	142	143	143	141	
36	141	145	144	143	142	141	144	143	

Figure 4.40 Selected RONI1 region of Lena image

Variables - StegoLena									
originalLena × StegoLena × recoverLena ×									
	11	12	13	14	15	16	17	18	
22	136	137	138	139	138	139	140	138	
23	138	139	138	139	138	139	140	140	
24	138	139	138	139	138	139	138	138	
25	138	140	138	139	138	139	140	139	
26	140	138	140	141	138	139	138	139	
27	140	140	140	139	138	139	138	139	
28	138	140	140	141	138	139	140	141	
29	138	140	140	139	138	137	140	139	
30	139	140	138	139	138	139	140	139	
31	141	140	140	141	140	139	140	141	
32	143	142	143	143	140	141	142	141	
33	139	140	141	141	140	138	140	141	
34	141	138	138	141	140	142	142	143	
35	141	138	140	141	142	142	142	141	
36	141	144	145	143	142	140	144	143	

Figure 4.41 Selected RONI1 region of Lena image after embedded the original bit of ROI1

	11	12	13	14	15	16	17	18
22	137	137	138	139	139	139	140	139
23	138	138	139	139	139	138	140	141
24	138	138	139	138	139	139	138	139
25	138	140	139	139	138	138	140	139
26	140	139	140	140	139	138	139	138
27	140	141	141	139	139	139	139	138
28	139	140	140	140	138	138	140	140
29	139	140	140	139	138	137	140	139
30	139	140	139	139	139	138	140	139
31	141	141	140	141	140	138	141	140
32	142	143	143	142	141	141	143	141
33	139	140	141	141	140	139	141	141
34	140	138	138	140	141	142	142	142
35	140	139	140	141	142	143	143	141
36	141	145	144	143	142	141	144	143

Figure 4.42 Selected RONI1 region of Lena image after reversible

Table 4.13 Comparison of selected pixels from RONI1

Pixel Coordinate (x, y)	Pixel Value		
	Original Image	Steganography Image	Reversed Image
(11, 28)	139	138	139
(13, 36)	144	145	144
(14, 24)	138	139	138
(15, 32)	141	140	141
(18, 22)	139	138	139

Variables - originalLena										
originalLena x StegoLena x recoverLena x										
512x512 uint8										
	422	423	424	425	426	427	428	429		
206	40	43	53	51	56	66	83	102		
207	40	44	50	51	61	75	90	104		
208	40	41	43	54	68	85	99	106		
209	47	47	47	55	76	95	103	113		
210	47	48	52	60	81	98	104	112		
211	45	50	57	67	88	101	105	110		
212	46	50	62	75	95	106	107	110		
213	49	55	71	86	104	111	109	109		
214	51	60	77	96	111	115	111	108		
215	53	66	84	97	112	116	108	105		
216	58	75	95	94	109	112	107	104		
217	69	84	99	110	114	115	111	104		
218	76	93	110	114	116	116	111	106		
219	80	96	111	115	116	114	109	103		
220	85	100	114	117	115	110	105	102		

Figure 4.43 Selected RONI2 region of Lena image

Variables - StegoLena										
originalLena x StegoLena x recoverLena x										
512x512 uint8										
	422	423	424	425	426	427	428	429		
206	40	42	52	50	56	66	82	102		
207	40	44	50	50	60	74	90	104		
208	40	40	42	54	68	84	98	106		
209	46	46	46	54	76	94	102	112		
210	46	48	52	60	80	98	104	112		
211	44	50	56	66	88	100	104	110		
212	46	50	62	74	94	106	106	110		
213	48	54	70	86	104	110	108	108		
214	50	60	76	96	110	114	110	108		
215	52	66	84	96	112	116	108	104		
216	58	74	94	94	108	112	106	104		
217	68	84	98	110	114	114	110	104		
218	76	92	110	114	116	116	110	106		
219	80	96	110	114	116	114	108	102		
220	84	100	114	116	114	110	104	102		

Figure 4.44 Selected RONI2 region of Lena image after embedded the secret key

	422	423	424	425	426	427	428	429
206	40	43	53	51	56	66	83	102
207	40	44	50	51	61	75	90	104
208	40	41	43	54	68	85	99	106
209	47	47	47	55	76	95	103	113
210	47	48	52	60	81	98	104	112
211	45	50	57	67	88	101	105	110
212	46	50	62	75	95	106	107	110
213	49	55	71	86	104	111	109	109
214	51	60	77	96	111	115	111	108
215	53	66	84	97	112	116	108	105
216	58	75	95	94	109	112	107	104
217	69	84	99	110	114	115	111	104
218	76	93	110	114	116	116	111	106
219	80	96	111	115	116	114	109	103
220	85	100	114	117	115	110	105	102

Figure 4.45 Selected RONI2 region of Lena image after reversible

Table 4.14 Comparison of selected pixels from RONI2

Pixel Coordinate (x, y)	Pixel Value		
	Original Image	Steganography Image	Reversed Image
(422, 217)	69	68	69
(424, 213)	71	70	71
(426, 210)	81	80	81
(427, 207)	75	74	75
(429, 213)	109	108	109

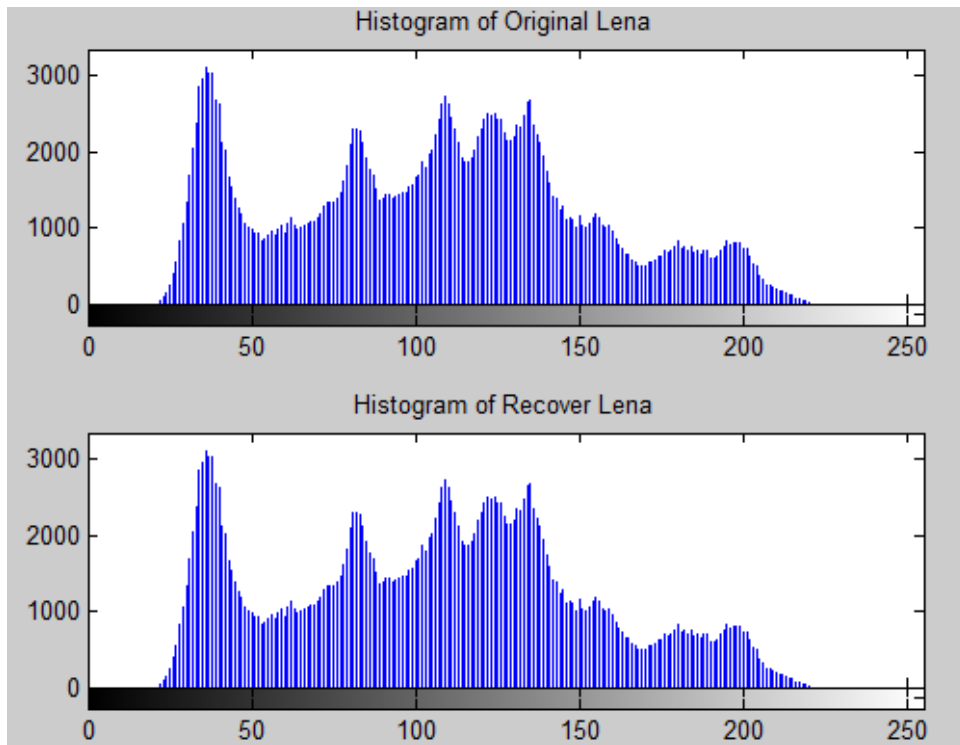


Figure 4.46 Histogram of Original Lena and Recover Lena

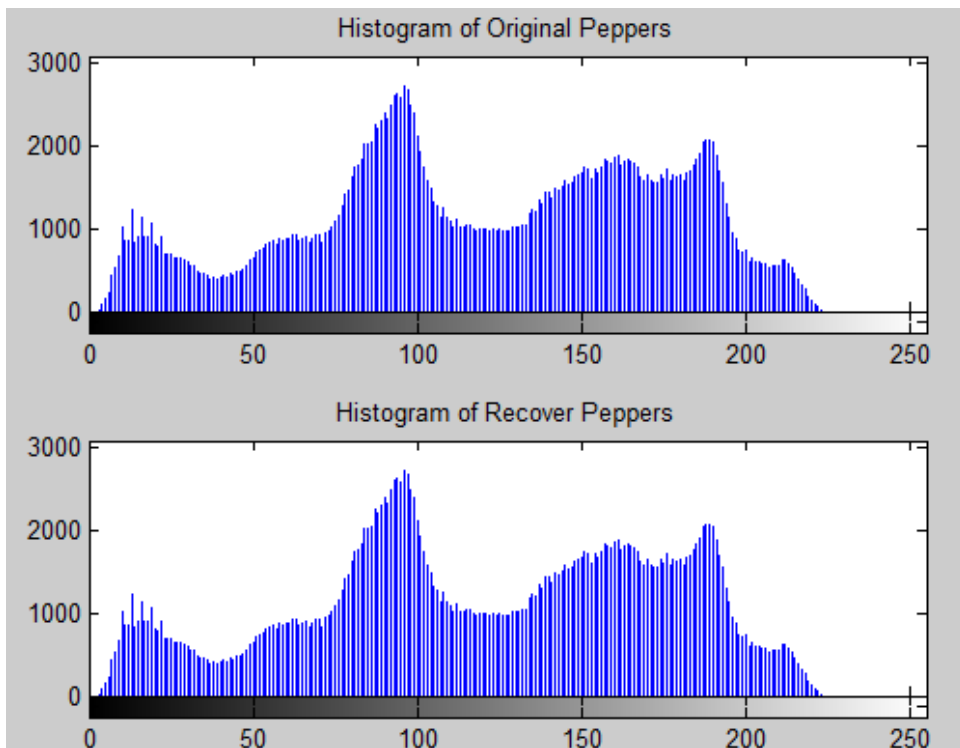


Figure 4.47 Histogram of Original Peppers and Recover Peppers

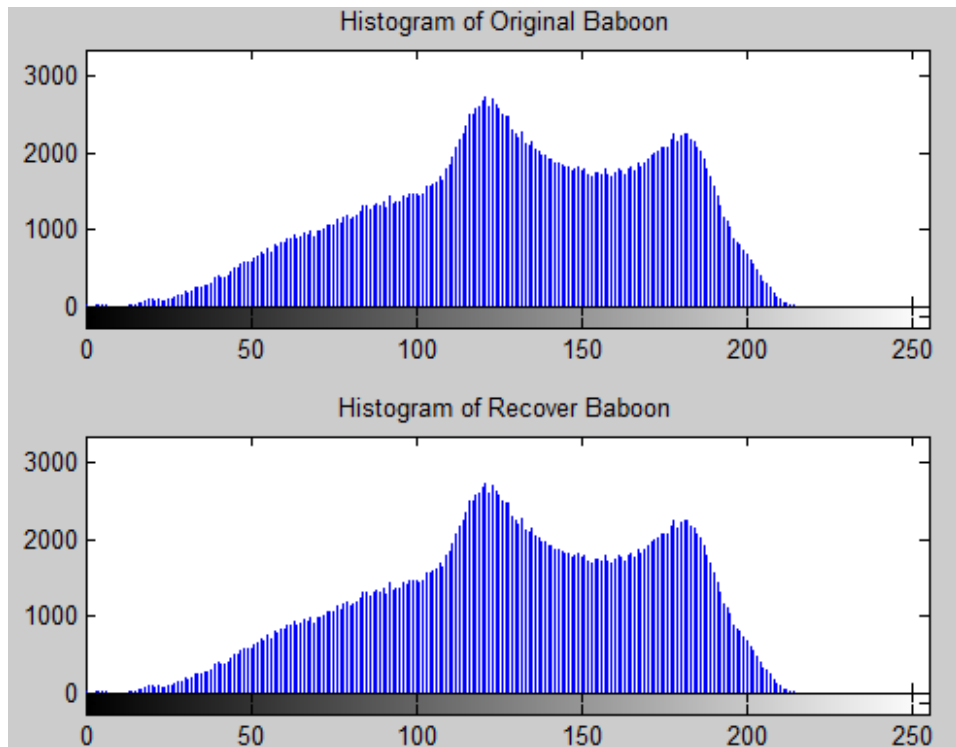


Figure 4.48 Histogram of Original Baboon and Recover Baboon

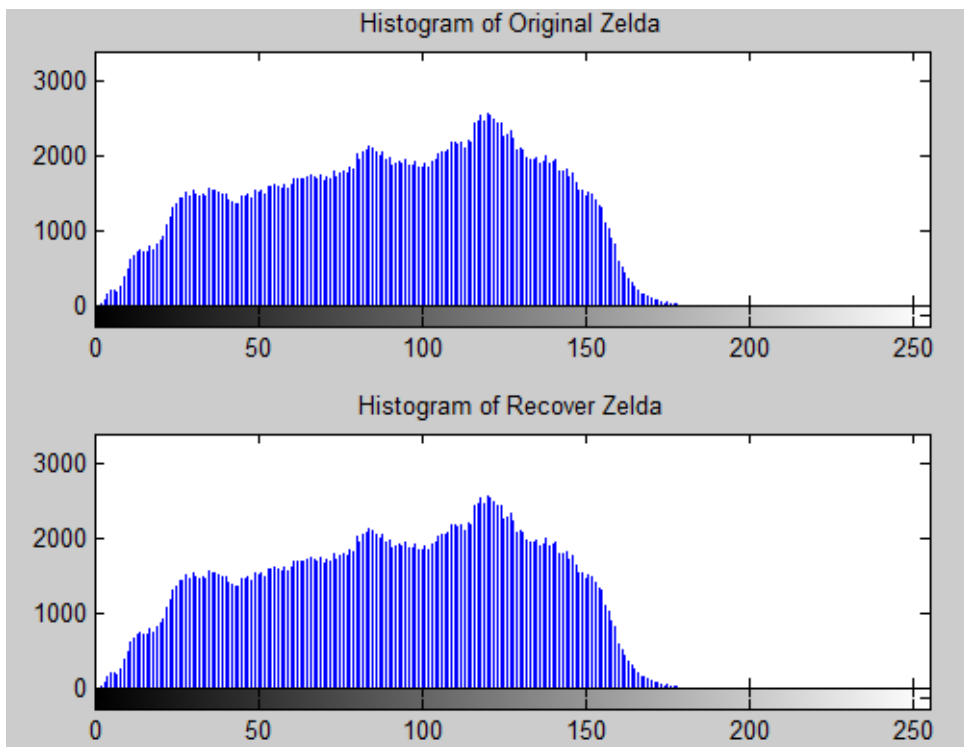


Figure 4.49 Histogram of Original Zelda and Recover Zelda

4.4 Discussion

The steganography images produced high quality in term of PSNR value. The average result PSNR of the four images is 52.6145dB which is higher than 37.3313dB as produced by M. Tang, J. Hu, W. Song and S. Zeng, 2015 which is highest PSNR among other steganography scheme reviewed. To ensure that image have embed image, message and key, the histogram analysis is carried on. The differences between original image and encrypted image can be visualized by analysing on the plotting pattern in the histogram.

There are four experiments to show the different text message file and user input secret key size. If the total bits to embed is larger than total available space of ROI2 or RONI2, it just only embedded the first 2601pixels. If the total bits to embed is equal or less than total available space of ROI or RONI, it will embed all pixels. User input secret key into RONI2, the receiver will request to extract the key from RONI and enter the key to access for extract the message from the ROI2.

Before embedded the original bits of ROI1 to RONI1, the original bits of RONIs were being stored in another storage image. Next, the original bits of ROI1 only embedded into RONI1. After key and message extraction, the ROI was reversed by take the ROI original bits that were store in RONI1 previously. Then, RONI1 and RONI2 were reversed by take the RONIs original bits of RONIs that were store in selected storage image in advanced. The recovers of ROI and RONI were successfully conducted. This was verified by calculated by MSE, PSNR and show visual difference in histogram in Figure 4.46 to Figure 4.49. Figure 4.50 shows the result of the reversed Lena image. The MSE value of the reversed image is calculated was zero. Therefore, PSNR value show infinity value because zero MSE and this show that the reversed image is 100% same to the original image.

```
Command Window
>> checkImage

MSE: 0.0000000 dB

PSNR:      Inf dB
```

Figure 4.50 PSNR and MSE of reversed image of Lena

CHAPTER 5

CONCLUSION

5.1 Introduction

This chapter include section 5.2 where make a conclusion for the steganography and reversible steganography. Section 5.3 listed out the limitations of this research. Section 5.4 discuss the future work based on the result of this research. Section 5.5 is about summary of this chapter.

5.2 Conclusion

ROIs and RONIs image are selected with the X-coordinate and Y-coordinate for embedded image and message. The user can select the position to select the position to embed the storage image. The size of the ROIs and RONIs will affect the available space to embed. This makes the embedding capacity is link with the selected size of ROIs and RONIs.

For the reversible process, the original bits of ROI were stored in RONI and the original bits of RONIs were stored in its respectively storage image. This method can reverse the stego-image to original image.

In the conclusion, the proposed method provides better flexibility of selecting ROI and RONI and able to recover original image.

5.3 Research Constraint

The limitations of this research are shown as below:

- a) Select ROIs and RONIs image spend time. User need to select the X-coordinate and Y-coordinate pixel by pixel. User needs to calculate the size of ROIs and RONIs to make sure the bits to embed is fit into the total available space.
- b) Coordinates of ROIs and RONIs must sent to the receiver. Receiver can't extract the encrypted things in the image and recover the original image when receiver didn't have the coordinates of ROIs and RONIs.
- c) Not able to recover area outside ROIs and RONIs. The embedding capacity limited of ROIs and RONIs. The embedding capacity of RONI1 is 940,240bits, therefore the bits of ROI1 cannot more than 940,240bits.
- d) The embedding capacity of character must be fit into the available capacity of ROI2 and RONI2. If the secret message is too long, only the first text message able can be embedded.

5.4 Future Work

According to the result of this research, there have some improvements can be performed in the future work. The future works are listed as below:

- a) Reduce time for user to select the ROIs and RONIs by plotting the coordinate of the image instead of finding the coordinate of the image.
- b) Find the method of check the extracted message is same with the original message or not. This will ensure the receiver can get the correct message.
- c) Reduce limitation of the size of ROI so that can embed the larger message into the ROI.

5.5 Summary

The objectives of this research in Chapter 1 have been achieved. In this chapter, the contribution and limitation of this research have been listed out. The future work also has been listed out for improve the research.

REFERENCES

- Akinola, S. O., & Olatidoye, A. A. ON THE IMAGE QUALITY AND ENCODING TIMES OF LSB, MSB AND COMBINED LSB-MSB STEGANOGRAPHY ALGORITHMS USING DIGITAL IMAGES.
- Atawneh, S. (2006). *A new algorithm for hiding gray images using blocks*. Paper presented at the Information and Communication Technologies, 2006. ICTTA'06. 2nd.
- Bhallamudi, S. (2015). Image Steganography Final project–Report.
- Hu, J., & Li, T. (2015). Reversible steganography using extended image interpolation technique. *Computers & Electrical Engineering*, 46, 447-455.
- Kaur, S., & Shukla, M. (2014). Reversible data hiding and its methods: a survey. *International Journal of Computer Science and Mobile Computing*, 3(5), 821-826.
- Lee, C.-F., & Huang, Y.-L. (2012). An efficient image interpolation increasing payload in reversible data hiding. *Expert Systems with Applications*, 39(8), 6712-6719.
- Sajna, U. (2014). LSB STEGANOGRAPHY BASED REVERSIBLE DATA HIDING'. *International Journal of Advances in Engineering & Technology*, 7(1), 105-112.
- Sarkar, T., & Sanyal, S. (2014). Reversible and irreversible data hiding technique. *arXiv preprint arXiv:1405.2684*.
- Tamimi, A. A., Abdalla, A. M., & Al-Allaf, O. (2013). Hiding an image inside another image using variable-rate steganography. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 4(10).
- Tang, M., Hu, J., & Song, W. (2014). A high capacity image steganography using multi-layer embedding. *Optik-International Journal for Light and Electron Optics*, 125(15), 3972-3976.

- Tang, M., Hu, J., Song, W., & Zeng, S. (2015). Reversible and adaptive image steganographic method. *AEU-International Journal of Electronics and Communications*, 69(12), 1745-1754.
- Walia, E., Jain, P., & Navdeep, N. (2010). An analysis of LSB & DCT based steganography. *Global Journal of Computer Science and Technology*.
- Wu, H.-T., Dugelay, J.-L., & Shi, Y.-Q. (2015). Reversible Image Data Hiding with Contrast Enhancement. *IEEE Signal Process. Lett.*, 22(1), 81-85.
- Zeng, X.-t., & Li, Z. (2012). Reversible data hiding scheme using reference pixel and multi-layer embedding. *AEU-International Journal of Electronics and Communications*, 66(7), 532-539.
- Zhang, Z., & Zhang, W. (2015). *Reversible steganography: Data hiding for covert storage*. Paper presented at the Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2015 Asia-Pacific.

APPENDIX A GANTT CHART

	Task Mode	Task Name	Duration	Start	Finish	1st Quarter Q1	2nd Quarter Q2	3rd Quarter Q3	4th Quarter Q4	1st Quarter Q1
1	★	▸ Planning Phase	23 days	Wed 2/21/18	Fri 3/23/18					
2	★	Identify problem statement based on title	3 days	Wed 2/21/18	Fri 2/23/18					
3	★	Define on the project scope and objective	3 days	Fri 2/23/18	Tue 2/27/18					
4	★	Submission of Chapter 1	1 day	Mon 3/5/18	Mon 3/5/18					
5	★	Do research on the project title background	5 days	Sun 3/4/18	Thu 3/8/18					
6	★	Understanding Steganography and Reversible Steganography	2 days	Thu 3/8/18	Fri 3/9/18					
7	★	Compare existing method and result	3 days	Fri 3/9/18	Tue 3/13/18					
8	★	Submission of Chapter 2	1 day	Tue 3/13/18	Tue 3/13/18					
9	★	Correction on Chapter 1 and Chapter 2	8 days	Wed 3/14/18	Fri 3/23/18					
10	★	▸ Analysis Phase	31 days	Sat 3/24/18	Fri 5/4/18					
11	★	Identify ROI and RONI process	7 days	Sat 3/24/18	Mon 4/2/18					
12	★	Draw flowchart of embedding and extraction process	10 days	Mon 3/26/18	Fri 4/6/18					

	Task Mode	Task Name	Duration	Start	Finish	1st Quarter Q1	2nd Quarter Q2	3rd Quarter Q3	4th Quarter Q4	1st Quarter Q1
13	★	Submission on Chapter 3	1 day	Mon 4/9/18	Mon 4/9/18					
14	★	Correction on Chapter 3	18 days	Tue 4/10/18	Thu 5/3/18					
15	★	Submission of PSM1 proposal	1 day	Fri 5/4/18	Fri 5/4/18					
16	★	Implementation Phase	90 days	Mon 5/7/18	Fri 9/7/18					
17	★	Implement the algorithm to Matlab	3 mons	Mon 5/7/18	Fri 7/27/18					
18	★	Test the algorithm with sample image	3 mons	Mon 5/7/18	Fri 7/27/18					
19	★	Take the result and compare	3 mons	Mon 6/18/18	Fri 9/7/18					
20	★	Documentation Phase	74 days	Sat 9/1/18	Wed 12/12/18					
21	★	Write result on Chapter 4	30 days	Sat 9/1/18	Thu 10/11/18					
22	★	Submission on Chapter 4	1 day	Fri 10/12/18	Fri 10/12/18					
23	★	Correction on Chapter 4	10 days	Sat 10/13/18	Thu 10/25/18					
24	★	Submission on Chapter 5	1 day	Fri 11/16/18	Fri 11/16/18					
25	★	Correction on Chapter 5	10 days	Sat 11/17/18	Thu 11/29/18					
26	★	Conclusion Phase	1 day	Thu 11/29/18	Thu 11/29/18					
27	★	Submission PSM2	1 day	Wed 12/12/18	Wed 12/12/18					