An Enhanced Least Significant Bit Steganographic Method For Information Hiding

Gabriel Macharia Kamau

A thesis submitted in partial Fulfillment for the degree of Master of Science in Software Engineering in the Jomo Kenyatta University of Agriculture and Technology

2013

DECLARATION

This thesis is my original work and has not been presented for a degree in any other University.

Signature:______________________________ Date:______________________________

Gabriel Macharia Kamau

This thesis has been submitted for examination with our approval as University Supervisors.

Signature:______________________________ Date:______________________________

Dr. Stephen Kimani

JKUAT, Kenya

Signature:______________________________ Date:______________________________

Dr. Waweru Mwangi

JKUAT, Kenya

DEDICATION

I dedicate this work to God for His grace and mercy in providing for me and enabling me to successfully pursue the course.

ACKNOWLEDGMENTS

My sincere acknowledgments and gratitude go to my supervisors Dr. Stephen Kimani and Dr. Waweru Mwangi for their tireless efforts, professional guidance and input throughout this study and research.

I wish also to acknowledge my lecturers for their input in class and for allowing God to use them during my course work for the Master of Science in software engineering.

I also express my gratitude to the University for providing the requisite environment and atmosphere to enable me carry out my studies.

Last but not least i thank all my colleagues and classmates for their encouragement, comradeship and support throughout the course.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABREVIATIONS

AAD	Average absolute difference
API	Application Programming Interface
HS	Histogram Similarity
IEEE	Institute of Electrical and Electronics Engineering
LCG	Linear Congruential Generator
LSB	List Significant Bit
MSE	Mean Square Error
MSM	Mid Square method
PNG	Portable Network Graphics
PSNR	Peak Signal to Noise Ratio
PRNG	Pseudo-random number generator
PVD	Pixel Value Differencing
RS	Reed Solomon
SSIS	Spread Spectrum Image Steganography
SPA	Sample Pair Analysis

LIST OF APPENDICES

DEFINATION OF OPERATION TERMS

Stegosystem A complete design for the implementation of steganography which defines all the stages involved in the steganographic process as well as their underlying concepts and techniques.

Cover Medium The input medium of the embedding process that will host the embedded data.

Embedded Data The input data or secret information hidden within the cover-medium.

Embedding The process of hiding data within a cover-medium.

Extraction The process of retrieving the embedded data from a stego-medium.

Steganalysis The deliberate and systematic attempt to detect the existence of hidden data in a cover-medium through statistical or computer-based examinations. It is an attempt to defeat a stegosystem.

Stego medium The output medium of the embedding process that hosts the embedded data.

Stego key The security key used in the embedding process, which is required in order to successfully extract the embedded data from the stego-medium.

ABSTRACT

The least significant bit (LSB) insertion method is a simple steganographic algorithm that takes the least significant bit in some bytes of the cover medium and swaps them with a sequence of bytes containing the secret data in order to conceal the information in the cover medium. However its imperceptibility and hiding capacity are relatively low. This is as revealed by the statistical characteristics of its resultant stego images compared to the original cover images.

To increase the level of imperceptibility and the hiding capacity in the LSB insertion method, this research proposes an enhanced LSB method that employs a selective and randomized approach in picking specific number of target image bits to swap with the secret data bits during the embedding process. To facilitate the selective picking of the target image bits, a variation of the standard minimal linear congruential pseudo random number generator (LCG) is used. The message digest (digital signature) of a user supplied password is used to seed the LCG and to extract the message from the cover medium.

In measuring the effectiveness of the proposed method, the study adopted an experimental research design where the statistical characteristics of the proposed method stego images were compared with those of the traditional LSB method in a comparative experiment designed to establish the levels of image distortion (noise) introduced in the original cover image when either of the methods is used under the same payload and image.

The experiment results indicated improved levels of imperceptibility and hiding capacity in the proposed method.

CHAPTER ONE

1.0 INTRODUCTION

This chapter provides an introduction and background of this study. It defines the main motivations for researching image based steganography and in particular the LSB steganography method. Additionally, this chapter outlines the main research objectives and attempts to propose a solution to challenge the start of the art.

1.1 Background to the study

The explosive and unprecedented growth in information communication technologies in the last one decade has brought about a shift in the way information and data is stored and retrieved. From homes to offices to the cyberspace, information is currently processed, stored, retrieved and transmitted electronically. Consequently the safety and security of information and data has become a fundamental issue of concern. Steganography, a technique used to conceal the presence of secret data in innocent looking containers like digital images and video files comes in handy particularly in open systems environments like the internet and other computer networks where secure links are not used and thus making information in transit vulnerable to interception and attacks. According to Mohammad and Abdallah (2008), Steganography "is the art and science of writing hidden messages inside innocent looking containers such as digital files, in such a way that no one apart from the sender and intended recipient realizes the existence of a hidden message". The secret message is normally embedded in a cover medium known as a stego file in a way that totally conceals the existence of any form of communication. Besides hiding important data for safety and confidentiality, this technique can also be applied in copyright protection for digital media including audio, video and images.

Digital images are the most widely used cover files in the world of digital steganography. The reason for this is pretty obvious as there is hardly a PC in the world today that does not make use of one image or the other. An internet website is not complete without the use of an image or a company logo. When secret data is properly embedded in a digital image, the human visual system can hardly pick the difference between an original image and the one containing the hidden information.

The LSB approach is the most widely used steganographic algorithm to embed secret data into a carrier image. This technique embeds the bits of the secret message directly into the least significant bit plane of the cover-image following a deterministic sequence. The least significant bits of the carrier image are swapped with those of the secret information following a definite sequential pattern.

Though this makes this approach perfectly imperceptible to the naked eye, its vulnerability to statistical steganalysis is relatively high.

1.2 Problem statement

Although the LSB method exploits the weakness of the human vision sensitivity (HVS) to hide secret data in a cover medium in such way that the human eye cannot perceive it, the statistical characteristics of the resultant stago images reveal high levels of distortion of the original cover images compromising on the security of the hidden data. Its hiding capacity is also relatively low as it uses only three bits in a single pixel or one bit per color channel.

To enhance and increase both the imperceptibility and the hiding capacity of the LSB method, this research proposes an enhanced LSB method that makes use of varied number of bits per image color channel selectively picked across the entire image by use of the linear congruential random number generator. This is to help avoid the predictability of the places where the secret data is hidden in the carrier image thus enhancing imperceptibility by reducing the gap between the statistical characteristics of the original cover image and those of the stego image. The number of bits used per color channel can also be varied to accommodate more data and thereby enhance the hiding capacity.

1.3 Research Objectives

The main purpose of this study is to enhance imperceptibility and the hiding capacity of the conventional LSB steganographic data hiding method by altering the approach through which the insignificant bits of the cover medium are replaced with the significant bits of the secret data during the embedding process.

The following are the specific objectives of the study

i. To review literature related to digital image steganography as an information hiding technique.

ii. To investigate and study the perceptibility metrics of the traditional LSB digital image steganography method in information hiding systems.

iii. To investigate on the impact of using the image's varied and randomly selected bits in the LSB steganography method embedding process on imperceptibility and hiding capacity.

iv. To develop and test an LSB steganography data hiding system prototype that makes use of the images' varied and randomly selected bits during the embedding process.

1.4 Research Questions

i. What is the current literature regarding steganography and the LSB steganography method.

ii. What are the perceptibility metrics of the existing implementations of the traditional LSB digital image steganography method in information hiding?

iii. What is the impact of using the image's varied and randomly selected bits in the LSB steganography method embedding process on the stego image imperceptibility and hiding capacity?

iv. What steganographic data hiding system prototype best implements an LSB steganography data hiding method that makes use of the image's varied and randomly selected bits during the embedding process.

1.5 Justification

According to Provos and Honeyman (2001), if a steganography method causes someone to suspect the carrier medium, then the method has failed. Given the confidentiality and security challenges brought about by the sheer volume of data stored, retrieved and electronically communicated daily world over, there is a high need to continuously improve on data security algorithms. There is therefore need to strengthen steganographic data hiding algorithms and methods to improve on both their imperceptibility against steganalysis and also on their data hiding capacity.

1.6 Scope

This research concentrates on the application of digital image steganography in data hiding systems. The technique presented in this study is not necessarily suitable for other cover media as it is specifically designed for digital images. The main concern of the proposed steganographic software is to improve imperceptibility and data hiding capacity in the LSB steganography data hiding method.

CHAPTER TWO

2.0 LITERATURE REVIEW

2.1 Introduction

In this chapter, an overview of the history of steganography as a technique of Information hiding is presented. Applications of steganographic techniques in information hiding are discussed. The different file types that can be used as cover media for digital steganography are described, and the main components of digital steganographic systems identified. The least significant bit method is explained and related work discussed. The steganalysis attacks that can be used to defeat steganography are also classified and explained. Finally the enhanced LSB method based on use of varied and randomly selected cover image bits is presented as the proposed improvement to the traditional LSB method. Increased imperceptibility and hiding capacity in data hiding is the basic concept of the proposed method.

2.2 History of steganography

The term steganography comes from the Greek word *Steganos*, which means covered or secret and *–graphy* which means writing or drawing. Literally therefore, Steganography means, covered writing. It is the art and science of writing hidden messages inside innocent looking containers such as digital files, in such a way that no one apart from the sender and intended recipient realizes the existence of a hidden message (Mohammad and Abdallah, 2008). Although many different cover files are used, digital images are the most frequently used due their proliferation on the internet and other open system environments. The objective of steganography is not to keep others from knowing the hidden information, but to keep others from thinking that the information even exists (Provos and Honeyman, 2001)

Though its use in information technology and computer science fields is relatively new, steganography has been in use for thousands of years. The first recorded use of steganography dates back to 440 BC as mentioned in “The Histories of Herodotus”. In the 5th century BC Histaiacus shaved a slave’s head, tattooed a message on his skull and the slave was dispatched with the message to instigate a revolt against the Persians after his hair grew back (Moulin and Koetter, 2005). In ancient Persia (Circa 559 B.C), confidential and secret messages were placed inside a dead rabbit and delivered using a man disguised as an ordinary hunter.

While technology has progressed, some techniques have remained the same. Aeneas used a method of putting small pin pricks above the letters of a cover text to represent letters of confidential message. This technique was also used in the World War I by Germans (Neil and Jajodia, 1998).

A well known example of steganography happened during the Vietnam War when American commander and naval aviator Jeremiah Denton, who had been captured by Vietnamese forces was paraded in front of the news media. He knew he would be unable to say anything against his captors so as he addressed the media, he blinked his eyes in Morse code, spelling the word

T-O-R-T-U-R-E (Denton, 2002)

For slightly over a decade now, a growing interest in the field of information hiding has been witnessed. In May 1998, the “First international workshop on Information hiding was held in Cambridge, UK. From then on, areas of information hiding like steganography and digital watermarking started to attract attention from the research community.

By 1998, the number of publications on digital watermarking alone increased from 2 in 1992 to 103 (Peticolas *et al.,* 1999).

Different research institutions published journal reports on this developing field. The Institute of Electrical and Electronics Engineers (IEEE) published many of these reports especially those in the area of steganography. Neil and jajodia's article on basic techniques for image steganography provided the first platform for understanding steganography (Neil and Jajodia, 1998).

In a special issue of the IEEE journal of selected areas in communications, Petitcolas published an article on limitations of steganography (Peticolas *et al.,* 1999). He also wrote a summary of various areas of information hiding and elaborated on several practical applications in the July 1999 issue of proceedings of the IEEE .

Realizing its significance in secret communications, the American Military also conducted various researches on steganography, one of which was lead by Lisa Marvel on Spread Spectrum Image Steganography (SSIS), which was sponsored by the US Army Research Laboratory (Marvel *et al.,* 1999). SSIS adopted concepts from spread spectrum communications to maximize embedding capacity and imperceptibility. The Air Force Research Laboratory, on the other hand, sponsored the work of Fridrich on steganalysis techniques for detecting LSB encoding in color images (Fridrich *et al.,* 2000).

2.3 Steganography and information hiding

Information hiding is an increasingly developing field that has grown to include applications such as watermarking, fingerprinting, copyright protection for digital media, and steganography (Isbell , 2002).

Figure 1 shows the basic classification of information hiding techniques.

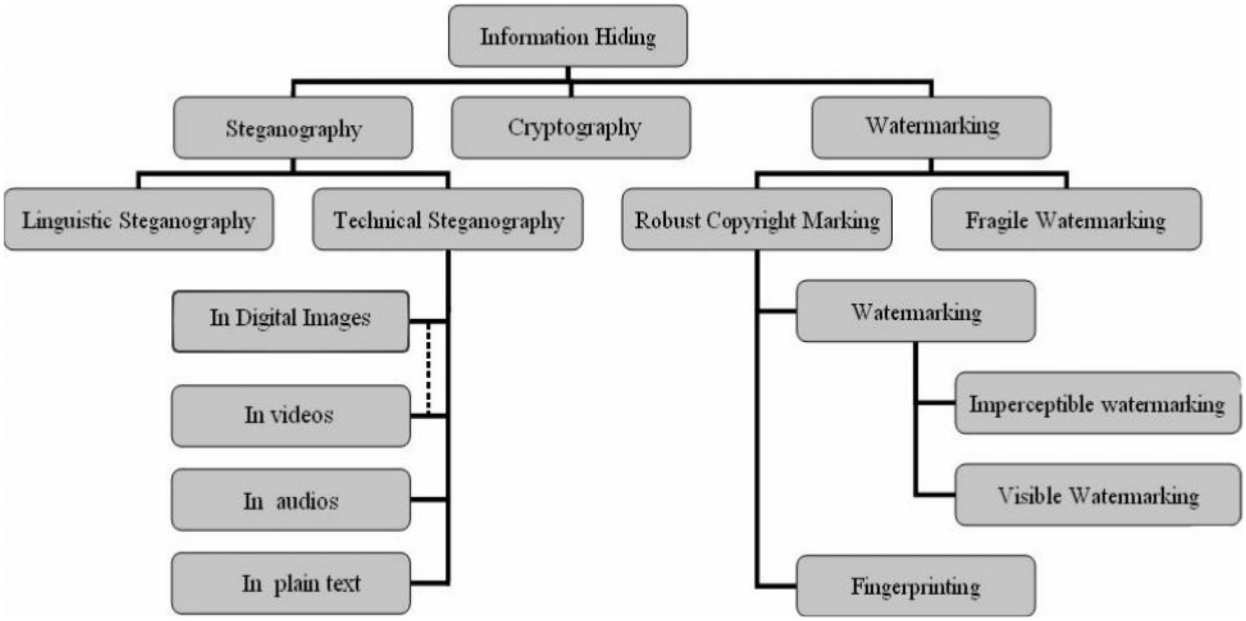


Figure 1 Classification of information hiding techniques

This research focuses on technical steganography in digital images.

2.4 Technical steganography

To secure secret messages in an open system environment, it is important to hide both its context and existence in order to conceal it from unintended recipients. Steganographic data hiding techniques help in implementing this fact (Morkel *et al.*, 2005). Steganography involves embedding information into a medium in such a way that it is not apparent to an observer. Technical steganography medium includes digital images, audio files and video files. Most technical steganography techniques however use images as stego-medium. The main goal of steganography is to communicate securely in a completely undetectable manner and to avoid arousing suspicion on the existence of hidden data in a cover medium (Neil and Jajodia, 1998). A basic technical steganographic system consists of a cover medium into which the secret message is embedded using a specific algorithm. The resultant is called the stego media (Sellars, 2006).

According to Krenn (2004), a typical steganographic technique basically includes the following steps.

i. Providing a file ie Cover Media in which the secret information is embedded.

ii. Providing a stego-key in order to get the output file ie the stego-media.

iii. Embedding the secret message file inside the cover media.

This is illustrated in the figure 2.

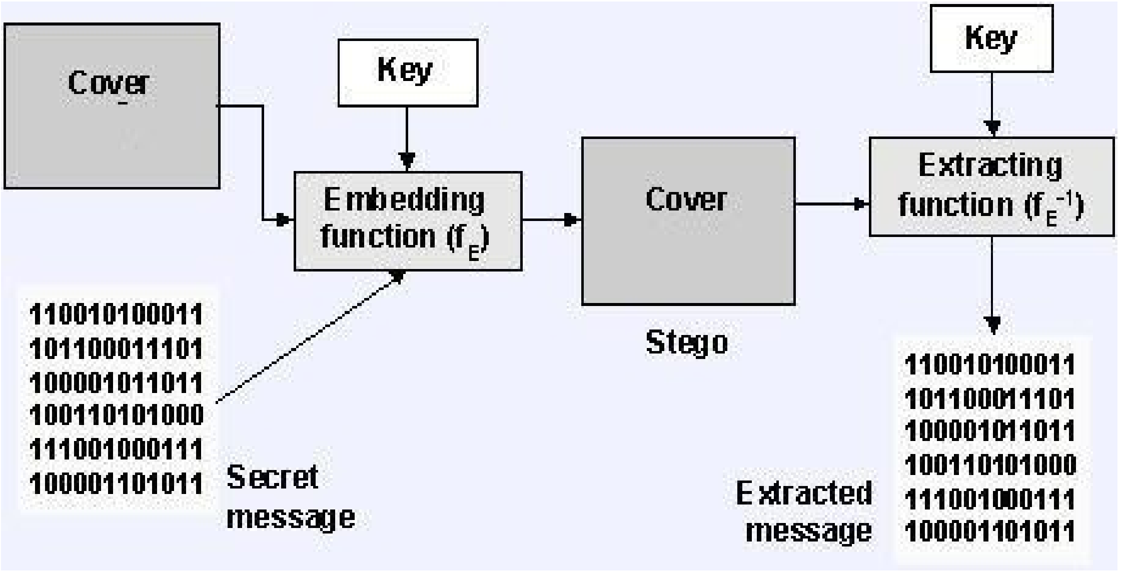


Figure 2 Overview of a steganographic system

The cover media can be any kind of image, audio or video file. However since in this research, the desired cover-file is a digital image, the name cover image will be used. The cover image changes to a stego image after the secret data is embedded into it. Therefore, the innocent-looking image used to hold the hidden information is referred to us a cover image and once the desired secret message is embedded and combined with cover-image; it makes the stego-image.

2.5 Digital steganographic techniques in information hiding

According to Cox (2009), there are three most popularly researched techniques of digital steganography in information hiding. These are:

i. Covert channels

ii. Embedded data

iii. Digital watermarking

2.5.1 Covert channels

Covert channels in TCP/IP involve masking identification information in the TCP/IP headers to hide the true identity of one or more systems. This is normally very useful for any secure communication need over open systems such as the Internet when complete secrecy is needed for an entire communication process.

2.5.2 Embedded data

This refers to the use of innocent containers (cover files) to hide and send information. It is by far the most popular use of Steganography today. This method of Steganography is very useful when a party must send top secret data, private or highly sensitive document over an open system environment such as the Internet. By embedding the hidden data into the cover file and sending it, one gains a sense of security in that no one is aware of the sent information other than the intended recipient.

2.5.3 Digital watermarking

Digital watermarking is usually used for copy write reasons by organizations or entities that wish to protect their property by either embedding their trademark into their property or by concealing serial numbers/license information in software, etc. Although not a pure steganographic technique, digital watermarking use Steganographic techniques to embed information into documents. It is also very important in the detection and prosecution of software pirates.

General applications of Steganography in modern digital information system include:

i. Copyright control.

ii. Smart IDs where individuals' details are embedded in their photographs

iii. Medical imaging systems where a separation is considered necessary for confidentiality between patients' images and their captions e.g., Physician, Patient's name, address and other particulars etc (Petitcolas, 2000).

2.6 Digital steganography cover media

Four main types of cover files are commonly used in digital steganographic system. These are text, digital images, audio and video files as depicted in figure 3.

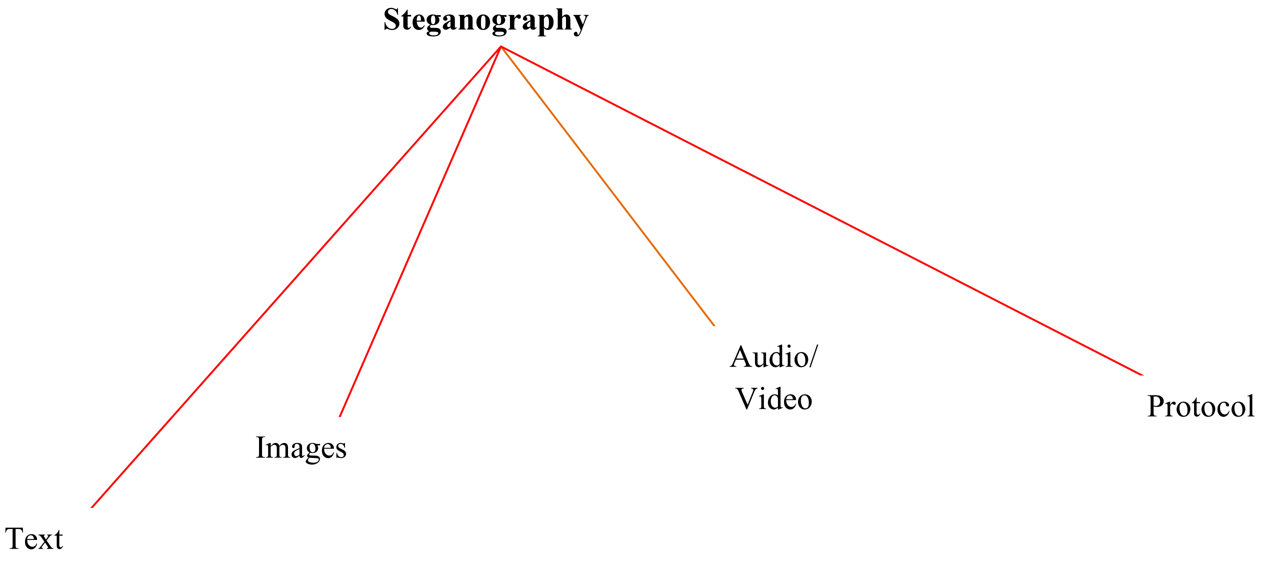


Figure 3 Categories of the steganography cover media

The factor however which is worth considering is the redundancy of the medium (Currie and Irvine, 1996). The redundant bits of an object are those bits that can be easily changed with an almost insignificant change been made in the medium (Anderson and Peticolas, 1998).

2.7 Steganographic techniques for embedding data in digital images

Currently there exist several techniques of hiding secret information in a digital cover medium. All these technique have different degree of success (Neil and Jajodia, 1998).

According to (Clair , 2001), the following are the three basic data embedding techniques as used in digital steganography.

2.7.1 Substitution

The substitution technique replaces the desired Least Significant Bits of the original image file with the bits of the secret information in a way that does not reveal any distortion to the original file. Substitution technique is used in the conventional Least Significant Bit algorithm.

2.7.2 Injection

This technique adds bits to unused sections of digital files to hide the secret message. This way, file bits relevant to an end-user are not modified leaving the cover file perfectly usable. The end-user may not even realize that the file contains additional hidden information.

2.7.3 Generation

Unlike injection and substitution, this technique does not require an existing cover file. This technique generates a cover file for the sole purpose of hiding the message.

2.8 Methods and approaches of steganography in digital images

According to (Lee and Cheng, 2000), the three most common implementation approaches used to hide information in digital images are:

i. Masking and filtering

ii. Transform embedding

iii. Least significant bit insertion

2.8.1 Masking and filtering method

This approach hides information by "marking an image in a manner similar to paper watermarks". They are therefore designed for applications in digital watermarking (Neil and Jajodia, 1998). Since watermarks become more integrated into the image, Digital watermarking techniques are more robust than traditional steganographic techniques. An example of digital watermarking is the Patchwork algorithm which embeds data by changing the brightness of certain pairs of points in an image (Bender, 1996).

2.8.2 Transform embedding method

Here, instead of manipulating the pixels or the spatial domain of an image the coefficients of the transform domain of a processed image are used (Lin and Delp*,* 1999*)*. This is because transformations employed by many image file formats results in loss of data.

Transform embedding techniques typically offer less hiding capacity. They are also format-dependent and embed data after the transformations have been made. This means the embedding process may alter certain statistical properties of the image that are common or unique to that particular image format. The technique is therefore more susceptible to steganalysis attacks and is only ideal for applications like copyright marking and authentication.

2.9 The least significant bit (LSB) method

The traditional LSB method is a common, simple approach for embedding information in a cover image. It is the simplest and most commonly used digital image steganographic method and is also relatively easy to implement (Neil and Jajodia, 1998). This method substitutes cover image least significant bits with the secret message bits in a deterministic and sequential manner ie it replaces only the least-significant bits (LSB) of the cover image with bits from the file that is to be embedded. The concept behind the algorithm is to replace the least significant bit (LSB) of each pixel in an image with the bits of the data to be hidden or embedded. It is based on the idea that since the least significant bit has a place value of 1, modifying it would result in a maximum difference of only 1. Because the human eye is unable to distinguishing minimal changes in color, such modifications would normally be imperceptible.

The embedding process consists of choosing a subset $\{j1,\ldots, j_l(m)\}$ of cover elements and performing the substitution operation:

$LSB(C_j) = m_i$ (m_i can be either 1 or 0). Where j represents cover image bits and i represents the secret message bits.

ALGORITHM

```
Input: Cover-Object C
for i=1 to length ($m$) do
    Compute the $j^{th}$ cover image index where to store the $i^{th}$ message bit of $m$
    LSB ($C_j$) = $m_i$
                                                                        used
end for
Output Stego-Object S
```

ALGORITHM

```
Input: Stego-Object S
for i=1 to length ($m$) do
    Compute the $j^{th}$ cover image index where the $i^{th}$ message bit of $m$ is stored
    $m_i$ = LSB ($C_j$)
end for
Output Message $m$
```

As Bender (1996) explains "To a computer, an image is an array of numbers that represent light intensities at various points, or pixels. These pixels make up the images raster data." When dealing with digital images for use with Steganography, 8-bit and 24-bit per pixel image files are typical. The least significant bit (in other words, the 8th bit) of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the Red, Green and Blue color components can be used, since they are each represented by a byte. One can therefore store 3 bits in each pixel. An 800 × 600 pixel image, can therefore store a total amount of 1,440,000 bits or 180,000 bytes of embedded data (Krenn, 2009). For example a grid for 3 pixels of a 24-bit image could be represented as shown in figure 4.

0	0	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	1	0	1	1	1	0	0
1	0	1	0	0	1	1	0	1	1	0	0	0	1	0	0	0	0	0	0	1	1	0	0
1	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	1	0	0	0	1	1

Figure 4 Original bits

When the number 200, whose binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as shown in figure 5.

0	0	1	0	1	1	0	1	0	0	0	1	1	1	0	1	1	1	0	1	1	1	0	0
1	0	1	0	0	1	1	0	1	1	0	0	0	1	0	1	0	0	0	0	1	1	0	0
1	1	0	1	0	0	1	0	1	0	1	0	1	1	0	0	0	1	1	0	0	0	1	1

Figure 5 Modified bits

Though the number has been embedded into the first 8 bytes of the grid, only the three highlighted bits have been changed. Mostly, only half of the bits in an image will need to be changed to hide secret data using the maximum cover size. Since there are 256 possible intensities of each primary color, changing the LSB of a pixel results in small changes in the intensity of the colors. These changes cannot be perceived by the human eye - thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the least as well as the second to least significant bit and still not see the difference (Neil and Jajodia, 1998).

2.10 Cover images used in the least significant bit method

According to (Krenn, 2004), in its simplest form, LSB method makes use of BMP images, since they use lossless compression. Unfortunately to be able to hide a secret message inside a BMP file, one would require a very large cover image. Nowadays, BMP images of 800 × 600 pixels are not often used on the Internet and might arouse suspicion. For this reason, LSB steganography method has also been improved to use other image file formats.

GIF

These are indexed images where the colors used in the image are stored in a palette, sometimes referred to as a color lookup table. A GIF image cannot have a bit depth greater than 8, thus the maximum number of colors that a GIF can store is 256. Each pixel is represented as a single byte and the pixel data is an index to the color palette. GIF images can also be used for LSB steganography, although extra care should be taken, for should one change the least significant bit of a pixel, it can result in a completely different color since the index to the color palette is changed (Neil and Jajodia, 1998).

JPEG

Initially it was thought that steganography would not be possible to use with JPEG images. This is because information is hidden in the redundant bits of an object and since redundant bits are left out when using JPEG it was feared that the hidden message would be destroyed. Though one could somehow keep the message intact, it would be difficult to embed the message without the changes being noticeable because of the harsh compression applied. However, properties of the compression algorithm have been exploited in order to develop a steganographic algorithm for JPEGs.PEG. Table 1 shows the performance of various image formats using LSB algorithm.

	LSB in BMP	**LSB in GIF**	**LSB in JPEG**
Invisibility	High (*Depends on cover image*)	Medium(*Depends on cover image*)	High
Payload Capacity	High	Medium	Medium
Robustness against statistical attacks	Low	Low	Medium
Robustness against image manipulation	Low	Low	Medium
Independence of file format	Low	Low	Low
Unsuspicious files	Low	Low	High

Table1 Comparison of various image file formats performance of traditional LSB

2.11 The pros and cons of the conventional LSB algorithm

One of the major advantages of the conventional LSB steganography method is the simplicity in which it embeds the bits of the secret message directly into the LSB plane of cover-image (Provos and Honeyman, 2009). Many techniques use this method (Chandramouli and Memon, 2001). Embedding using the LSB method does not result in a human-perceptible difference because the amplitude of the change is small. To the human eye, the resulting stego-image will look identical to the cover-image. This allows high perceptual transparency of LSB.

The major weakness of the LSB algorithm which forms the main thrust and focus for this study is its limited hiding capacity and also imperceptibility level. Though it is fairly imperceptible to visual attacks, the statistical characteristics of the resultant stego images reveal high levels of distortions compared to the original cover images. In terms of hiding capacity, the size of information to be hidden relatively depends on the size of the cover-image and therefore the message size must always be smaller than the image. A large capacity within the required fidelity could allow the use of a smaller cover-image for the message of fixed size, thereby decreasing the bandwidth required to transmit the stego-image (Cachin, 2008).

2.12 Existing enhancements of the traditional LSB algorithm

The existing enhancements of the LSB method found in literature are as follows:

i. The Optimal LSB Insertion Method

ii. The Pixel Value Differencing (PVD) Method

iii. Blind Hide algorithm

iv. Filter First algorithm

v. Algorithm Pixel Swap

2.12.1 The optimal LSB insertion method

This insertion method improves the stego-image quality by finding an optimal pixel after performing an adjustment process. Three candidates are picked out for the pixel's value and compared to see which one has the closest value to the original pixel value with the secret data embedded in. The best candidate is then called the optimal pixel and used to conceal the secret data (Chan and Cheng, 2004). This however makes the hiding capacity of the carrier image very low.

2.12.2 The pixel value differencing (PVD) method

The pixel-value differencing (PVD) method is proposed by (Wu and Tsai, 2003). In this approach, the payload of each individual pixel is different, and the resultant stego-image quality is extremely fine with perfect modification and invisibility. The resultant stego-images quality that the method produces is better in terms of human visual perception. However steganalysis is easy as the hidden message is not well spread across the entire image.

2.12.3 Blind hide algorithm

According to (Bailey and Curran, 2006), this algorithm blindly hides the secret data in the image starting at the top left corner of the image and working its way across the image (then down - in scan lines) pixel by pixel changing the least significant bits of the pixel colors to match the message. To extract the hidden information, the least significant bits starting at the top left are read off. This embedding procedure is not very secure as it's really easy to read off the least significant bits starting from the top left corner of the image sequentially.

2.12.4 Filter first

As discussed by Umamaheswari *et al*. (2010), this algorithm filters the image using one of the inbuilt filters. It filters the most significant bits, and leaves the least significant bits to be changed for the purpose of hiding the secret data. Because the pixels are changed, there is needed to be careful about filtering the picture because one may use information for filtering that might change. If this happens, then it may be difficult (if not impossible) to retrieve the message again.

To extract the hidden message, the least significant bits starting at the top left are read off. This is not very secure since the message is not completely spread across the image resulting to only a portion of the image being degraded and hence making steganalysis easy.

2.12.5 Algorithm pixel swap

This method is proposed by Lee *et al.* (2010). This works as follows

i. Randomly select 2 pixels *x1* and *x2* from the cover image using a pseudo–random sequence.

ii. If the two pixels lie within a specified distance α (α=2 or 3 generally), they are suitable for embedding, otherwise generate another set of pixels.

iii. Take the specific message bit to hide. If the message bit is zero, check if x1 > x2 otherwise swap x1 and x2 and hide the bit in the LSB of the pixel. Do the reverse operation if the message bit is one.

iv. For extracting the hidden message, select the pixels using the same pseudo-random sequence. Check if the 2 pixels are within the pre-specified range α. If x1>x2, the message bit is zero (one) otherwise the message bit is one (zero).

This method does not add visible distortions to the cover image since only one bit is changed per pixel but its hiding capacity is highly limited. An implementation of this is Hermatic stego version 9.3

2.13 Imperceptibility requirement for steganography methods

All steganographic algorithms and methods must comply with a few basic requirements. The most important requirement is that a steganographic algorithm has to be imperceptible. In order to conceal the existence of hidden information, it is important that the embedding process does not produce perceptible distortions to the cover-medium. Bender related this concept to the magician's trick of misdirection, which allows "something to be hidden while it remains in plain sight" (Bender, 1996).

Imperceptibility also defends steganography against computer-based steganalysis by preserving certain statistical characteristics of the cover-medium. It is concerned with the stego-medium's consistency with the statistical characteristics of the original cover-medium (Fridrich *et al.*, 2000).

Authors propose a set of criteria to further define the imperceptibility of a steganography method. These are outlined below:

2.13.1 Invisibility

This is the most important requirement for an imperceptible steganographic algorithm since the strength of steganography lies in its ability to be unnoticed by the human eye.

If one is able to notice that an image has been tampered with, the algorithm is already compromised.

2.13.2 Payload capacity

Payload capacity or the embedding capacity was defined by Lin and Delp as “the size of information that can be hidden relative to the size of the cover” (Lin and Delp*,* 1999). It is the amount of secret information that can be hidden in a cover image without compromising imperceptibility. A good steganographic algorithm should have high payload capacity without compromising imperceptibility.

2.13.3 Robustness against statistical attacks

Statistical steganalysis is the process of detecting hidden information by applying statistical tests on stego image (Huaiqing, 2004). An imperceptible steganographic algorithm should not leave a signature when embedding information as this would be statistically significant in steganalysis.

2.13.4 Robustness against image manipulation

There is always a possibility that image manipulation, such as rotating, can be performed on the image before it reaches its destination. Depending on the manner in which the message is embedded, these manipulations may destroy the hidden message. It is preferable for steganographic algorithms to be robust against either malicious or unintentional changes to the image. However Marvel et al. stated that capacity and robustness is impossible to maximize at the same time while adhering to high imperceptibility rates (Marvel *et al.*, 1999).

W. Bender discusses this trade-off between capacity and robustness as the *data-hiding problem space* (Bender, 1996). He outlines that in order to achieve robustness, redundant encoding of the embedded data on the cover-medium must be performed, which in turn definitely compromises capacity.

This is illustrated in figure 6 based on Fridrich's diagram for the data-hiding problem space, which depicts the mutually competitive nature of these parameters (Fridrich *et al.*, 2000).

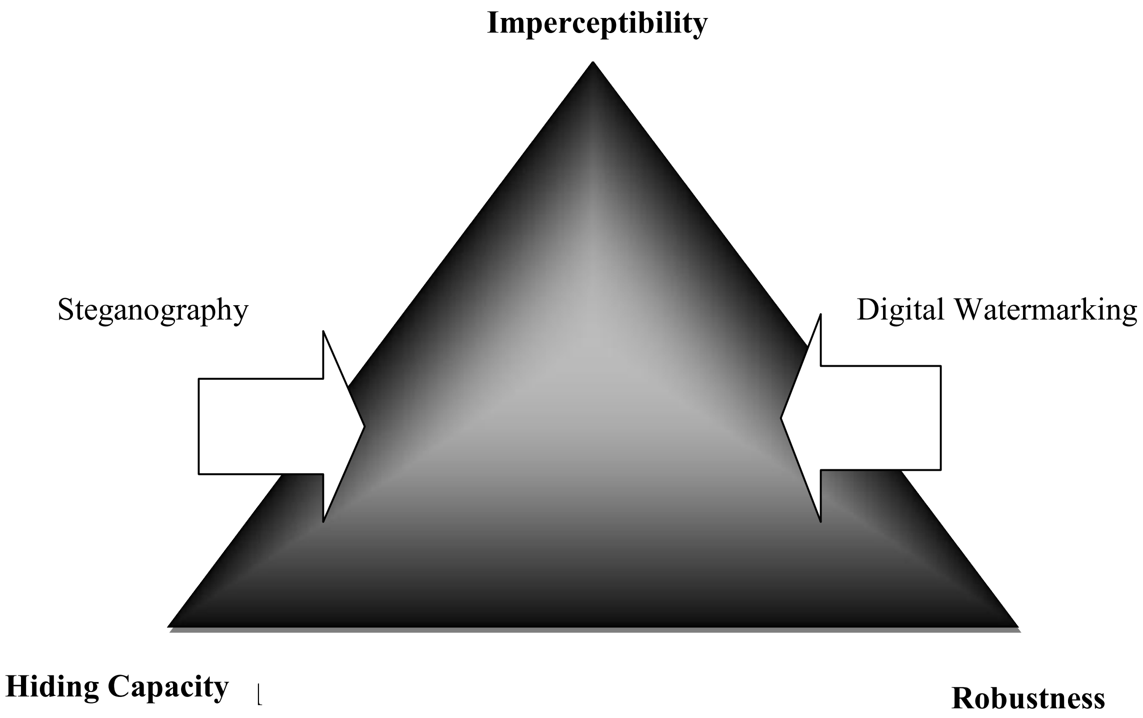


Figure 6 Data hiding problem space

As illustrated in the diagram, it is difficult to maximize the three opposing parameters all at the same time. For Steganography, working on the midpoint between imperceptibility and hiding capacity provides optimum balance between the two parameters but at the expense of robustness. On the other hand, digital watermarking systems compromise on capacity in favor of robustness, as is required by the application.

2.13.5 Independence of file format

Good steganographic algorithms must possess the ability to embed information in any type of file format. Eavesdroppers could be suspicious if only one type of file format is continuously communicated between two parties. This also gives one the liberty of any image format available instead of only relying on a specific format.

2.13.6 Unsuspicious files

All steganographic algorithms should use files in a manner that does not arouse suspicion. Abnormal file size, for example, is one property of an image that can result in further investigation of the image by an eavesdropper.

2.14 Steganalysis

According to Huaiqing (2004), Steganalysis is the practice of detecting hidden data or information through applying statistical tests on a stego image. It is the process of detecting steganographically hidden messages in cover files. This science is utilized to disrupt the transmission of Steganographic embedded messages, through detection, extraction, disabling or destruction of such hidden information. Steganalysis takes advantage of statistical or perceptual distinction between the stego objects and the original cover mediums.

Since secret and hidden information normally has a value to the ones who are not allowed to know it, people and or organizations will always try to retrieve information that is hidden from them (Ismail, 2003). Though the data hiding algorithms would naturally be ahead in technology, the techniques to decode the hidden information also grow.

Distinctly there exist two categories of attacks on steganography. These are:

i. Detection attacks

ii. Destruction attacks

2.14.1 Detection attacks

The purpose of a steganographic method is to hide information within a cover image. If it is possible for an attacker to determine if an image has hidden information, then the purpose of the steganographic method is defeated, and hence insecure. Such attacks are detection attacks. Detecting an embedded message defeats the primary goal of Steganography techniques, that is concealing the very existence of a message (Neil, 1998).

However Without knowing which algorithm is used and which Stego-key is used, detecting the hidden information is quite complex.

2.14.2 Destruction attacks

These concentrate on removing the hidden messages from the stego objects (Neil and Jajodia, 1998). According to Petitcolas (2000), the following are some of the known attacks for the Steganography in the two categories discussed above.

a) **Stego-only**

This is where only the stego media is available for analysis. The original cover file is not available.

b) **Known cover**

Here the cover media and the stego media are both available. An attack compares the original cover image with the stego image in an effort to detect pattern differences. If the cover image is a popular one, for example in images found in emails forwarded as a chain letter from one person to another, then a known cover attack could be mounted.

c) **Known message**

Here the embedded message is supposed to be known to the steganalyst. The steganalysis may use a known message attack i.e attacker may attempt to analyze the stego-object for future attacks. Even with the message, this may be very difficult and is equivalent to the stego-only attack (Huaiqing, 2004).

d) **Chosen stego attacks**

The chosen stego attack is one where both the steganography tool or algorithm and stego-object are known.

e) **Chosen message.**

A chosen message attack is one where the steganalyst generates stego-object from some Steganographic tool or algorithm from a known message. The goal in this attack is to determine corresponding patterns in the Stego object that may point to use specific Steganographic algorithms.

2.15 Proposed enhancement of LSB method imperceptibility

Imperceptibility and hiding capacity are clearly the most vital qualities in enhancing the least significant bit method and indeed any other steganographic method. This study proposes an LSB embedding approach that utilizes varied and randomly chosen bits from the cover image to hide the secret information.

The LCG algorithm can be used to assist in identifying the target bits in the cover image since given an initial state, the algorithm can produce a sequence of pseudorandom numbers. The generated sequence of numbers is repeatable, has known mathematical properties and can be implemented without the need of any specialist hardware. The main idea here as applied to this study is to generate a series of varied, random numbers of length equal to the secret message length. These numbers can then be used in targeting specific bits in the cover image for hiding the secret data. The number of bits used can be increased to improve on the hiding capacity provided the perceptibility metric levels of the stego image are within acceptable levels.

Proposed by D.H. Lehmer, the Linear Congruential Pseudorandom numberr Generator (LCG) is one of the most successful random number generators particularly with computer memory. It is also fast and easy to implement.

According to Donald Knuth, LCG can be used to compute each successive random number from the previous (Knuth, 1971).

Equation (1) shows the general formula for the linear congruential method:

$$\mathbf{X_{n+1} = (aX_n + c) \bmod m} \qquad \textbf{(1)}$$

Where:

X_0 is the starting value , the seed ; $0 <= X_0 < m$

a is the multiplier; $a \geq 0$

c is the increment; $c \geq 0$

m is the modulus; $m > X_0$, $m > a$, $m > c$

The desired sequence of random numbers $< Xn >$ is then obtained by setting

$X_{n+1} = (aX_n + c) \bmod m, \ n \geq 0$

X_n is chosen to be in [0, m-1], $n \geq 0$

In the proposed enhancement of the LSB method LCG can be used to generate the location of the next bit to be replaced with the bit of the secret information. For example, given that the previous random number was X_i, the next random number X_{i+1} can be generated as shown in equation (2) below.

$X_{i+1} = f(X_i, X_{i-1,\ldots,} X_{i-n+1})(\bmod m) = (a_i x_i + a_2 x_{i-1} + \ldots + a_n x_{i-n+1} + c)(\bmod m)$ **(2)**

To communicate the secret information, both communicating partners will share a special hushed stego key (k) which will be used to seed the LCG.

Example:

Suppose in our quest to generate random numbers for hiding data in a grid of three pixels the following values are assigned to the LCG equation (1) as follows:

$X_0 = 7$

$a = 7$

$c = 7$

$m = 9$

The following random numbers are then obtained as follows:

$X_0 = 7$

$X_1 = (aX_n + c) \bmod m$

$= (7*7+7) \bmod 9$

$= 2$

$X_2 = (aX_n + c) \bmod m$

$= (7*2+7) \bmod 9$

$= 3$

$X_3 = (aX_n + c) \bmod m$

$= (7*3+7) \bmod 9$

$= 1$

$X_4 = (aX_n + c) \bmod m$

$= (7*1+7) \bmod 9$

$= 5$

$X_5 = (aX_n + c) \bmod m$

$= (7*5+7) \bmod 9$

$= 6$

$X_6 = (aX_n + c) \bmod m$

$= (7*6+7) \bmod 9$

$= 4$

$X_7 = (aX_n + c) \bmod m$

$= (7*4+7) \bmod 9$

$= 8$

Therefore the generated numbers are:

7

2

3

1

5

6

4

8

These numbers can then be used to pick the bits in each color channel upon where to hide the bits of the secret data.

For example considering storing the 200, which binary representation is 11001000 in a grid of 3 pixels of a 24-bit image utilizing a single LSB of each color channel, the enhanced LSB algorithm will store the significant bits of the message randomly into the cover image bits as shown in figures 8.

0	0	1	0	1	1	0	1	0	0	0	1	1	1	0	0	1	1	0	1	1	1	0	0
1	0	1	0	0	1	1	0	1	1	0	0	0	1	0	0	0	0	0	0	1	1	0	0
1	1	0	1	0	0	1	0	1	0	1	0	1	1	0	1	0	1	1	0	0	0	1	1

Figure 7 Pixel bits before embedding

0	0	1	0	1	1	0	0	0	0	0	1	1	1	0	1	1	1	0	1	1	1	0	0
1	0	1	0	0	1	1	0	1	1	0	0	0	1	0	1	0	0	0	0	1	1	0	0
1	1	0	1	0	0	1	1	1	0	1	0	1	1	0	0	0	1	1	0	0	0	1	1

Figure 8 Pixel bits after embedding

Retrieving the message back requires an extraction key that was used during the encoding process. This key is then used to generate the positions of the bits where the secret data bits are hidden. The hiding capacity can be enhanced by varying the number of bits used per color channel.

The extraction key is that is used to seed the LCG is the message digest of the user supplied password.

2.16 Conclusion

The rapidly growing possibilities of modern digital communications urgently require the use of more secure means of protecting information during transmission against unauthorized access. The increased need for protection of digital media and content means that research in this area will continue to attract more attention in the years to come. Steganography, an approach of hiding secret data inconspicuously inside a host data set, creates possibilities of enhanced security in open system environments (Amin, 2003).

The LSB algorithm is the most simple and a straight forward approach in digital steganography. The message is embedded deterministically in the least significant bits of each pixel of the cover image. Although this algorithm embeds the secret data in an imperceptible way to the human eye, statistical analysis of its resultant stego images reveal high levels of distortion of the original cover images compromising on the security of the hidden data. Its hiding capacity is also relatively low as it uses a maximum of three bits in a single pixel. Some of its implementations also make use of only one type of image format bringing about the problem of the inability to support a variety of image file formats.

This study attempts to contribute to area of digital image steganography and in particular the least significant bit algorithm imperceptibility against steganalysis and its data hiding capacity by proposing the use of an embedding procedure that utilizes varied and pseudo-randomly picked cover image bits to hide the secret information. The idea is to generate a group of random numbers of length equal to the secret message length. This series of random numbers are then used to identify the target pixels, color channels and bits in a cover image to embed the secret message. The hiding capacity is also improved by varying the number of bits per color channel that can be used without compromising on imperceptibility

CHAPTER THREE

3.0 METHODOLOGY AND REQUIREMENT ANALYSIS

3.1 Introduction

In this chapter, the research method used in attempting to answer the research question "What is the impact of using the image's varied and randomly selected bits in the LSB steganography method embedding process on the stego image imperceptibility and hiding capacity?" is identified and justified. The chapter also outlines an overview of the roadmap on how the proposed method was developed tested and evaluated. An in depth analysis of some existing systems is carried out to elicit the desired requirements for the proposed prototype which were established and analyzed.

3.2 Research method

The main aim of this research study is to increase the imperceptibility and the hiding capacity of the traditional LSB steganography method. In order to achieve this research aim, the researcher collected information from the literature documents, conference and research papers on the previous work related to the area of research in order to establish the state of the art. The experimental research method was used to arrive at causal inferences. This method represents the standard practice applied in manipulating independent variables in order to statistically analyze the generated data to test the research hypotheses. A notable advantage of the experimental research method is the fact that it enables other researchers to easily replicate the experiment and be able to validate the results. It is therefore considered an accurate method of research (Shuttleworth, 2008), as the researcher can effectively establish a causal relationship between variables by manipulating independent variable(s) to assess the effect upon dependent variable(s).

In this study, the researcher aimed to measure the effect of using varied and randomly selected bits during the embedding process of the LSB steganography method on imperceptibility and hiding capacity of the resultant stego image. Accordingly, an experimental design represented the best choice for this aim and objective.

According to (Hinkelmann and Kempthorne, 2008), experimental designs should clearly outline the nature of the problem under investigation, the type of the experimental design, the implementation of the experiment, the analysis of the data, and the interpretation of the results. In this study, the researcher wish was to improve both the imperceptibility and the hiding capacity of the hidden data in the traditional LSB steganography method. Accordingly, an enhanced LSB steganography method that utilizes varied and randomly determined bits during the embedding process was proposed and a set of perceptibility metrics to evaluate the quality of the stego images. The hypothesis therefore was that this different data embedding approach (independent variable) has significant effect on the hiding capacity and the imperceptibility stego image generated (dependent variable).

Being the most accurate and obvious standard for testing a hypothesis (Shuttleworth, 2008), the experimental design methodology was used in order to achieve the research objectives. The effect of dispersing the significant bits of the hidden message across a cover image during the embedding process represented the variable that was to be understood in this study. Thus an experiment was carried out to test the relationship between the specific embedding process (i.e Proposed method) and the outcome (ie Hiding capacity imperceptibility level). Essentially the output of traditional least significant bit steganography method was used to evaluate the performance and effectiveness of the proposed method's output by comparing the stego images generated by the proposed method with those generated by the traditional LSB method. This is commonly referred to as comparative experiment (Hinkelmann and Kempthorne, 2008).

The researcher developed a software prototype to demonstrate the desired functionalities of the proposed LSB data hiding method and to effectively mount the experiment to test the hypothesis and the stated objectives. As an outline for the software development section of this study, the researcher followed the generic software engineering paradigm presented by (Pressman, 2008). However, this generic paradigm is an outline for organization/company-based software systems. The application software developed in this research was not designed for any specific organization or individual. Its main purpose was to demonstrate the proposed enhanced LSB steganographic algorithm in data hiding and to test its effectiveness. Bearing this in mind, the generic paradigm was modified to suit the needs of the project. The five phases of the modified paradigm are as shown in figure 9.

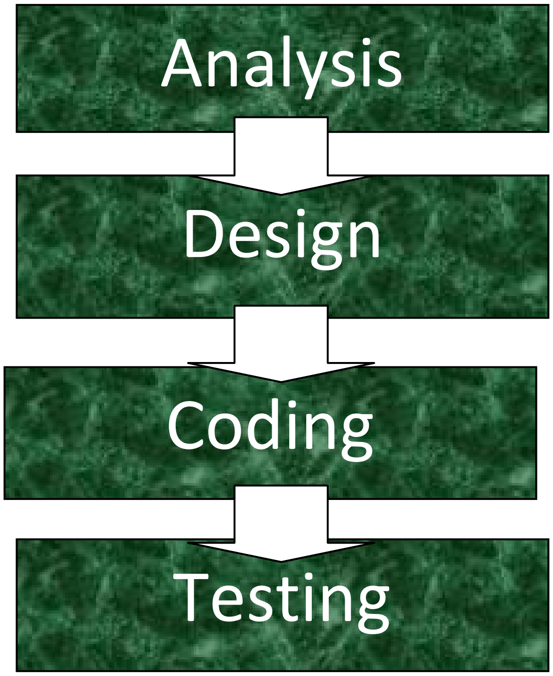


Figure 9 Software engineering paradigm

3.2.1 Analysis

Here, the required scope and functions of the software prototype to be developed were correctly outlined. The specific attributes that needed to be incorporated to the prototype were assessed.

3.2.2 Design

In the design phase, concepts and techniques were put together to form an abstract model of the proposed system.

3.2.3 Coding

In this phase, the framework conceptualized in the design phase was translated to computer code by construction of a software prototype.

3.2.4 Testing

After the construction of the prototype, testing was carried out to evaluate the effectiveness of the design.

Since the software is not a company-based software system, organizational based adaptive maintenance was not required. The software was designed for a specific use, not for a specific user. Being generic, it did not need to adapt to a specific organization or individual.

Requirement analysis was also limited. No company-related data gathering instruments, like interviews and questionnaires, was needed in this study. The requirements defined for the proposed system was based entirely on the concepts of information hiding and of other fields in computer science and information technology.

3.3 Data gathering

The prototype proposed and developed in this research is a general-purpose security data hiding application software for use in hidden communication. Accordingly, data gathering for this study was focused on the modern researches and developments in the field of information hiding, particularly in digital steganography. Therefore, no interviews or surveys on companies and company personnel were conducted during the course of the study, as they were not necessary.

The primary information resource used in was the internet and books on the subject under investigation. As it is with most subjects, the internet is evidently the best source for up-to-date and detailed information regarding digital steganography particularly being a relatively new area of research. Several research papers, technical reports, and journal articles gathered from the internet provided the technical information required for the designing and development of the proposed software prototype.

Most of the papers referenced during the course of this study were IEEE digital library, Springer Scientific Literature Digital Library and also a comprehensive archive of scientific documents sponsored by the Nippon Electric Company (NEC) Research Institute. Copies of the IBM Systems Journal downloaded from the IBM Research website were also referenced in this study.

3.4 Prototype development methodology

Due to the ever changing demands and challenges of data and information security particularly in open systems environments, data hiding systems and techniques must undergo continuous maintenance and redesign. To effectively facilitate this, an object-oriented development methodology was followed in designing and developing the prototype. Object Oriented Methodology helps to closely represent the problem domain and because of this, it is easier to produce and understand designs. It also encourages more re-use where new applications can use the existing modules, thereby reducing the development cost and cycle time.

The UML modeling language which is the industry standard for modeling object oriented systems was used.

3.5 Prototype implementation

The proposed prototype was implemented in Java programming language. The choice of language in this case was not unusual. The core value proposition of this platform according to Sun is its ability to "Write once, run anywhere". This means that the most important promise of Java technology is that you only have to write your application once - for the Java platform - and then you'll be able to run it anywhere. Fortunately, Java support is becoming ubiquitous. It is integrated, or being integrated, into practically all major operating systems. It is built into the popular web browsers, which places it on virtually every Internet-connected PC in the world. It is even being built into consumer electronic devices, such as television set-top boxes, PDAs, and cell phones. Other notable benefits of the java platform include the following;

Dynamic, extensible programs

Java is both dynamic and extensible. Its code is modularly organized in object-oriented units referred to as classes. These classes are then stored in separate files and called and loaded into the Java interpreter only when needed. This means that an application has the ability to decide as it is running what classes it needs and can load them exactly where and when it needs them. It also means that a program can dynamically extend itself by loading the classes it needs to expand its functionality.

A Java application can also dynamically extend itself by loading new classes over a computer network. This means that a java application ceases to be a monolithic block of code. Instead, it becomes an interacting collection of independent software units and components enabling a powerful new metaphor of application design and development.

Network-centric programming

From a programmer's point of view, Java makes it unbelievably easy to work with resources across a network and to create network-based applications using client/server or multitier architectures. Sun's corporate motto has always been "The network is the computer." The designers of the Java platform believed in the importance of networking and designed the Java platform to be network-centric. This means that Java programmers have a serious head start in the emerging network economy.

Security

Java is one of the first programming languages to consider security as part of its design. The Java language, compiler, interpreter, and runtime environment were each developed with security in mind. The compiler, interpreter, and Java-compatible browsers all contain several levels of security measures that are designed to reduce the risk of security compromise, loss of data and program integrity, and damage to system users.

Performance

Java programs are compiled to a portable intermediate form known as byte codes, rather than to native machine-language instructions. The Java Virtual Machine runs a Java program by interpreting these portable byte-code instructions. This architecture means that Java programs are faster than programs or scripts written in purely interpreted languages.

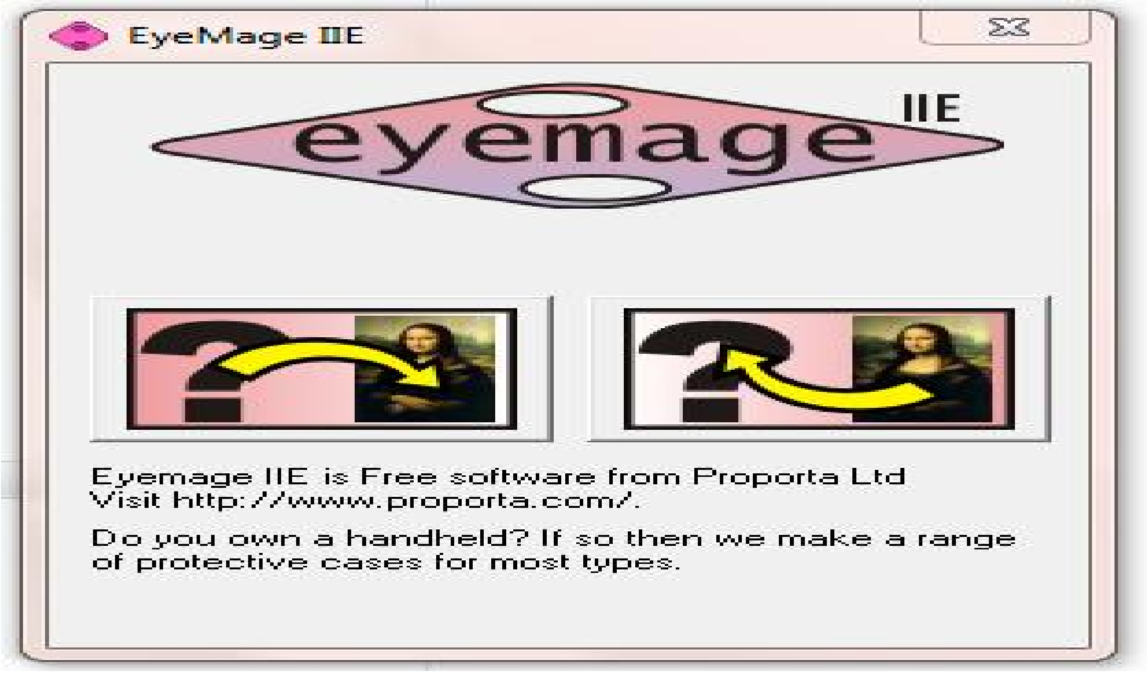
Programmer efficiency

Java has a powerful and well-designed set of APIs. Studies have consistently shown that switching to Java increases programmer efficiency. Because Java is a simple and elegant language with a well-designed, intuitive set of APIs, programmers write better code with fewer bugs than for other platforms, again reducing development time.

3.6 Analysis of the current systems

EyeMage

EyeMage is a free steganography application by Proporta Ltd. This software supports data hiding in Windows bitmaps only. It allows data to be embedded in more significant bit planes, not just on the LSBs. One of the unique features of EyeMage is its ability to create a custom noise image in which a selected data file may be embedded. However, noise images are made up of random colors and form no meaningful picture. Using noise images as cover does not qualify as valid steganography since the media is obviously cryptic and does not provide a convincing illusion. The user interface of EyeMage is quite simple and is relatively easy to use as shown by its screenshot in figure 10.



Eyemage IIE is Free software from Proporta Ltd
Visit http://www.proporta.com/.

Do you own a handheld? If so then we make a range of protective cases for most types.

Figure 10 Screen shot of EyeMageIIE

Invisible secrets 4

Developed by NeoByte Solutions (NeoByte Solutions, 2002), Invisible Secrets is a commercial steganography software that supports a variety of image formats. Its latest version supports at least four different image formats ie

i. Windows Bitmap (.bmp)

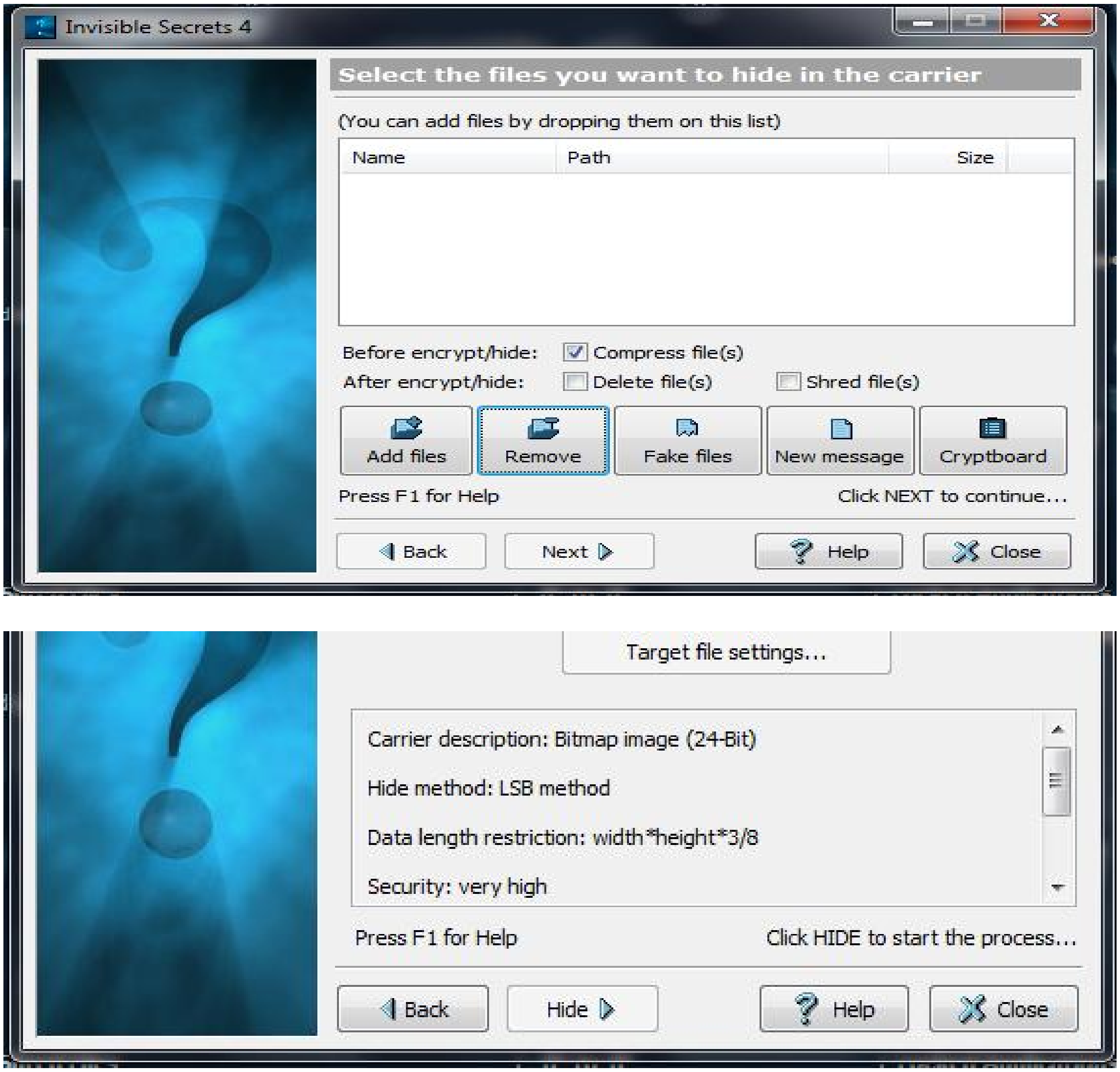
ii. JPEG File Interchange Format (.jpg)

iii. Portable Network Graphics (.png)

iv. Wave sound files (.wav)

Invisible Secrets uses standard traditional LSB encoding method for Windows bitmaps. For JPEG and PNG images, however, a technique referred to as *comment insertion* is used. In this technique, the hidden data file is embedded as comment in the header of the image. Although this approach safely does not make any changes to the image itself, such a technique is highly insecure. Since the data bits are not encoded as integral parts of the cover image, the hidden data is readily extractable by potential attackers. Moreover, inserting large files would greatly increase the size of the stego image and may attract suspicion since JPEG and PNG images often have very small file sizes.

Invisible Secrets employs a wizard-style user interface that makes it relatively user friendly as shown in its screenshot in figure 11.



Select the files you want to hide in the carrier

(You can add files by dropping them on this list)

Before encrypt/hide: ☑ Compress file(s)

After encrypt/hide: □ Delete file(s) □ Shred file(s)

Press F1 for Help

Click NEXT to continue...

Carrier description: Bitmap image (24-Bit)

Hide method: LSB method

Data length restriction: width*height*3/8

Security: very high

Press F1 for Help

Click HIDE to start the process...

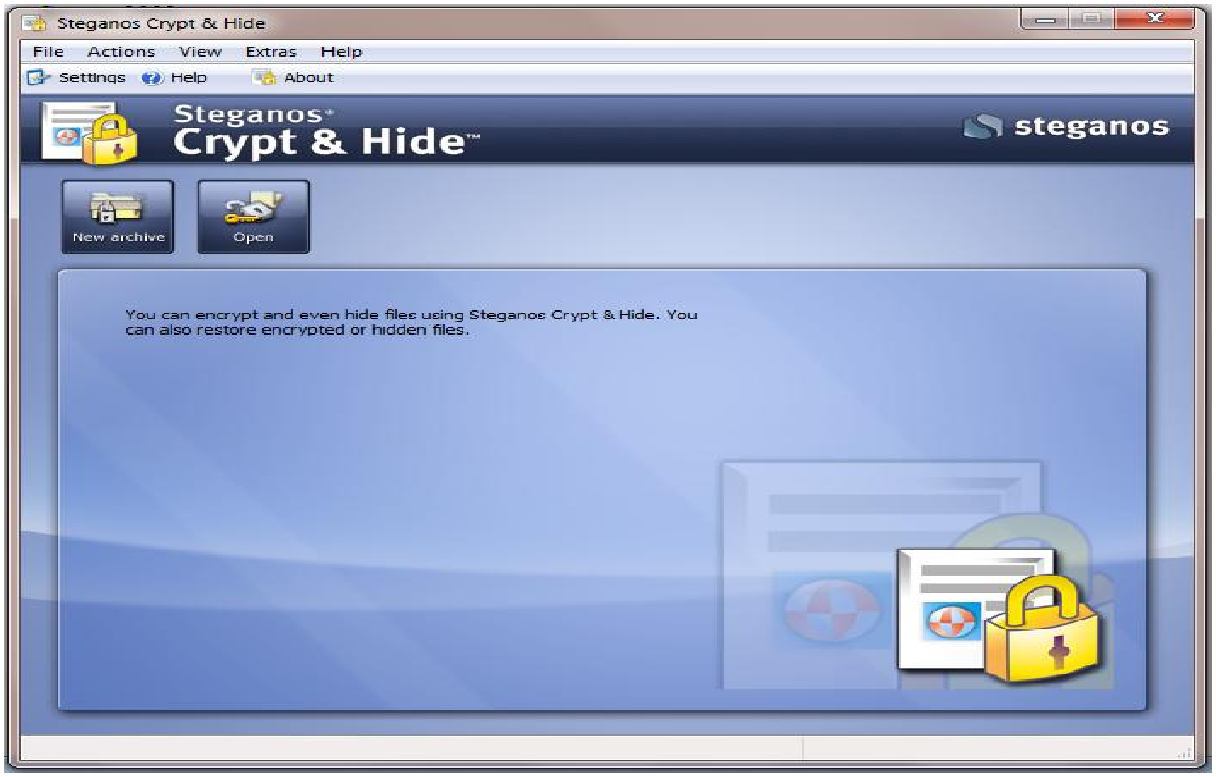
Figure 11 Screen shot for Invisible secrets

It is the observation of this study that such interface designs are very effective in enhancing the usability of steganographic software.

Invisible Secrets is designed and tested to run under the Windows 95, Windows 98, Windows 2000, Windows ME, Windows NT, and Windows XP operating systems. However NeoByte Solutions website does not provide any information regarding the minimum system requirements of Invisible Secrets 4.

Steganos crypt and hide

This is a steganographic software program distributed as a part of the Steganos Privacy Suite, which includes several security features such as file shredding, file encryption, disk encryption,and Internet tracks erasing (Steganos GmbH, 2006)). Although recognized as one of the most popular steganography software on the Internet, Steganos uses the LSB embedding approach and supports embedding on standard Windows bitmaps only. A screen shot of the steganos crypt and hide is shown in figure 12.



You can encrypt and even hide files using Steganos Crypt & Hide. You can also restore encrypted or hidden files.

Figure 12 Screen shot for Steganos crypt and hide

The user interface of the Steganos File Manager is quite similar that of the WinZip file compression software and is not as easy to use as other steganographic software.

The following are the minimum system requirements of Steganos as listed on its Help file.

i. Pentium processor

ii. 64 MB RAM

iii. 9 MB hard drive space

iv. Screen resolution of 640x480 pixels at 256 colors.

v. Mouse or other pointing devices (since it does not provide full keyboard support)

The third eye

This is a freeware steganographic application software program developed by Satya Kiran (Kiran, 2007). Its latest version supports steganography in three image formats:

i. • Windows Bitmap (.bmp)

ii. • Graphics Interchange Format (.gif)

iii. • PC Paintbrush Format (.pcx)

The Third Eye also uses standard LSB encoding like Steganos and Invisible Secrets. The user's manual defines the following minimum system requirements:

i. Intel 80386 processor

ii. Windows operating system

iii. 1.5 MB free hard disk space

The application features a Multiple Document Interface (MDI) that is minimalistic and quite easy to use since very few controls and menu options are on the main window. Figure 13 shows its user interface screenshot.

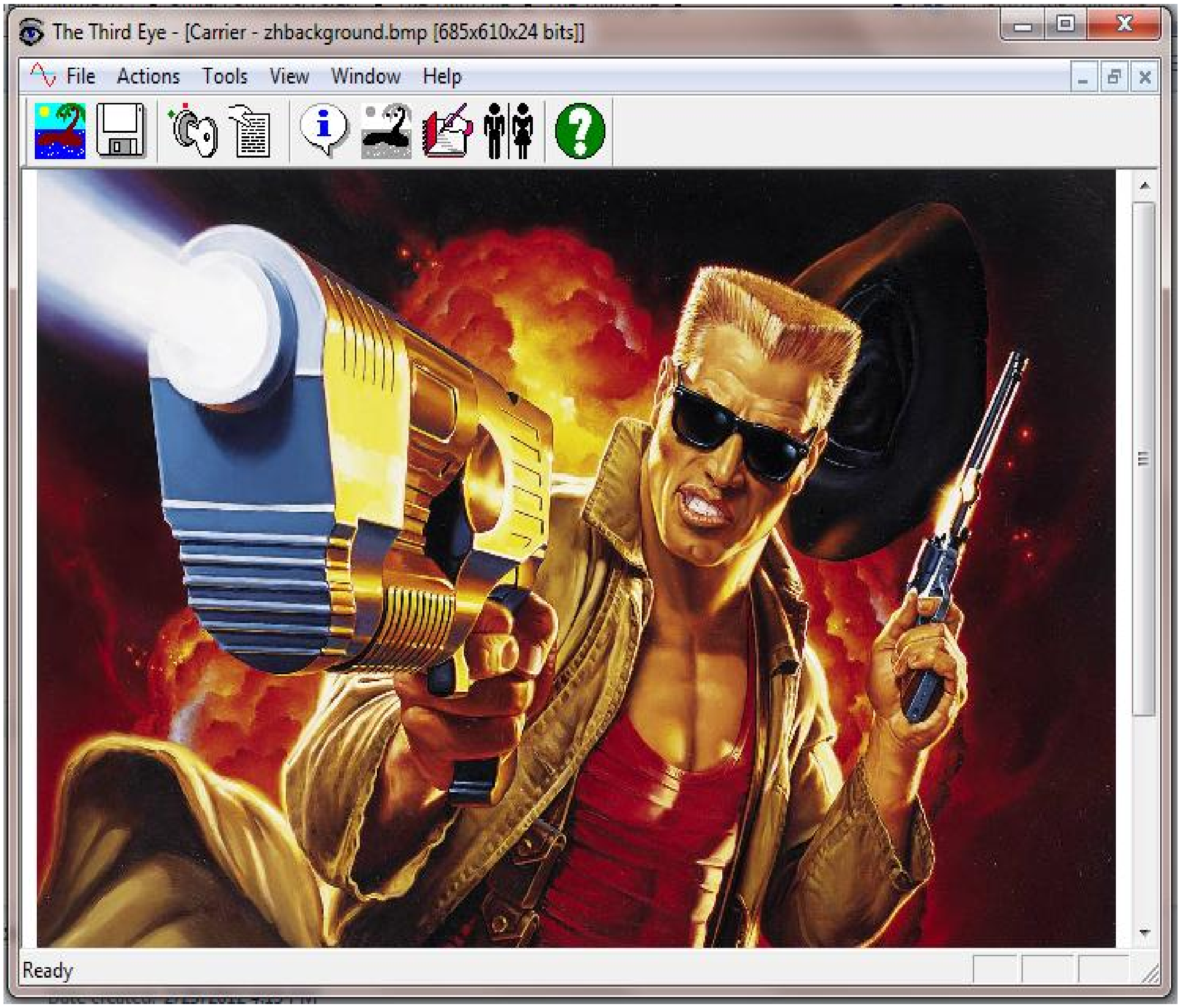
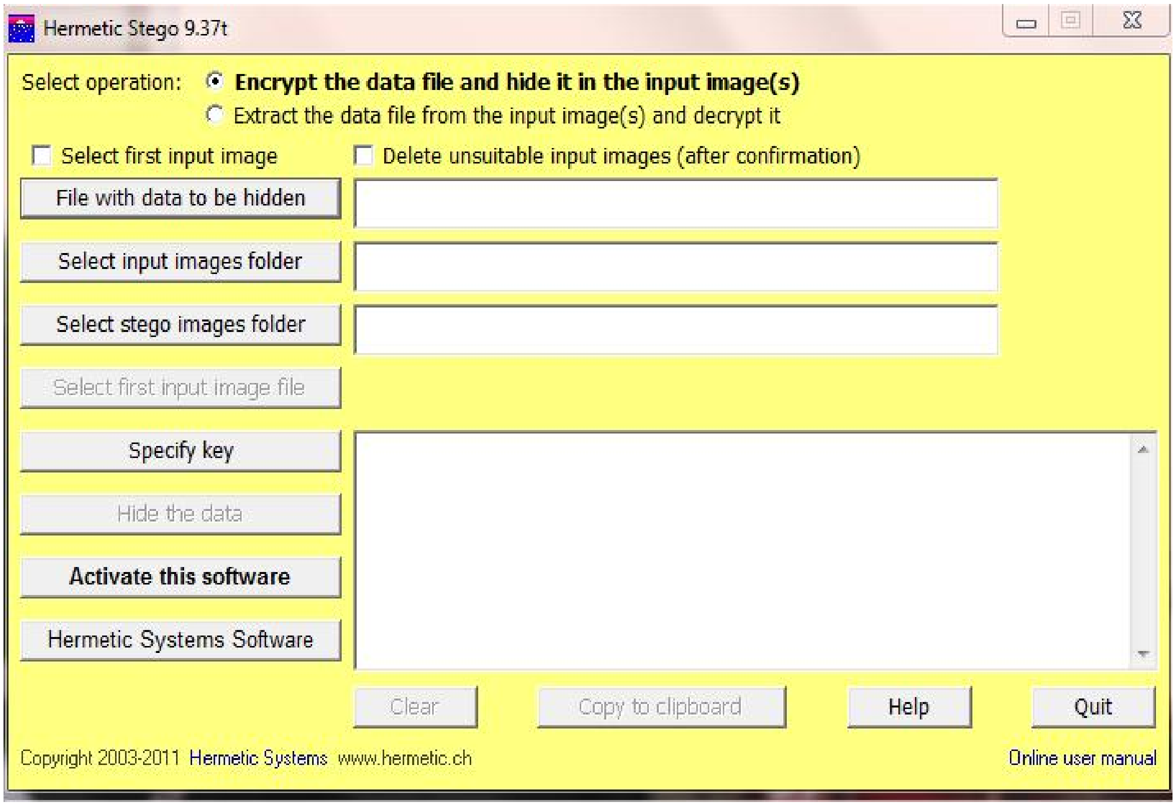

The Third Eye - [Carrier - zhbackground.bmp [685x610x24 bits]]
File
Actions
Tools
View
Window
Help
Ready

Figure 13 Screen shot of the ThirdEye

Hermetic stego

Hermetic Stego is a program for hiding a data file in a single BMP image or in a set of BMP images and for extracting a data file hidden in this way. The file may be of any kind, not just a text file, and thus may be a file such as an MS Word document, an Excel spreadsheet or an image file. Furthermore the file may be of any size up to 200 MB, as long as the BMP image files are sufficient in number and size to hold the data.

This steganography software follows an LSB pixel swap method proposed by Lee *et al.* (2009). In this method a pseudo- random sequence is used to randomly pick one of two LSB neighbors to swap with the secret data bit. If the two pixels lie within a specified distance α (α=2 or 3 generally), they are suitable for embedding, otherwise another set of pixels is generated. The screen shot of Hermetic stego is as shown in figure 14.



Select operation: ● **Encrypt the data file and hide it in the input image(s)**
○ Extract the data file from the input image(s) and decrypt it

□ Select first input image □ Delete unsuitable input images (after confirmation)

Hermetic Stego 9.37t
File with data to be hidden
Select input images folder
Select stego images folder
Select first input image file
Specify key
Hide the data
Activate this software
Hermetic Systems Software
Clear
Copy to clipboard
Help
Quit
Copyright 2003-2011 Hermetic Systems www.hermetic.ch
Online user manual

Figure 14 Screen shot of Hermatic stego

3.7 Analysis of Perceptibility metrics for the traditional LSB and other methods

Data analysis of perceptibility metrics in the traditional LSB method and other existing current applications for the traditional LSB method was carried out using standard images and a common secret message (a 51 kilo byte word document). This data is tabulated as shown in table 2 and 3 respectively.

Steganographic Application	**A AD (db)**	**M S E (db)**	**SNR (db)**	**PSNR (db)**	**RS (db)**	**SPA (db)**
EyeMage	0.1155	0.3599	53.2	62.0	11.4644	10.9577
Invisible Secrets	1.5006	3.0025	44.0	52.8	103.8117	96.4961
The Third Eye	0.5727	0.8978	49.2	58.1	33.1363	33.9031
Hermetic Stego	1.8592	4.3366	42.4	51.2	86.59	77.02

Table 2 Anaysis of perceptibility metrics for the various LSB method applications

IMAGE	A AD (db)	M S E (db)	SNR (db)	PSNR (db)	RS (db)	SPA (db)
Banana.jpg	0.1341	0.268	57.9	63.4	19.47	19.54
Dancers.jpg	0.1514	0.304	53.3	62.8	16.22	17.29
Graduation.jpg	0.6695	1.34	51.3	56.4	49.74	51.38
Office.jpg	0.151	0.302	58.4	62.9	17.46	18.97
zhbackground.bmp	0.5011	1.003	48.8	57.6	39.33	39.51

Table 3 Perceptibility metrics for the traditional LSB method

3.8 Requirements analysis specifications for the proposed system

This research focused on developing a steganographic application that is suitable and easy to use for data hiding. In view of the application's intended function, certain points were considered in designing the software.

3.8.1 Functional requirements

Functional requirements (FRs) were used to capture the desired behavior of the system, in terms of the services or tasks the system was required to perform (Sousa, 2004). They outlined the system's product features and described the work that is to be done. These requirements directly support the user requirements by describing the "processing" of the information or materials as inputs or outputs. They were therefore used to capture the required behavior of a system in terms of functionality.

The developed enhanced least significant bit steganographic system is therefore able to do the following:

i. Embed data/files in any image format using the traditional LSB Algorithm
ii. Embed data/files in any image format using the enhanced random LSB Algorithm
iii. Decode stored data/file from an image using the traditional LSB Algorithm
iv. Decode stored data/file from an image using the enhanced random LSB Algorithm
v. Provide for browsing the location of the data/file to be embedded in a cover image
vi. Provide for browsing the location of the desired cover image
vii. Provide for choice of using a hushed password value as key
viii. Use a random image (a custom noise image) as cover image
ix. Provide for choice of maximum number of bits to use per color channel in a cover image
x. Provide for feedback message whether the data/file has been successfully embedded/extracted to/from the cover image

3.8.2 Non functional requirements

Non functional requirements (NFRs) are generally requirements used to impose restrictions related to use attributes on the product being developed (Sousa, 2004). These were used to capture the desired overall user experience (user attributes). They are descriptive of the desired parameters for the system performance, quality, attributes, reliability and security.

The proposed enhanced least significant bit steganographic system was therefore to satisfy the following non functional requirement properties:

Usability

i. The user interface should be minimalistic, intuitive and easy to learn.

ii. The user interface should be fairly easy-to-use with the mouse.

iii. Informative error messages should accompany the entire steganographic process

Security

i. The system should provide for password whose hushed value should be used for protection of the embedded data/file.

ii. The enhanced steganographic algorithm should follow a scattered or randomized pattern for embedding.

Reliability

i. The system should always decode the exact data/file that was encoded in the cover image in a user specified folder and display the file name.

3.9 Testing and evaluation

The primary goal for testing was to find out whether the proposed enhanced LSB method achieves the objective of improved imperceptibility and hiding capacity compared to the traditional LSB algorithm. In order to establish this, two types of tests were carried out in stego images produced by the traditional LSB algorithm and compared to those produced by the proposed enhanced algorithm. These tests are:

i. Subjective tests (for visual attacks)

ii. Objective tests (for statistical attacks)

3.9.1 Subjective tests

These tests depend on the visual analysis of the stego image ie the process of detecting hidden messages in stego files through inspection by naked eye. Visual attacks represent one of the easiest steganalysis methods (Wang and Wang, 2004). If the human eye can pick distortions by comparing the original image and the stego image, then steganography is already compromised as this can be a sign that there is hidden message within the image file (Provos and Honeyman, 2003)

Subjective testing and analysis by viewers is still a method commonly used in measuring and determining the quality of the image**.** This evaluation emphatically examines both the fidelity and the intelligibility of the image. When taking a subjective test, a viewer focuses on the difference between the stego image and the original image, he or she should point out such details where information loss cannot be accepted. This type of evaluation test is based on a survey, usually of 10 - 15 persons who have examined the image before and after the embedding of the hidden data. They are then asked if they had found any alterations or distortions in the perceptual vision of the image. The human vision test is the first type of evaluation test that has been found to measure the quality of steganographic objects after embedding the hidden data (Ghrare *et al*., 2008).

3.9.2 Objective testing

This refers to the automated statistical analysis that examines the statistical properties of the stego images produced by the proposed method and the traditional LSB method.

Statistical attacks are more powerful than visual attacks as they are able to reveal the most tiny modifications in the statistical properties of a image (Artz, 2001).

The following image quality metrics were employed:

Perceptibility metrics

After embedding secret data in a carrier image, cumulative image distortion brought about by the secret data bits results in a change in image quality metrics making it easier to guess the existence of hidden message. Generally, the smaller the changes introduced in image quality metrics, the better the embedding method and the lesser the distortion introduced in the image. So there is a relation between image's distortion and image quality.

Peak signal to noise ratio (PSNR) and mean square error (MSE)

Both of these metrics are the most common and widely used full reference metrics for objective image quality evaluation. In particular, PSNR is used in many image processing applications and considered as a reference model to evaluate the efficiency of other objective image quality evaluation methods (Wang *et al*., 2002b). PSNR as a metric computes the peak signal-to-noise ratio, in decibels (**db**), between two images. It is used in steganography to measure the peak signal-to-noise ratio in the original image and the stego image after embedding the hidden data. In the literature, PSNR has shown the best advantage almost over all other objective image quality metrics under different image distortion environments and strict testing conditions (Wang *et al*., 2002a).

On the other hand, MSE measures the statistical difference in the pixel values between the original and the reconstructed image (Stoica *et al*., 2003; Wang *et al*., 2003). The mean square error represents the cumulative squared error between the original image and the stego-image.

A lower MSE value means a better image quality ie lesser distortion in the cover image while the higher the PSNR value the better the degree of hidden message imperceptibility (Mei *et al.,*2009).

Signal to noise ratio (SNR)

This metric estimates the quality of the stego image compared with the original image. It is used to quantify how much a signal has been corrupted by noise. It compares the level of a desired signal to that of the background noise. The higher the SNR ratio between the original image and the stego image the lesser obtrusive the background noise. Signal to Noise Ratio compares the level of noise between the cover image and the stego image and shows the difference between them in order to indicate the quality of the stego image after the process.

SNR is measured in **db**. The larger the value of SNR the higher the imperceptibility level of the embedding algorithm (Mei *et al.,*2009).

The average absolute difference (AAD)

AAD is an objective metric used to quantify the difference between original image and the stego image after the process of data hiding ie to determine the mean difference between the two images**.**

Correlation quality (CQ)

This is used to measure the overall quality of an image. Its value generally depends on the image and the quality of the digital camera used to take the image. Lower values of correlation quality metric indicates poor quality image. (Mei *et al.,* 2009).

Normalized cross-correlation (NCC)

NCC establishes a ratio of quality between two digital images. If the images are of the same quality, the NCC value should be 1. (Mei *et al.,* 2009). For a good embedding process, the NCC between the two images (cover and stego) must tend to 1.

Robustness metrics

Steganalysis tools can be used to measure the robustness of an embedding algorithm i.e to determine the existence of embedded data in a carrier file. Steganalysis applications are used to detect the very presence of hidden information in stego files (Lee, 2006). According to (Fridrich *et al.,* 2002), among the most recognized approaches to practical steganalysis are the first order statistics (histogram analysis) and the higher-order statistics (RS Steganalysis and Sample pair analysis).

First – order statistics

Histogram analysis

An image histogram analysis is used to help detect significant changes in frequency of appearance of the colors between the original cover image and the resultant stego image. More pronounced changes in the frequency of colors reveals greater distortion and shows poor embedding.

High – order statistics

i) Reed-solomon (RS) analysis

This is a method proposed by Fridrich *et al*. (2001) for detecting the use of LSB steganography. RS measures the smoothness of the changes among pixels of an image (the lower the value, the smoother the changes among them, or the lesser the noise of the image). RS is one of the most reliable quantitative steganalysis methods (Fridrich *et al.,* 2001).

ii) Sample pair analysis (SPA)

Dumitrescu *et al*. (2003) proposed SPA, a method to detect LSB steganography via sample pair analysis. It is based on probabilities of transitions between sample pairs due to LSB embedding operations.

RS and SPA are the most reliable detectors of thinly-spread LSB steganography (Andrew, 2004).

Hiding capacity

In the traditional LSB method, each image pixel least significant bit is changed because the change is not perceptible to the human eyes. Therefore each pixel has the same payload which is equal to 1 bit. The term payload is used to indicate the maximum number of bits that can be hidden with an acceptable resultant stego-image quality.

Payload (P) is calculated as shown in equation (3) below.

$$P = t / N, 0 \leq T \leq 1 \qquad \textbf{(3)}$$

Where:

t = the length of the desired secret message

N = the total number of the pixels in the cover-image.

When more than one bit is inserted inside each pixel, the total number of t could be higher than 1 bit subsequently making P or payload higher.

3.9.3 Conclusion

The perceptibility metrics of the traditional LSB method revealed that the statistical characteristics of the carrier files are significantly distorted compared to those of the original files. The various implementations of the traditional LSB method tested showed the same conclusion with some indicating slight improvements. According to Venkatraman *et al*. (2004), a good embedding algorithm should as much as possible endeavour not to adversely distort the statistical attributes of the cover medium to avoid obvious perception of the hidden data.

The proposed method in this study seeks to address not only the issue of the quality of the stego images in pursuant of increased perceptibility of the secret data but also to significantly increase the payload capacity of the carrier file by using the first three least significant bits of each color channel. The experimental research method that was adopted for this study helps in positions the research findings to the scrutiny of other researchers who can easily replicate the experiment and to validate the results (Shuttleworth, 2008).

CHAPTER FOUR

4.0 DESIGN

4.1 Introduction

In this chapter, concepts and techniques of software design are put together to form a clear and coherent abstract model of the proposed system. Various design techniques and tools are employed in order to make sure that errors are minimized during the translation of the design into the subsequent deliverables.

4.2 System design specification

The proposed system model framework is based on the generic model presented by Birgit Pfitzmann (Pfitzmann, 1996) whose block diagram is depicted in figure 15.

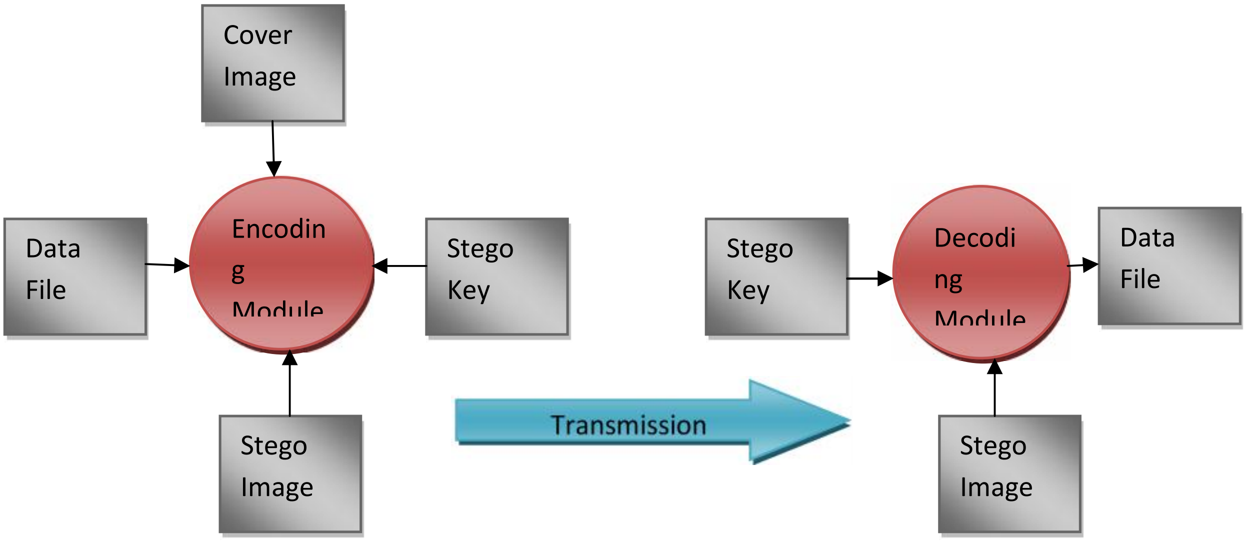


Figure 15 Framework of the proposed system

According to Pfitzmann (1996), a complete steganographic transaction process is made up of three distinct processes:

i. Embedding
ii. Transmission
iii. Decoding

4.2.1 Embedding

This involves all the tasks performed in embedding hidden data in a cover image. The module for these tasks makes use of three inputs:

i. Data file
ii. Cover image
iii. Password message digest

The module then produces a single file (the stego file/image) as the output of the entire process. Unlike other steganographic systems discussed earlier, the proposed system utilizes a more complex variation of LSB encoding by incorporating a password whose hash function is used as the seed for the generation of the pseudo random numbers which are used to facilitate a walk path through the image to pick specific bits for embedding the secret data.

4.2.3 Generation of pseudo random numbers

A pseudo random number generator (PRNG) is a number generator that produces numbers that have the appearance of randomness, but nevertheless exhibiting a specific repeatable pattern. Numbers generated this way are useful in many different kinds of applications as discussed below.

Simulation

A computer can be used to simulate natural phenomena. This is achievable by making use of random numbers. Simulation is used in many fields such as in operations research (for example in an airport where people come at random intervals). (Law and Kelton, 2000)

Sampling

In instances where large and voluminous data is under consideration, it is often impractical to examine all possible cases. Random samples however can be used to provide insight into what constitutes typical behavior.

Numerical analysis

Random numbers comes in very handy in designing ingenious techniques for analysis and solving seemingly complicated numerical and mathematical problems.

Computer programming

Computer algorithms can be used in designs and implementation of programming logic for solutions in various areas. Random values make a good source of data for testing the effectiveness of such computer algorithms. (Knuth, 1997)

Desirable attributes for random numbers

According to Knuth (1997), the following are the desirable attributes of good random numbers.

i. The numbers should be uniformly distributed

ii. The numbers should be statically independent

iii. Though the stream of the random numbers will eventually repeat depending on parameters used to generate them, the stream length should be sufficiently larger than the desired length for a particular application.

iv. The generation of random numbers should be fast.

4.3 The Mid square method

This method was proposed by Newmann and Metropolis (Knuth, 1997). In this method, an initial seed is selected, then squared. The middle four digits of the squared value are then taken as the first random number in the series. The random number generated most recently is again squared and the middle most four digits of this squared value taken as the next random number. This is continuously repeated until the required number of random numbers is generated. This is demonstrated in table 4 assuming an initial seed value of 8765.

SN	n (FOUR DIGITS)	n^2
1	8765	76825225
2	8252	68095504
3	0955	00912025
4	9120	83174400
5	1744	03041536
6	0415	00172225
7	1722	02965284

Table 4 Sample random numbers using mid square method

The algorithm for Mid Square is as summarized using the algorithm shown below.

Algorithm

```
Input n (four digits)
N = No of random numbers required
For I = 1 to n do
 {
    n_square = n^2
    n_string = n_square(convert n_square into n_string)
    count the number of characters (n1) in n_string
      if n1<8 then
         {
            add (8-n1) zeros to left of n_string
            }
 X= middle four characters on n_ string
 n=X (Convert string into integer)
 Print I, n
}
Stop
```

4.3.1 Limitations of mid square method

The mid square pseudo random number generator has various short comings which include the following.

i. It is relatively slow.

ii. Statistically unsatisfactory.

iii. Sample of random numbers may be too short.

iv. There is no relationship between the initial seed and the length of the sequence of random numbers.

4.4 The linear congruential generator (LCG)

The Linear Congruential Generator (LCG) method proposed by D.H. Lehmer is one of the most successful random number generators particularly with computer memory. Because its formula is simple and straight forward, LCG can also reproduce a sequence of repeatable random numbers with any computer programming language making its implementation easy. This method was therefore used in this research in generating the pseudo random numbers used to match the specific bits in the cover image where the secret data bits are hid.

According to Hull and Dobell (1972), a linear congruential sequence defined by *m, a, c* and X_0 has full period if and only if the following three conditions hold:

i. The only positive integer that exactly divides *m* and *c* is 1

ii. If *q* is a prime number that divides *m*, then *q* divides a -1

iii. If 4 divides *m*, then 4 divides *a* -1

Additionally:

The value of *m* should be rather large since the period cannot have more than *m* elements. The value of *m* should also necessitate a fast computation of (aXn + c) i.e speed the generation of random numbers. Observing all these requirements, the parameters for the LCG used in this research were as follows

Modulus (*m*)

The 48- bit computer word length was been picked as the value of *m.* A Pentium IV computer and above should have this word length. This in essence provides the size of *m* to be 2^{48} which is equivalent to **281,474,976,710,656.** For this experiment and bearing in mind that the digital Images being used are a few kilobytes in size, this period is considered sufficient enough for setting up the experiment. To ensure faster generation, *m* is recommended to be a power of 2 or close to a power of 2 and hence the choice of the word length. The AND operation also enhances speed instead of the normal division operation which is considered slower.

The seed (X_0)

The first value or the seed (X_0) is supplied by the message digest of the user supplied password. This is done using a special form of Encryption that uses a one-way algorithm which when provided with a variable length unique input (message) will always provide a unique fixed length output called hash, or message digest. This is both to enhance the security of the system and also to make sure that the same hush function is used during the extraction of the message to facilitate generation of the same random numbers used during embedding. The password hush value is therefore used as the seed and not the password itself. So even if the password is leaked it would still not help in the extraction process.

a* (Multiplier) and *c (Increment)

To ensure full period and in following with the requirements identified above, the values of the multiplier and the increment were picked as follows:

a (Multiplier) = 25214903917

c (Increment) = 11

These have been tested and have been confirmed to fulfill the identified requirements to obtain a full period. The values were used to initialize the random number generator used for the prototype.

4.4.1 Reasons for choosing the LCG method

LCG method was chosen for the following reasons:

i. It is fast and requires minimal memory (typically 32 or 48 bits) to retain state. This makes it valuable for simulating multiple independent streams.

ii. LCG is widely used in simulations and Monte Carlo calculations. Because it is very fast, and because it has minimal state space, it remains attractive for use in parallel computing environments.

iii. A problem with LCGs is that the lower –order bits of the general sequence have a far shorter period than the sequence as a whole if *m* is set to a power of 2.

iv. Nevertheless, LCGs is a good option for instance, in an embedded system, the amount of memory available is often very severely limited. Similarly, in an environment such as a video game console taking a small number of high –order bits of an LCG may well suffice. (Deng and Lin, 2000)

The system utilizes a 128-bit Message Digest 5 (MD5) hash-value of the user-given stego key (password) to seed the LCG.

The numbers generated by this PRNG determines the specific bits in the pixel bytes of the cover image where data bits of the secret data file are to be embedded. This helps in randomizing the otherwise deterministic and sequential pattern of embedding in the traditional LSB algorithm thereby protecting the hidden data from unauthorized extraction. Figure 16 illustrates the Hierarchical Input-Process-Output (HIPO) chart for the embedding process while the comprehensive embedding algorithm is outlined in figure 17.

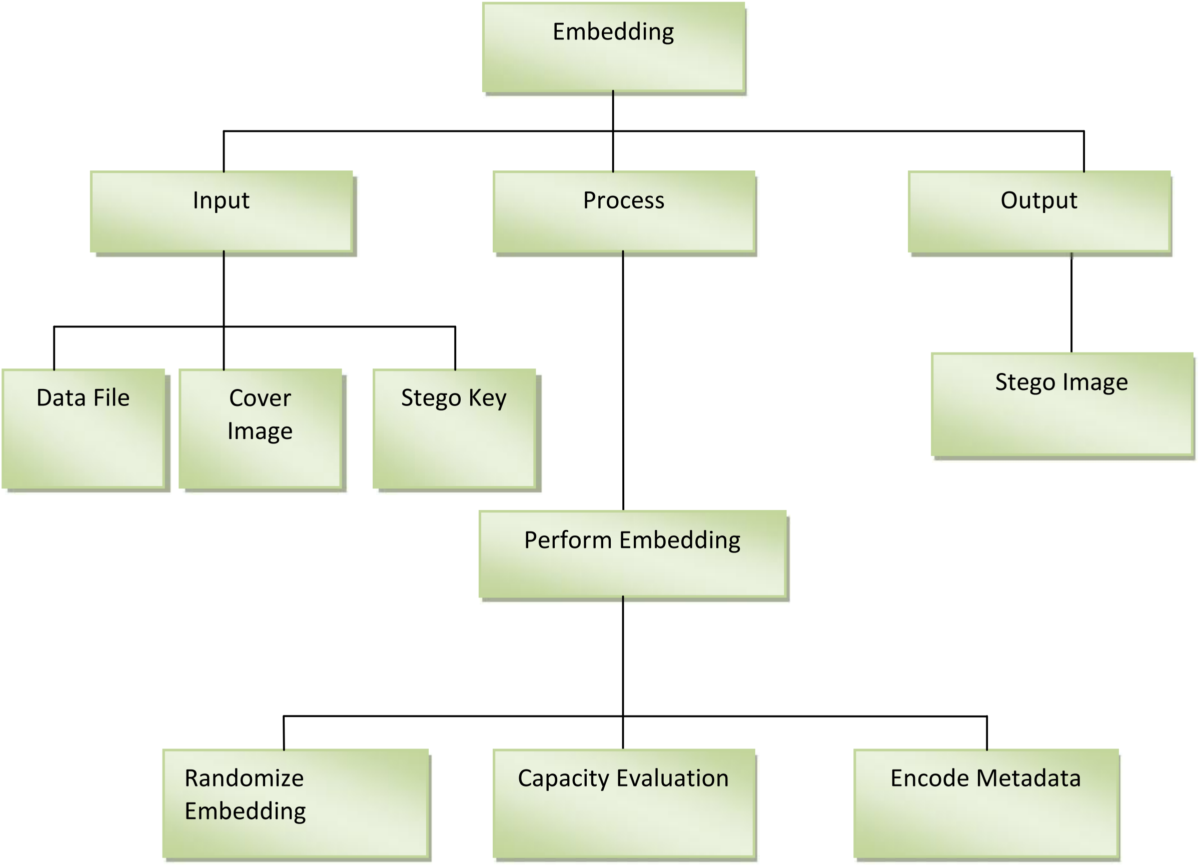


Figure 16 Hierarchical Input-process-output chart for embedding process

```
Input     : Cover Image, Secret file (Payload)
Output    : Stego image (image containing hidden file)
1. Convert secret file to a byte stream
2. Get secret file length
3. Get image width
4. Get Image Height
5. Construct buffered Image
6. Convert buffered image to byte array
7. Number of pixels = Image Width * Image Height
9. Initialize random number generator with the password message digest
8. Initialize bit to use per color channel
9. Get secret file header size { File name + File length}
10. If file size > bits available for writing based on bits per color channel
11. Increment bits per color channel
12. If file size > maximum bits per color channel
13. Display error message {insufficient image size} else
14. Initialize image byte array
14.  Write file header to image
15. Update channel bits used
16. Use LCG to  Select a random pixel,  Select a random pixel color channel,   Select a random color channel bit
17. Let bitToWrite [x][y][channel][bit] denote the selected bit in a specific color channel for writing
18. Let $m_i$ denote the message bit embedded in a color channel bit,  bitToWrite[x][y][channel][bit]
19. For all image color channels do the following
20. If LSB (bitToWrite[x][y][channel][bit]) = $m_i$ then
21. do nothing
22. If LSB(bitToWrite[x][y][channel][bit])  not equal to  $m_i$ then
23. bitToWrite[x][y][channel][bit] = $m_i$
24. while secret file length; Repeat step 16 to 23 to embed the entire message
25. Close stream.
```

Figure 17 The proposed enhanced LSB method embedding algorithm

4.5 Transmission

This is the process of the transfer of the stego image from a sender to the targeted recipient. This could be through Internet, a local area network (LAN), a disk, or any other digital media. Using the Internet, the stego image could be sent as an email attachment or even posted on a web page. Unlike emails, web pages have no specified recipients making them an excellent media for steganographic transactions as it is virtually impossible to track down the intended recipient of the hidden message.

The proposed software prototype does not cover this part of the steganographic transaction as using the same steganography software for file transfer is quite impractical. Common or traditional tools for data transfer such as popular web browsers must be used to enhance that innocuous effect.

4.6 Extraction

This is the process of decoding the hidden data from the stego image. This is done at the receiving end by the recipient. The decoding module uses the stego key used during the input to perform a reverse operation of the embedding process. If the stego key entered is correct, the original data file will be properly extracted from the cover image. The original cover image however cannot be regenerated out from the stego image. The process therefore requires two inputs:

i. Stego Image

ii. Password message digest

Figure 4.2.3.1 illustrates the Hierarchical Input-Process-Output (HIPO) chart for the decoding process while the comprehensive extraction algorithm is outlined in figure 18.

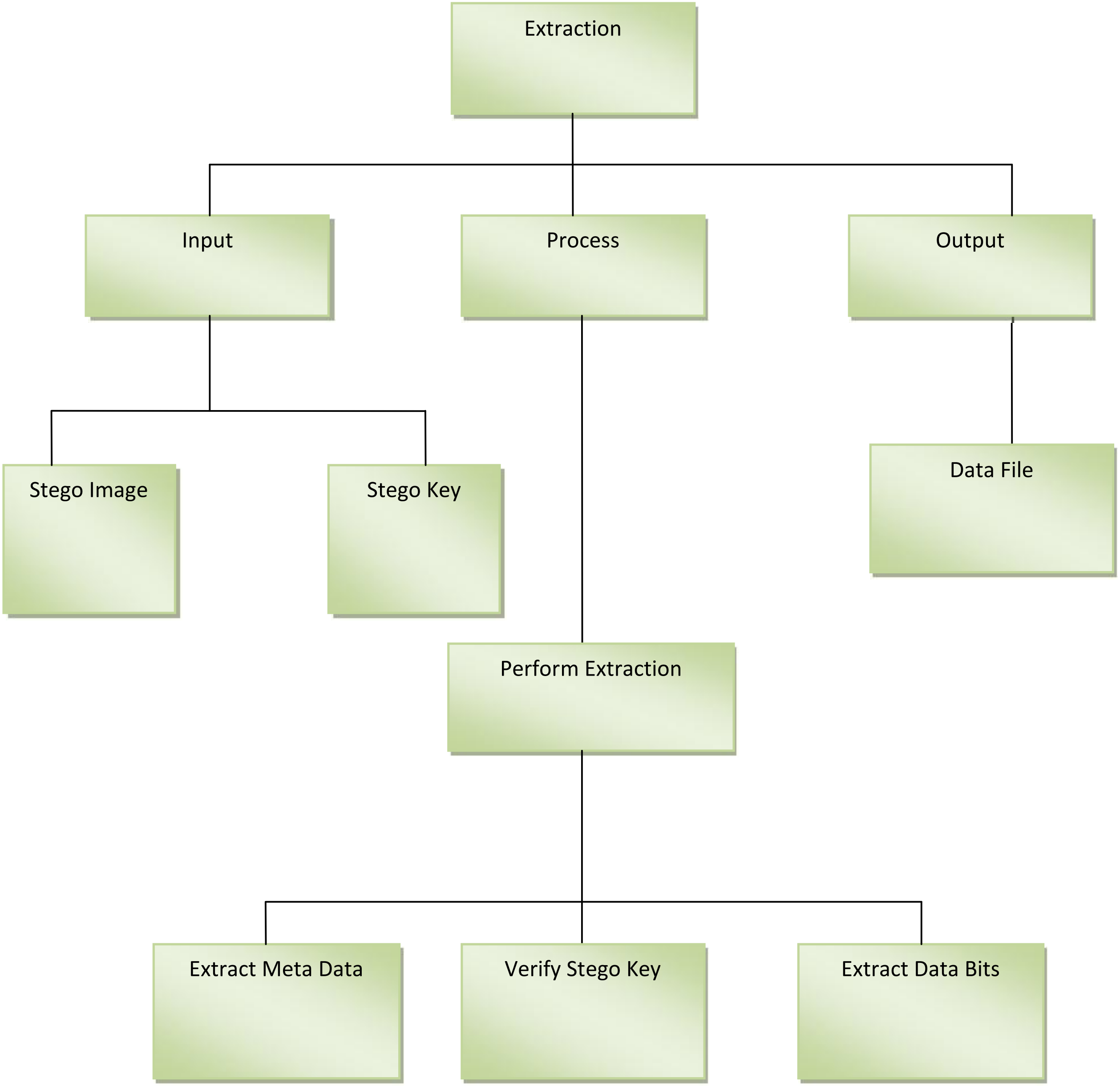


Figure 18 Hierarchical Input-process-output chart for the decoding process

```
Input    : Stego Image, Password message digest
Output   : Secret file

1. Get image width
2. Get Image Height
3. Construct buffered Image
4. Convert buffered image to byte array
5. Initialize random number generator with the password message digest
8. Initialize image bits read to null
9. Read secret file header
10. Update channel bits read
11. Use LCG to
        Select a random pixel
          Select a random pixel color channel
           Select a random color channel bit
12. Let bitToRead ([x][y][channel][bit])denote the selected bit in a specific color channel for reading
13. Let $m_i$ denote the message bit read in a color channel,  bit bitRead ([x][y][channel][bit])
14. For all image color channels do the following
15. If LSB(bitToRead([x][y][channel][bit])=  $m_i$ then
16. do nothing
17. If LSB (bitToRead([x][y][channel][bit]) not equal to  $m_i$ then
18. bitToRead ([x][y][channel][bit])= $m_i$
19. Pack bit in bitSet
21. While secret file length; Repeat step 11 to 20 to read the entire file
22. Close stream.
23. Obtain the entire message stream and convert it back into ASCII format
```

Figure 19 The proposed enhanced LSB algorithm general extracting algorithm

During the decoding process, the metadata is extracted in order to determine the data file size and verify the stego key entered by the recipient. If the stego key entered by the recipient is verified to be correct, the data/file is properly decoded and extracted as final output. However, the original cover image cannot be regenerated out of the stego image as the data/file still remains embedded.

System Description

Pseudocode for the Stegosystem

The following are the pseudo codes for the embedding and extraction processes for the proposed enhanced LSB method. They are used to depict only the essential logic and fundamental generalized structure of the processes.

Embedding function

```
Start
initialize image coordinates (x and y) to zero
initialize number of bits used per color channel to 1
initialize current bit number to be read to zero
initialize array for bits in the image written to null
initialize LCG to null
read message  (Secret data file)
get image
if image = null
display error message
else
```

```
get secret data file length
get image width
get image length
get image type
if image is true color image
construct buffered image
for x=0;x<image width ;x++
 for y=0;y<image height; y++
get initial image Pixel RGB value
number of bits used per color channel =  1
get the secret data file name
initialize random number generator using the password message digest
channel Bits = 1
header size = 0
noOfPixels = imagewidth*imageheight
get file dataHeader
get file header size
if file size{header size+datalength} > selected bits per colo channel
increment color channel bits
if file size>maximum color channel bits
display error message
else
```

```
set Bits used per color channel = 1
initialize hit check array {setting the array values false]
write headerData
reset channels bits used per color channel to channel Bits
for each bit in the secret message byte
select a random pixel
  select a random pixel color channel
   select a random color channel bit
Change bit to secret message bit value
loop
Write message in cover image
close stream
get image with embedded data
end.
```

Extraction function

```
Start
initialize number of bits used per color channel to 1
initialize array for bits in the image read to null
initialize random number generator to null
get image
if image = null
```

```
display error message
else
set channel bits used per color to 1
get image width
get image height
initialize hit-checkt array
initialize random number generator using the hushed password as seed
read DATA_STAMP
read stego hearder version
read fixed header length
get file name length
data length = data header + filename length
channel Bits used = channel Bits used by header
re-initialize hit-checkt array based on read channel Bits used
if the current channel Bits used > 1
currentBit position = currentBitread
for i = 0; i<imagewidth; i++
for j=0; j<imageheight; j++
maintain the current bit hits
end;
set bitSet length to 8
for each bit in the secret message byte
```

```
select a random pixel
select a random pixel color channel
select a random color channel bit
read bit
insert bit to bitSet
loop
end;
if bytes read !=message.length
display error message
close stream
return data
end.
```

4.7 Use-Case Diagrams

These have were used to describe what the system does from the point of an external observer. The emphasis is on what a system does rather than how. Figures 20 and 21 illustrates the Embed and the decode use case diagrams respectively.

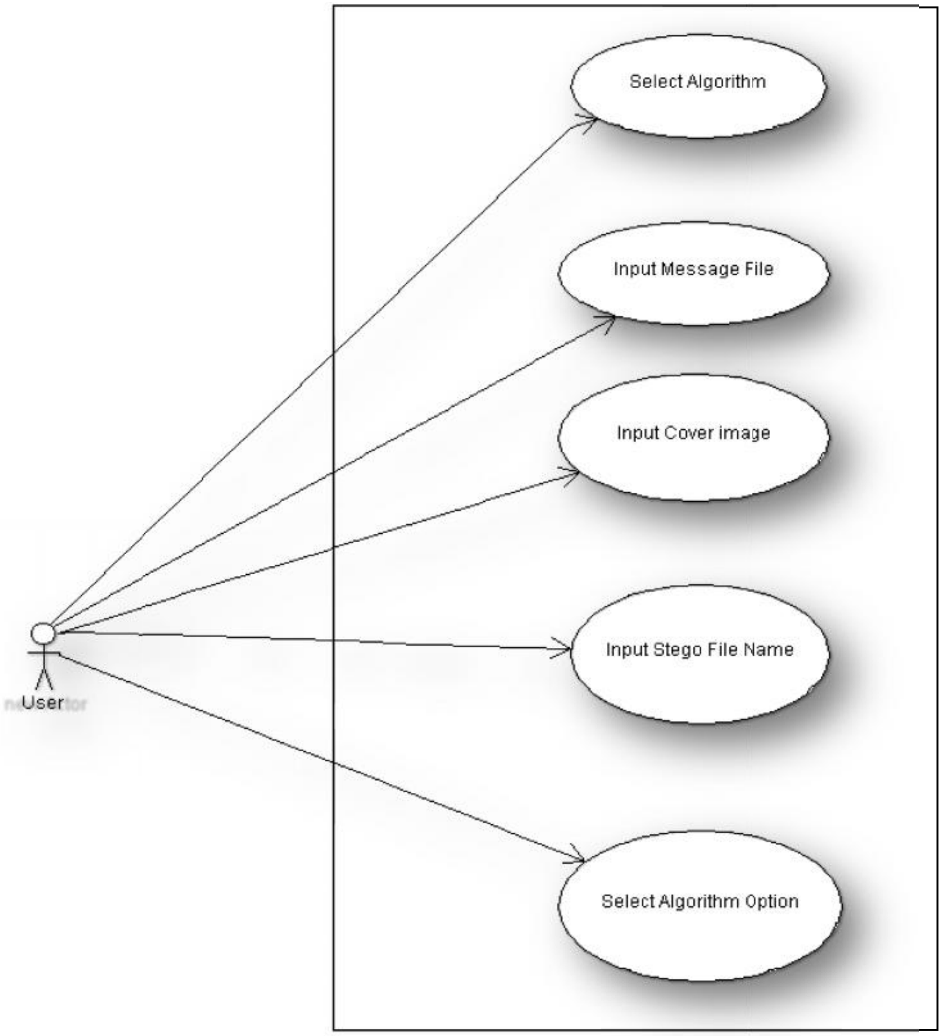


Figure 20 Embed use case diagram

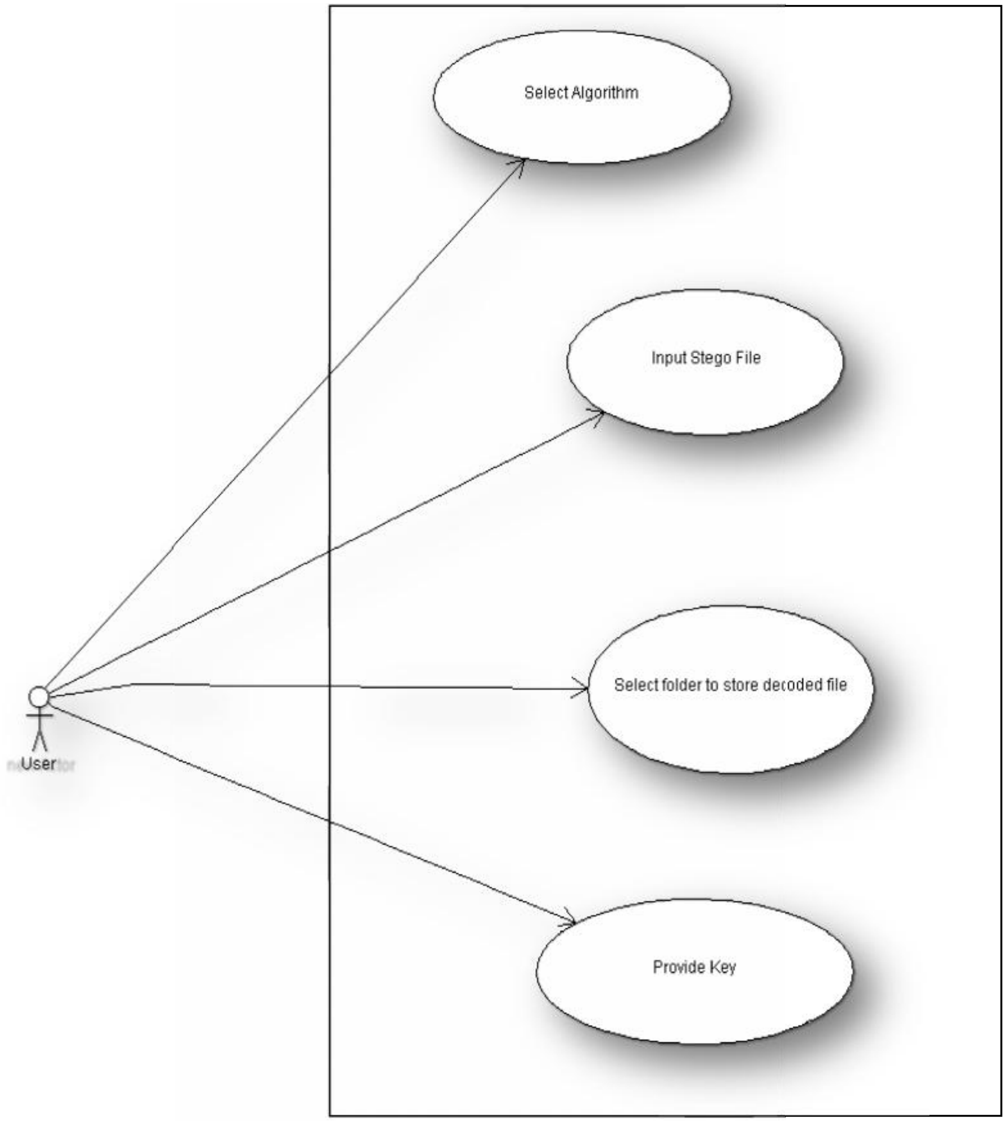


Figure 21 Decode use case diagram

4.8 Activity Diagrams

Activity diagrams were used to show the sequence of activities in the steganographic process, including sequential and parallel activities, and decisions that are made (Kendall, 2008). Figures 22 and 23 illustrates the Embed and the decode activity diagrams respectively.

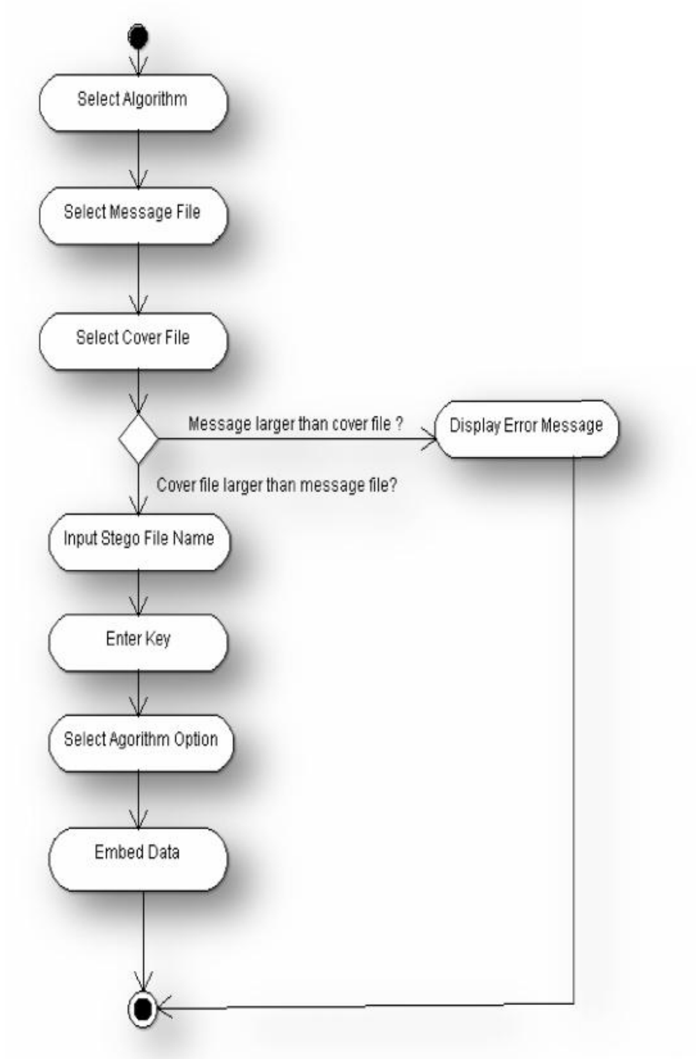


Figure 22 Embed activity diagram

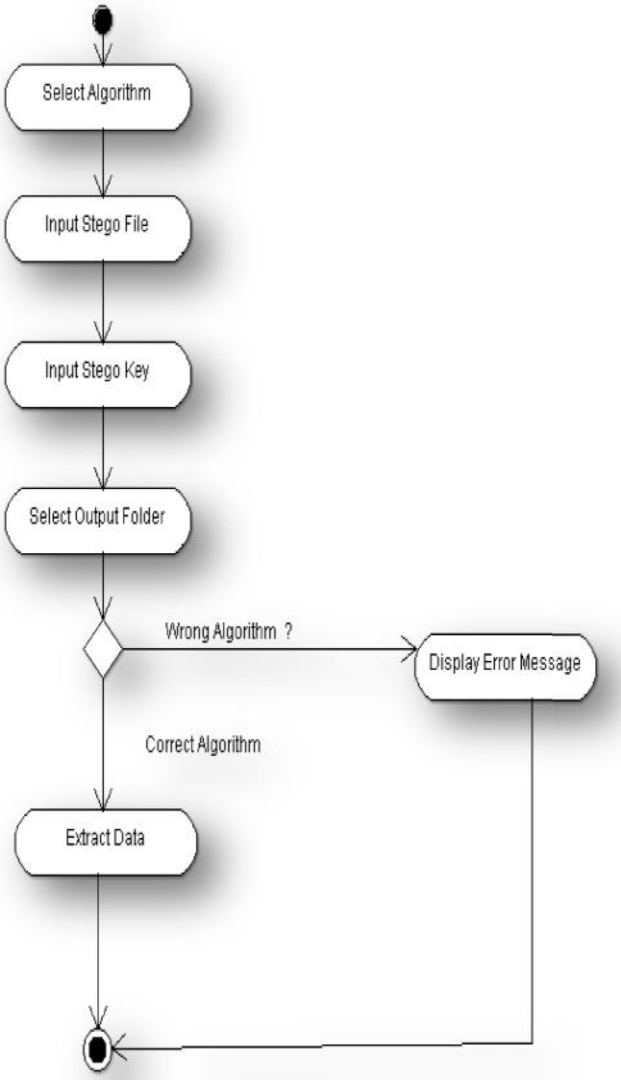


Figure 23 Decode activity diagram

4.9 Sequence Diagrams

These were used to derive the interactions, relationships, and methods of the objects in the system. They show the overall pattern of the activities or interactions in use cases

(Kendall, 2008). Figures 24 and 25 illustrates the Embed and the decode activity diagrams respectively.

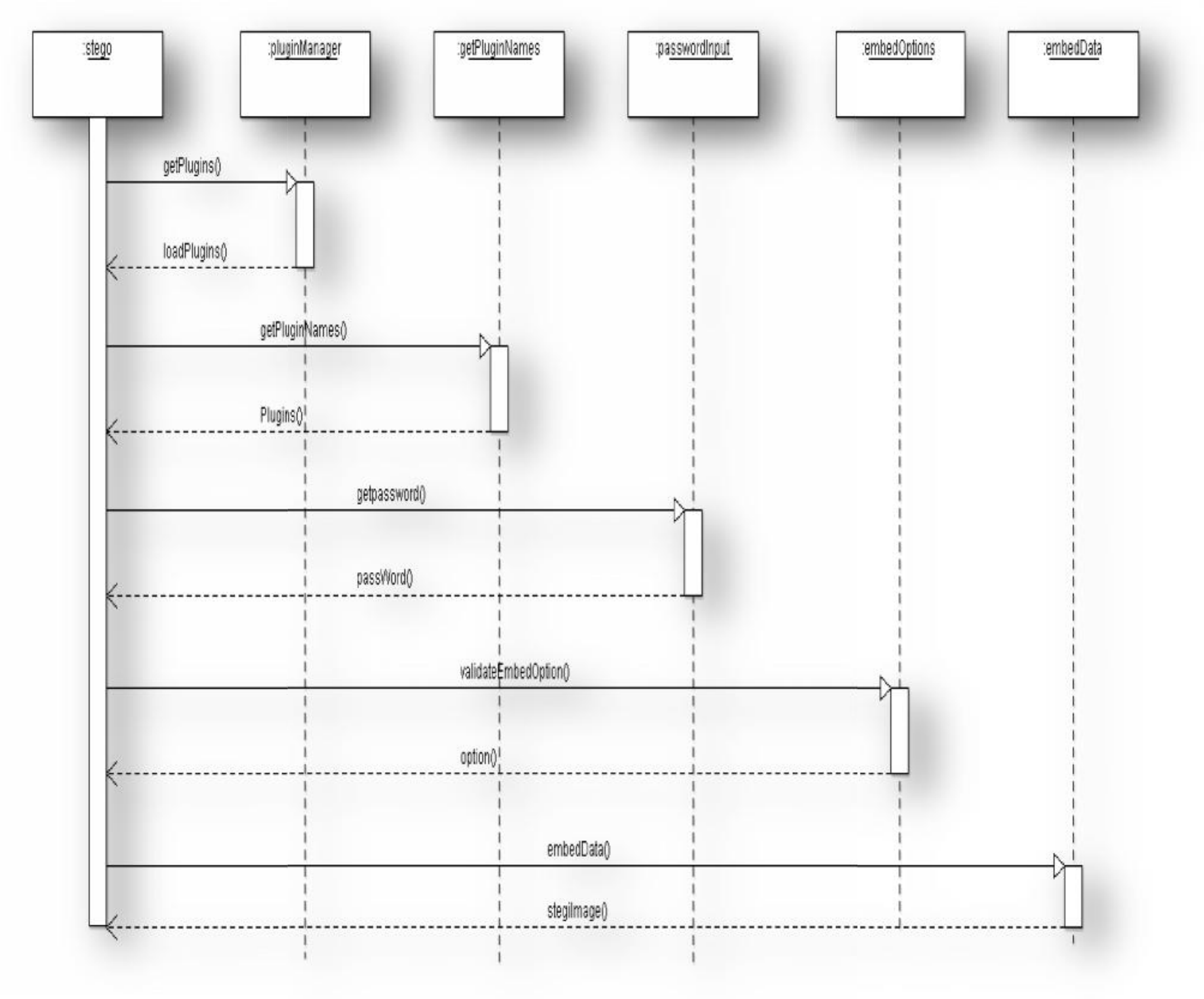


Figure 24 Embed sequence diagram

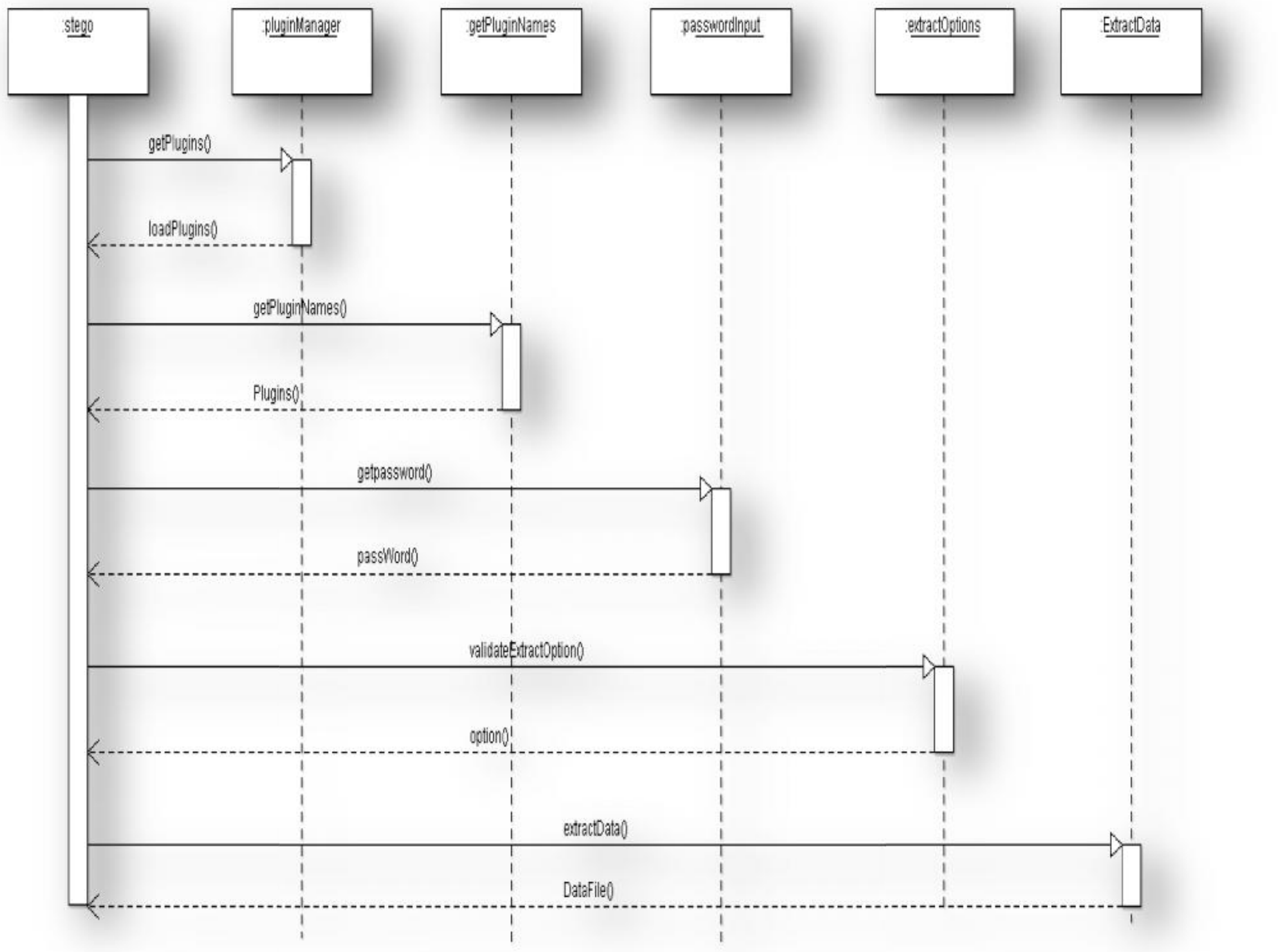


Figure 25 Extract sequence diagram

4.10 Class Diagrams

Class diagrams were used in this system to show the static features of the system and do not represent any particular processing. Figures 26 through to 30 shows the nature of relationships between the various classes that were used to implement the prototype.

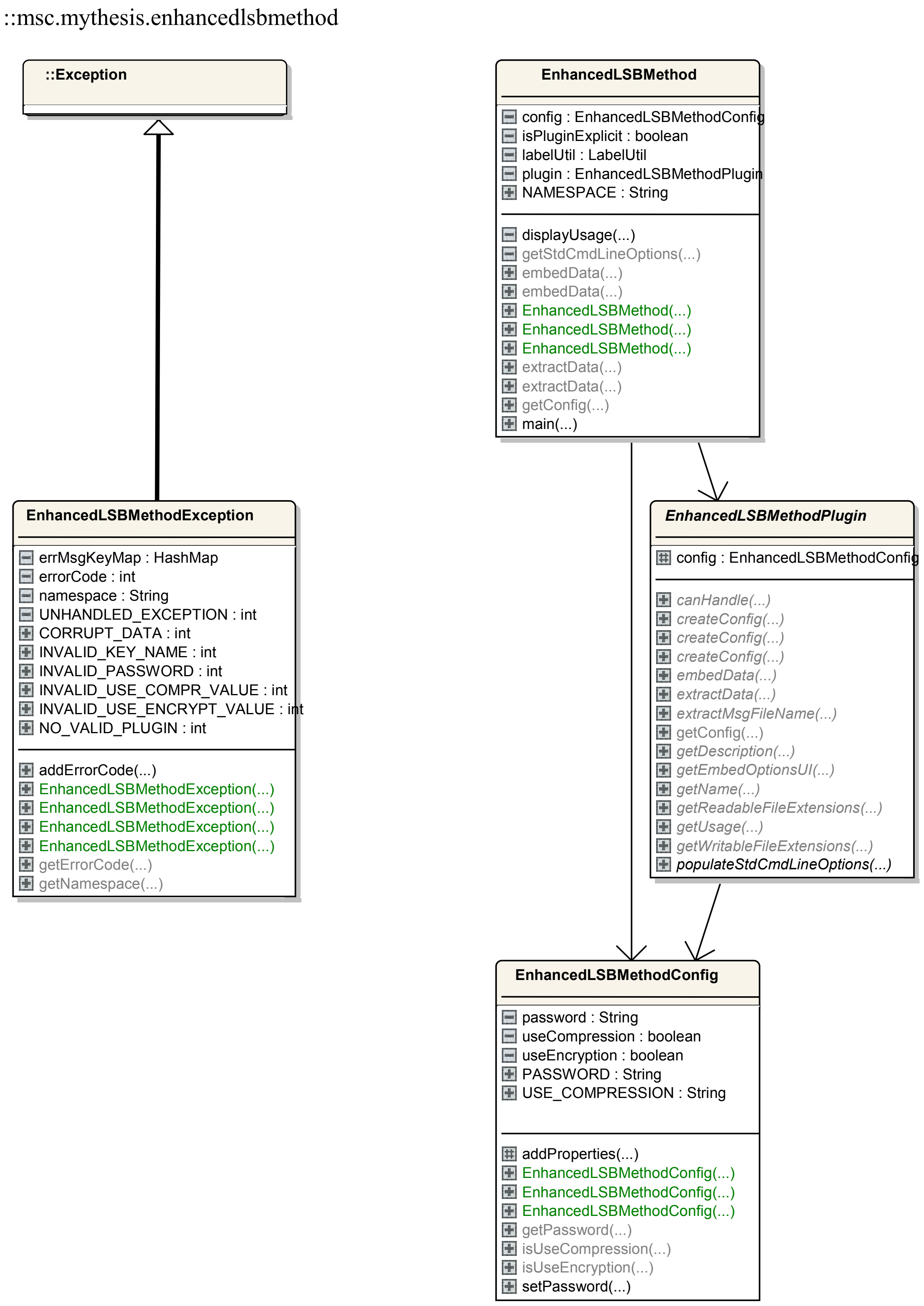


Figure 26 Enhancedlsbmethod class relationship diagram

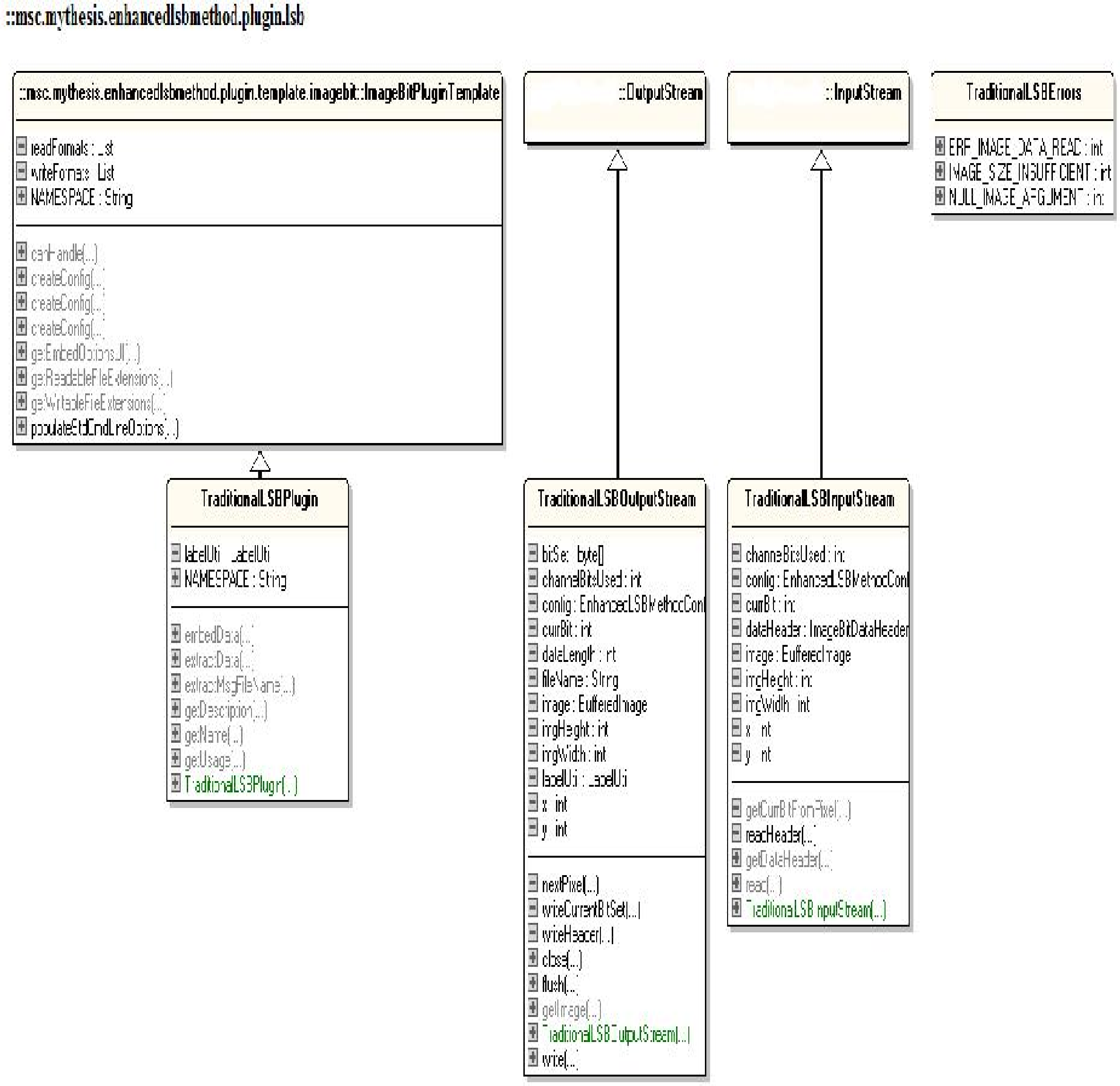


Figure 27 Enhancedlsbmethod.plugin.lsb class relationship diagram

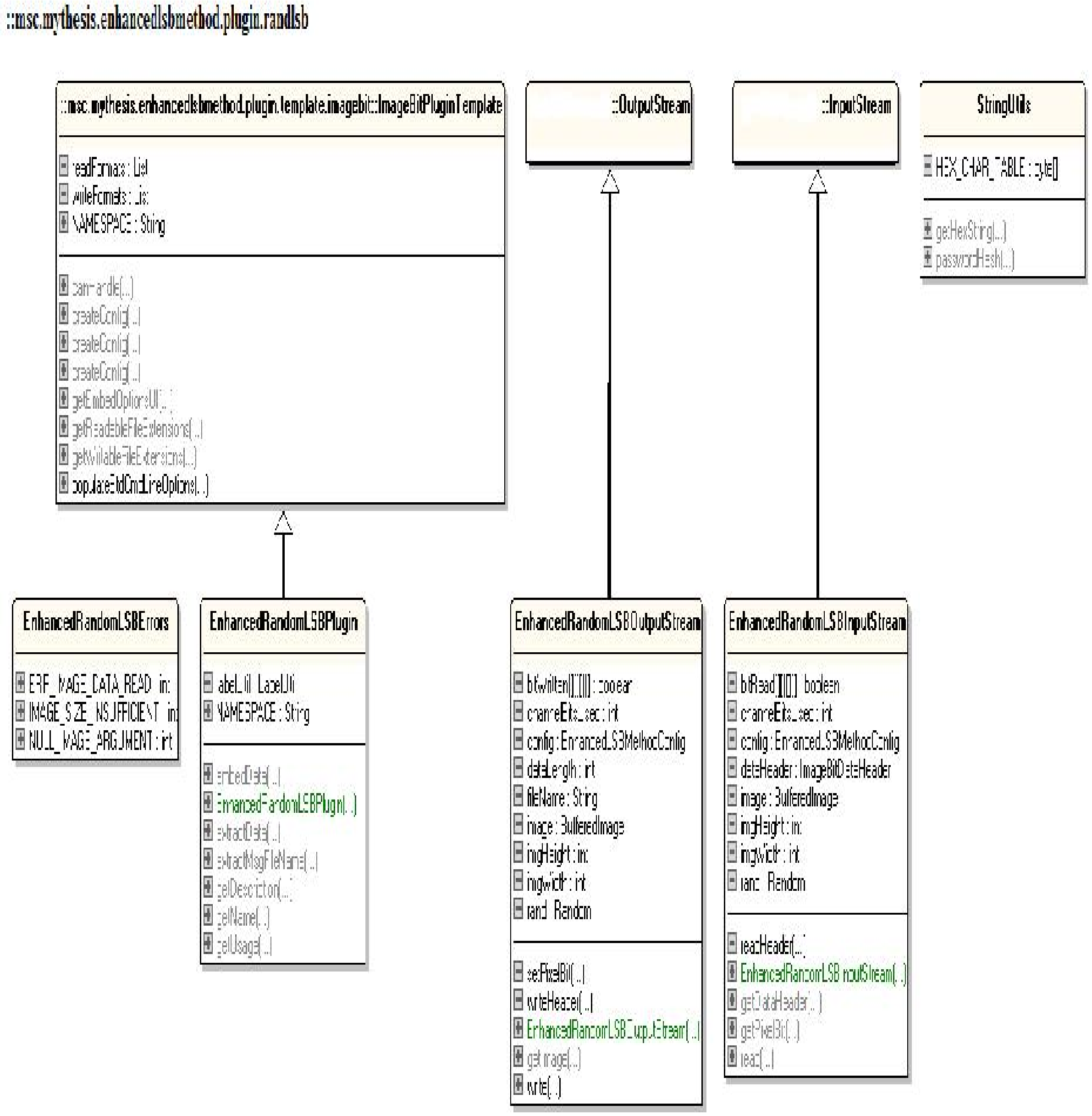


Figure 28 Enhancedlsbmethod.plugin.randlsb class relationship diagram

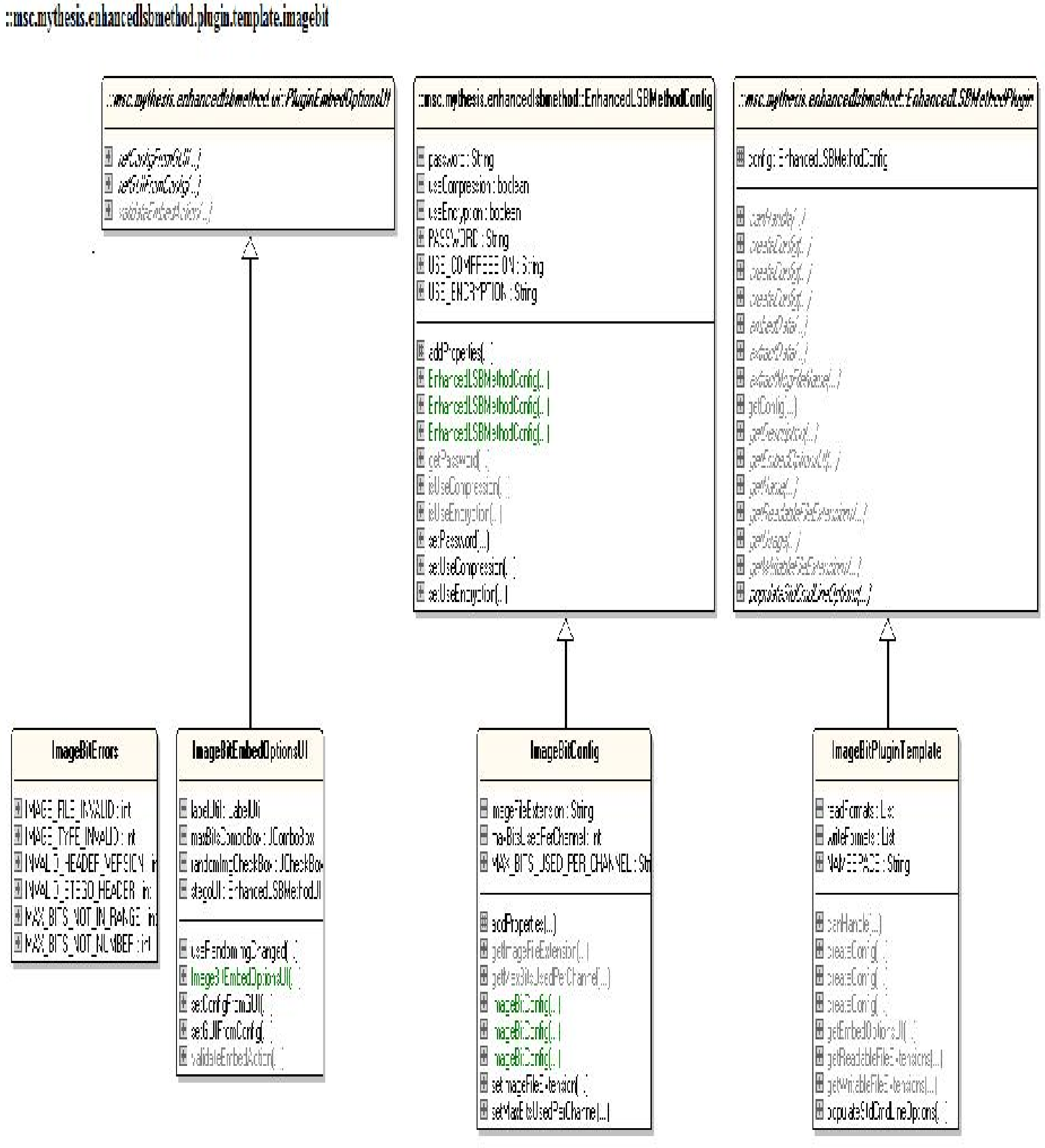


Figure 29 Enhancedlsbmethod.plugin.template.imagebit class relationship diagram

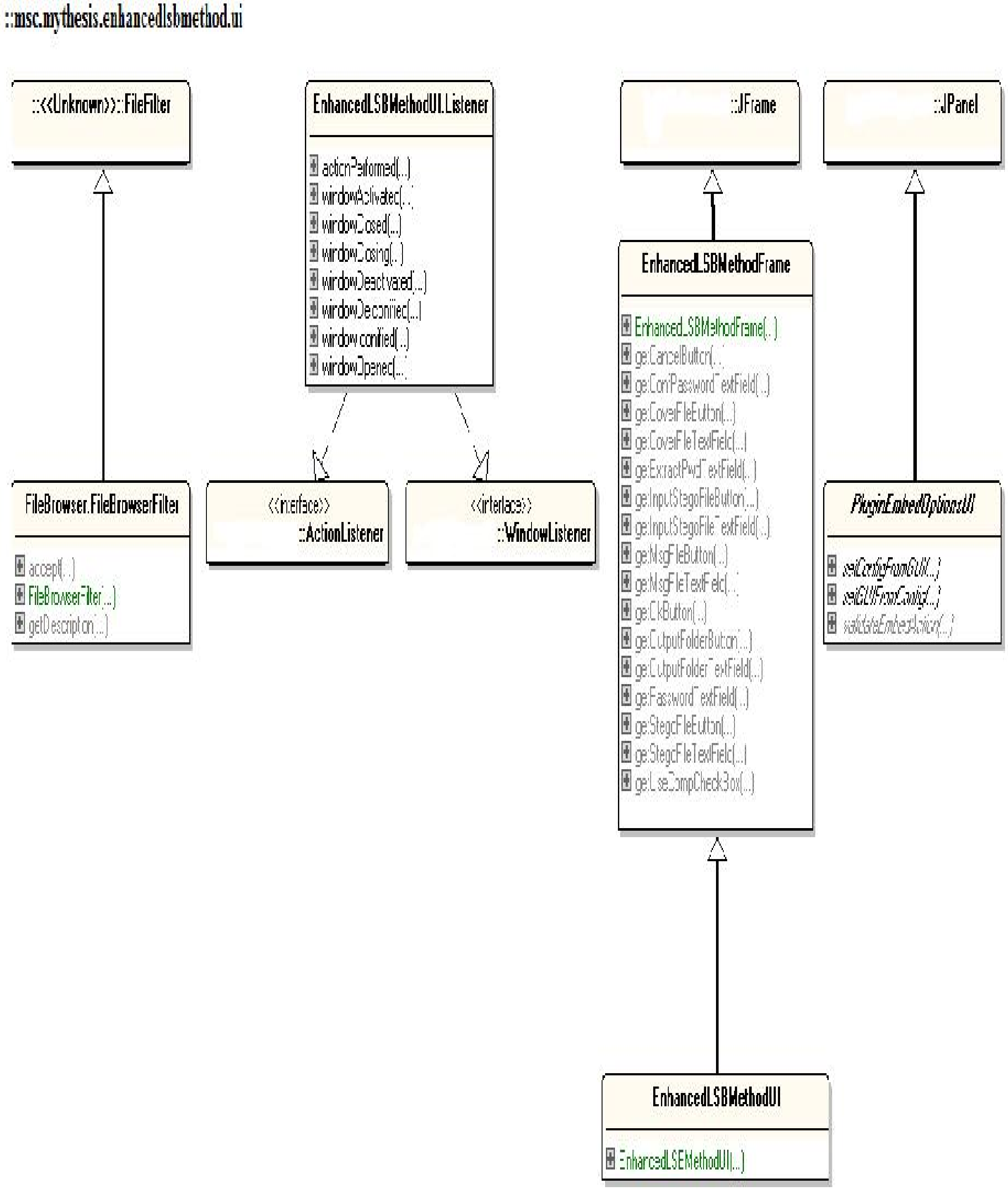


Figure 30 Enhancedlsbmethod.ui class relationship diagram

4.11 User interface design

The user interface design for the proposed prototype adopts a simplistic, purposeful and minimalistic approach. According to Larry and Lucy (1999), the user interface design should be organized purposefully, in meaningful and useful ways based on clear, consistent models that are apparent and recognizable to users, putting related things together and separating unrelated things, differentiating dissimilar things and making similar things resemble one another. The design should make common tasks simple to do, communicating clearly and simply in the user's own language, and providing good shortcuts that are meaningfully related to longer procedures.

The visibility of the design was implemented by ensuring that the user is not distracted by extraneous or redundant information. The design also ensures that the user is not overwhelmed with too many alternatives to avoid confusing him with unneeded information.

The interface design keeps users informed of actions, interpretations, changes of state and errors or exceptions that are relevant and of interest to the user through clear, concise, and unambiguous language familiar to users. It also endeavors to maintain consistency with purpose rather than merely arbitrary consistency, thus reducing the need for users to rethink and remember (Larry and Lucy, 1999).

Figures 31 and 32 shows the interface forms for both input and output.

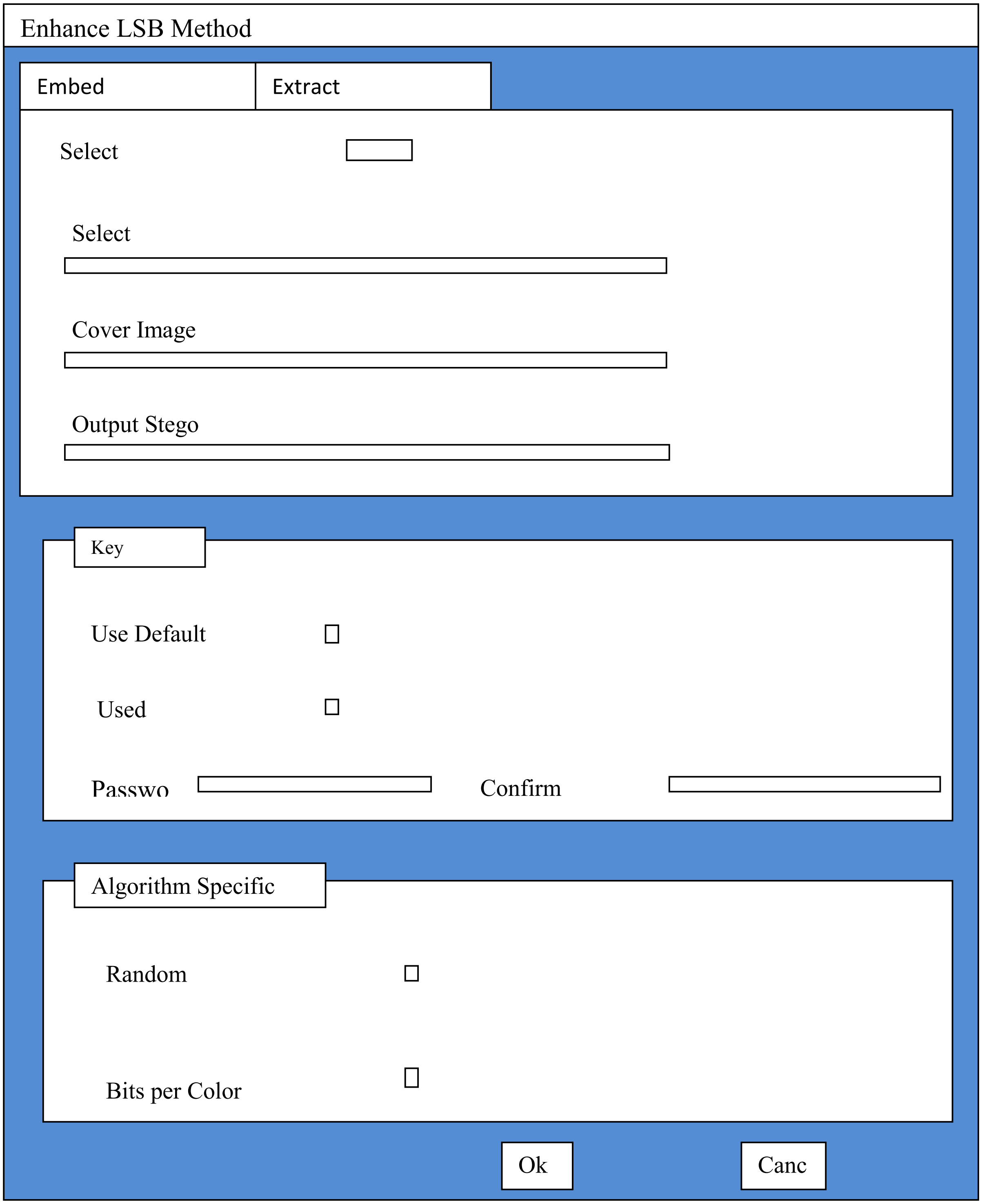


Figure 31 Input interface design

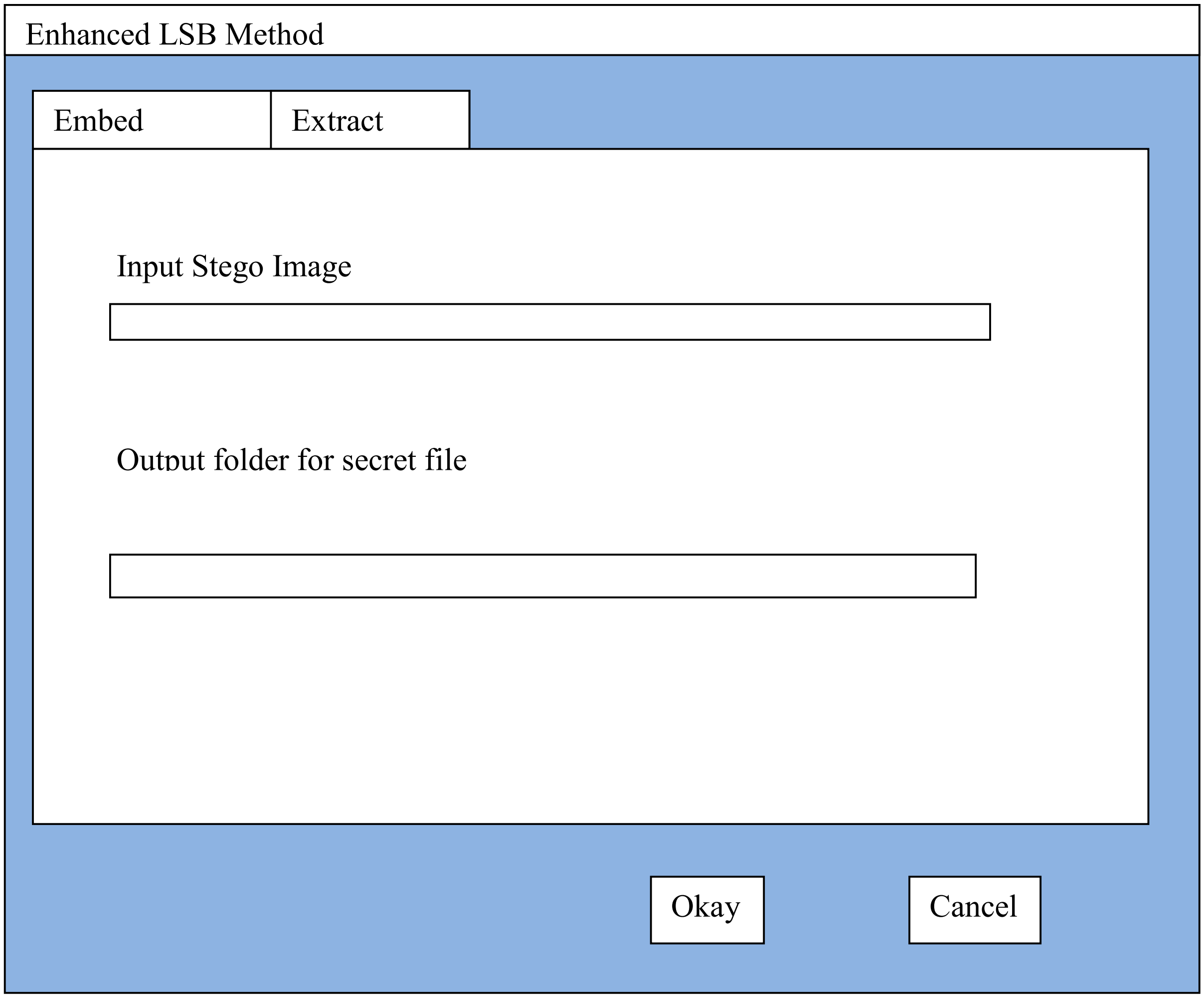


Figure 32 Output interface design

4.12 Conclusion

The standard minimal LCG (Park and Miller, 1988) pseudo random number generator used in designing the algorithm for the implementation of the proposed method has a unique advantage over the traditional LCG generator and other pseudorandom number generators like the mid-square generator, as it can be implemented without any division instructions, making it faster and also saving a lot on computer memory.

CHAPTER FIVE

5.0 IMPLEMENTATION

5.1 Introduction

With respect to the major goal of this study which is improving imperceptibility and hiding capacity in the LSB method, and the subsequent algorithm developed and incorporated in the presented prototype, the software was named "Enhanced LSB Method". Therefore from here on, the software developed in this study will continually be referred to as Enhanced LSB Method.

5.2 Development environment and tools

To guarantee platform independence i.e "Write once, run anywhere", the Java platform was used in building Enhanced LSB Method. Being an image steganography application, the choice of Java is was not unusual. It has a rich set of functionality for digital imaging. In particular, Java incorporates an advanced Imaging Application Programming Interface (API) that allows sophisticated, high-performance image processing to be incorporated into Java applets and applications. This API implements a set of core image processing capabilities including image tiling, regions of interest, deferred execution and a set of core image processing operators, including many common points, area, and frequency domain operators. Java as a programming language is also built into the popular web browsers, which places it on virtually every Internet-connected PC in the world.

Java also has a built in Message Digest 5 (MD5) hush function support required by the steganographic modules of the software.

5.3 System requirements

Enhanced LSB Method was designed, developed, and tested on a clone Intel Pentium-IV-powered computer running Microsoft Windows XP. This means that it will perfectly execute and run even on reasonably old PC. However, since Enhanced LSB Method does carry out a rather extensive analysis of the images during embedding and extraction processes, it is advisable to use faster computers particularly if the images used have high resolution which is the most ideal in this case.

The minimum estimated computer setup requirements for Enhanced LSB Method are as outlined below.

i. 2.0 GHz processor

ii. 512 MB RAM

iii. A video card that supports 1024x768 True-Color display

iv. 14-inch color monitor

v. Microsoft Windows operating system

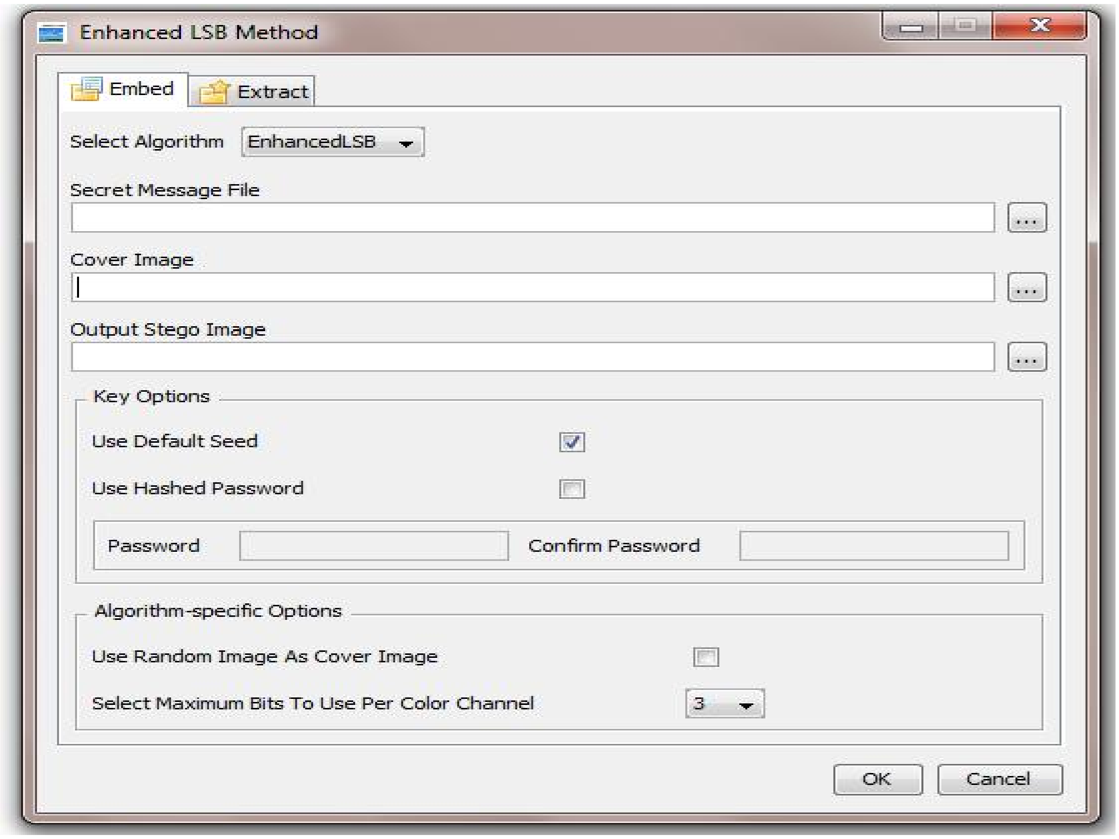
If faster embedding is required particularly where extremely large images are used, a more powerful processor will be necessary.

Cover-images used to hide data should preferably be private photographs from private digital cameras. Use of the widely available images e.g those downloaded from the internet help compromise security of the stego image as potential attackers can easily compare stego-images with their original unprocessed copies for statistical differences thereby aiding in extracting or destroying the hidden data.

5.4 General interface and screen shots

5.4.1 Embedding

Enhanced LSB Method like most steganographic software follows a minimalistic interface design to make the user experience easy. The main window interface is illustrated in figure 33.



Select Algorithm EnhancedLSB

Use Default Seed ☑

Use Hashed Password □

Use Random Image As Cover Image □

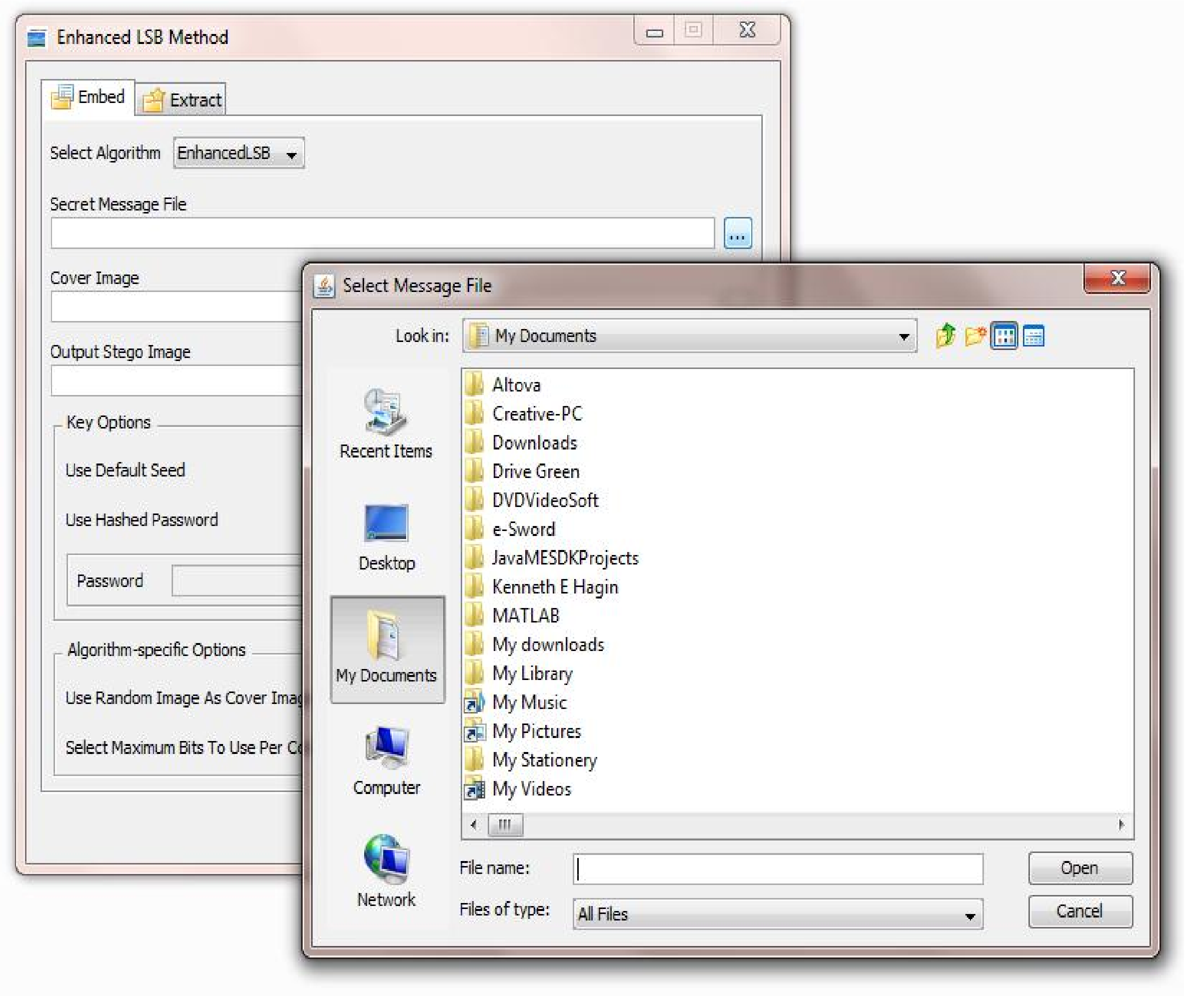
Select Maximum Bits To Use Per Color Channel 3

Figure 33 The main window interface - Enhanced LSB method (Embed option)

The main interface window is the control panel of this application where the user is able to carry out the various tasks.

The user first selects the algorithm to use. Two algorithms are implemented ie the Traditional LSB Algorithm and the new Enhanced LSB Algorithm.

The user then goes ahead to pick the secret message file that he or she intends to embeds in a cover image. The application prototype allows for browsing for the file within the computer's main storage or in external storage devices as shown in figure 34.



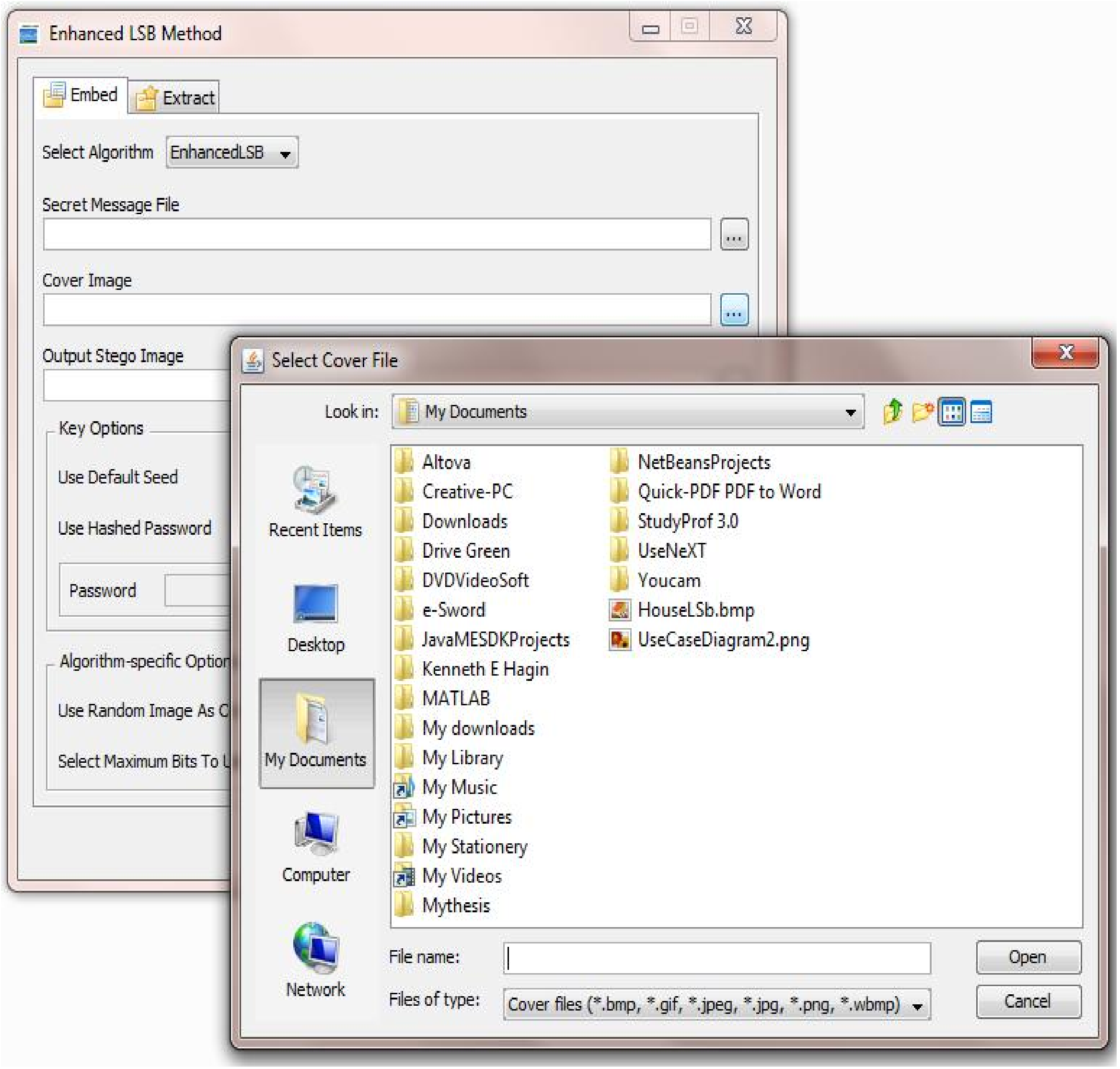
Select Algorithm EnhancedLSB

Look in: My Documents

Files of type: All Files

Figure 34 Browsing for the secret message file to embed in a cover image

Next the user picks the cover image for embedding the secret message file. The application prototype allows for browsing for the cover image within the computer's main storage or in external storage devices. This is as illustrated in figure 35.



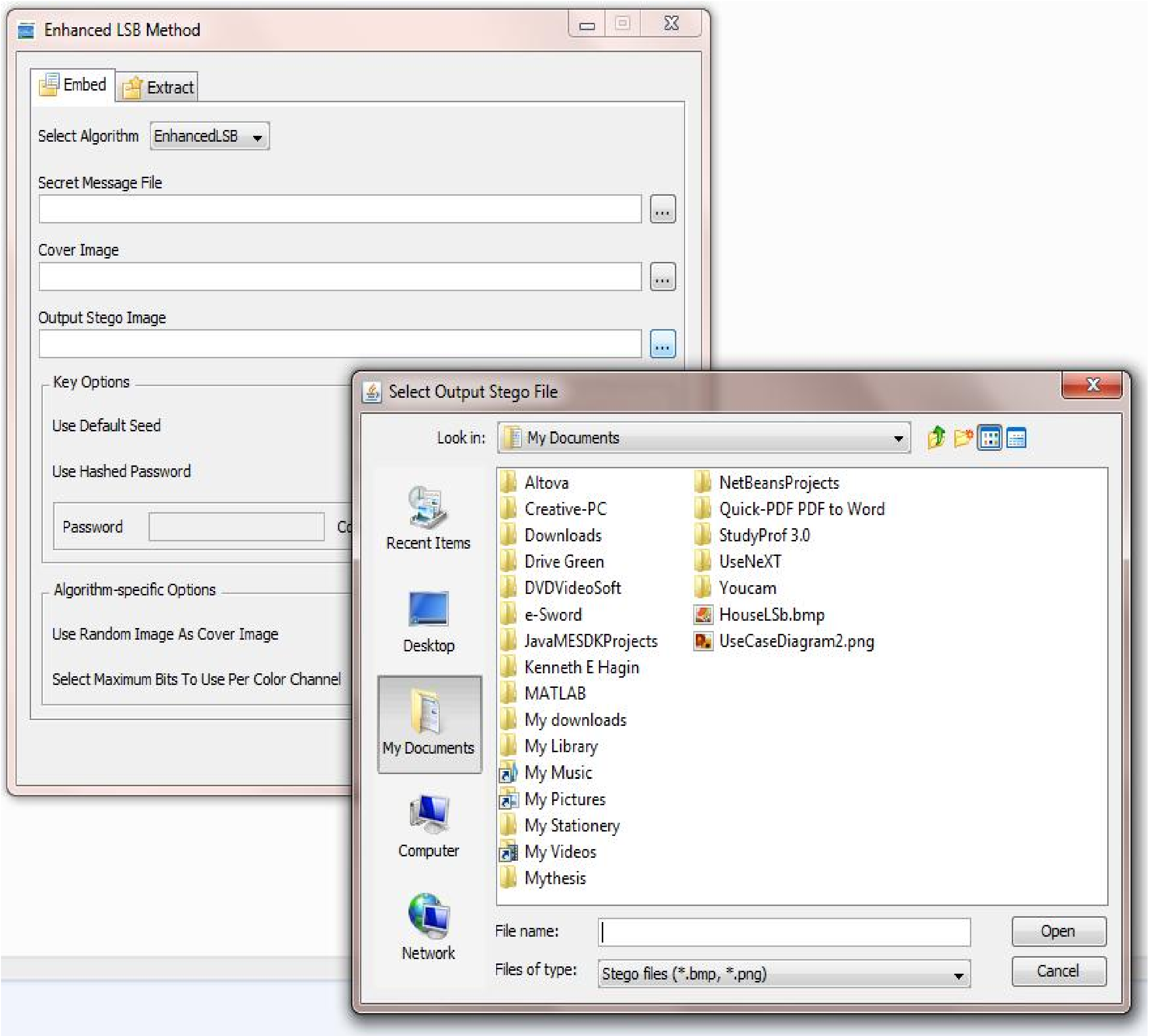
Select Algorithm: EnhancedLSB

Look in: My Documents

Files of type: Cover files (*.bmp, *.gif, *.jpeg, *.jpg, *.png, *.wbmp)

Figure 35 Selecting a cover file to hide the secret message file

After this the user provides a filename for the stego image to facilitate easier retrieval during extraction. This also makes sure the stego image assumes a different identity from the original cover image. This is as depicted in figure 36.



Select Algorithm: EnhancedLSB

Look in: My Documents

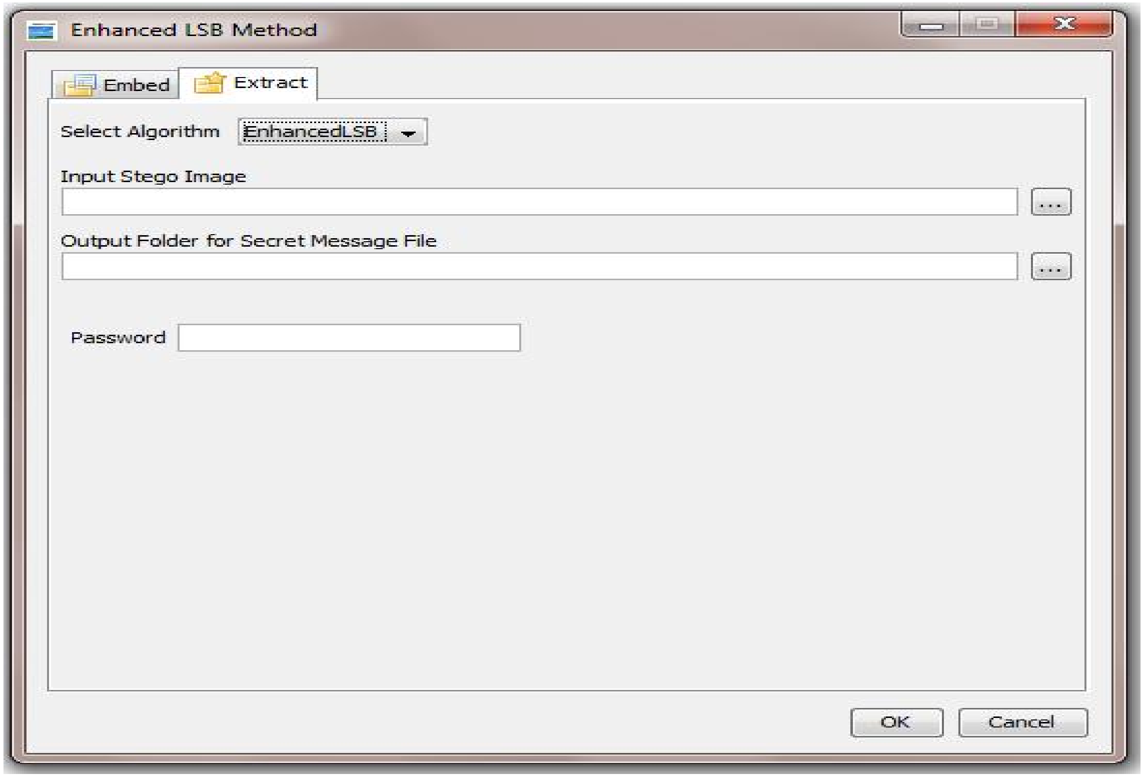
Files of type: Stego files (*.bmp, *.png)

Figure 36 selecting the output stego file

The user can then use the default seed or input a password to be used as the seed for embedding. This application prototype also provides for an option to determine how many bits to use per color channel of the cover image. The more the bits selected per color channel, the more the payload of the image. However the more number of bits used, the higher the likelihood of distorting the image and thereby reducing imperceptibility and undetectability of the hidden data.

5.4.2 Extraction

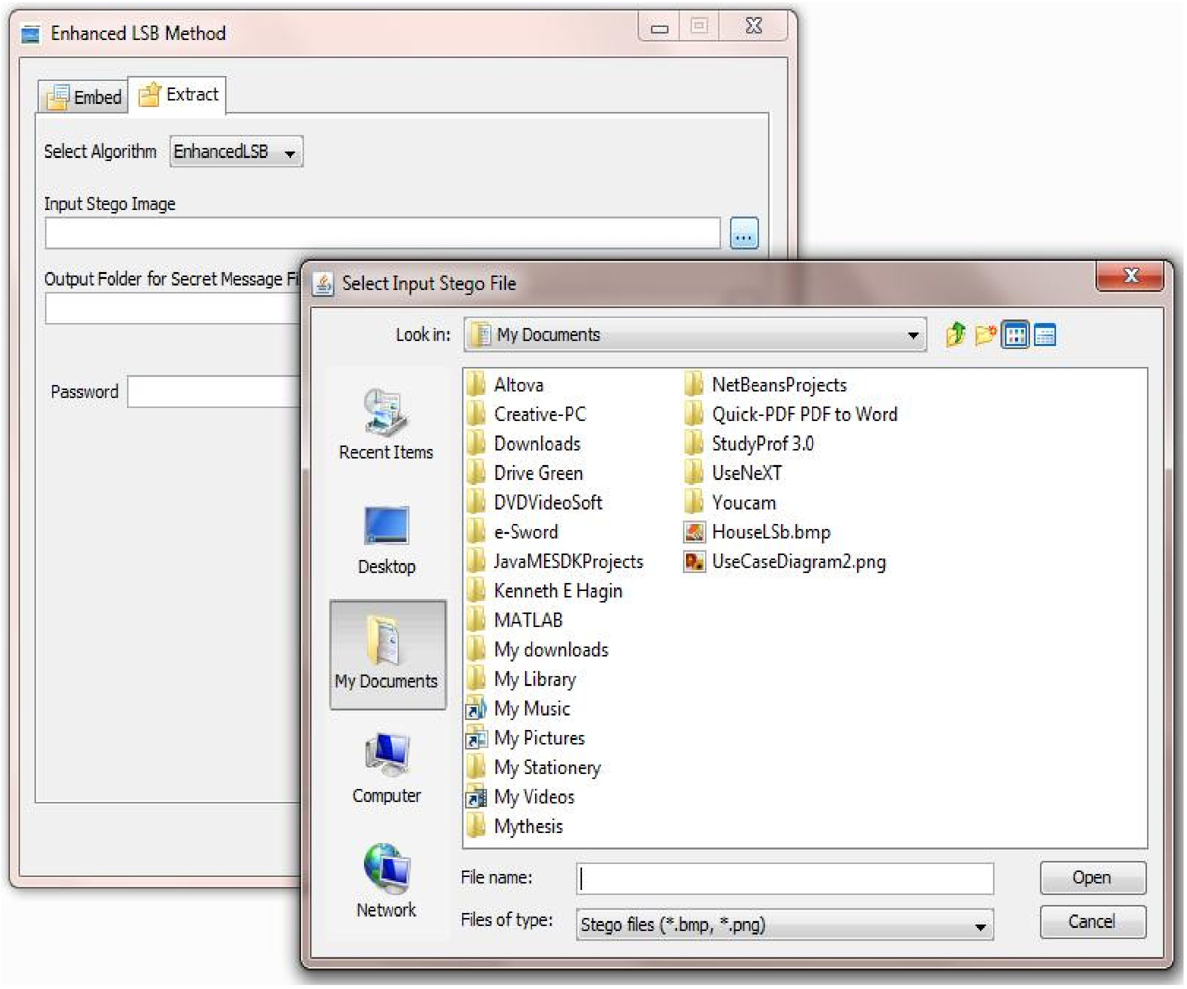
Extraction is the process of getting or retrieving the exact copy of the secret file originally embedded within a cover image. First the user selects the algorithm that was used to embed the data. However the application prototype provides for an auto-select functionality for the user for automatically picking the algorithm that was used to embed the data. The extraction interface for the developed prototype is shown in figure 37.



Select Algorithm: EnhancedLSB

Figure 37 The main window interface - Enhanced LSB method (Extract option)

Next, the user browses for the stego image ie the image file containing the hidden secret information. This application prototype provides for browsing the file within the computers’ main storage or in other external devices. This is illustrated in figure 38.

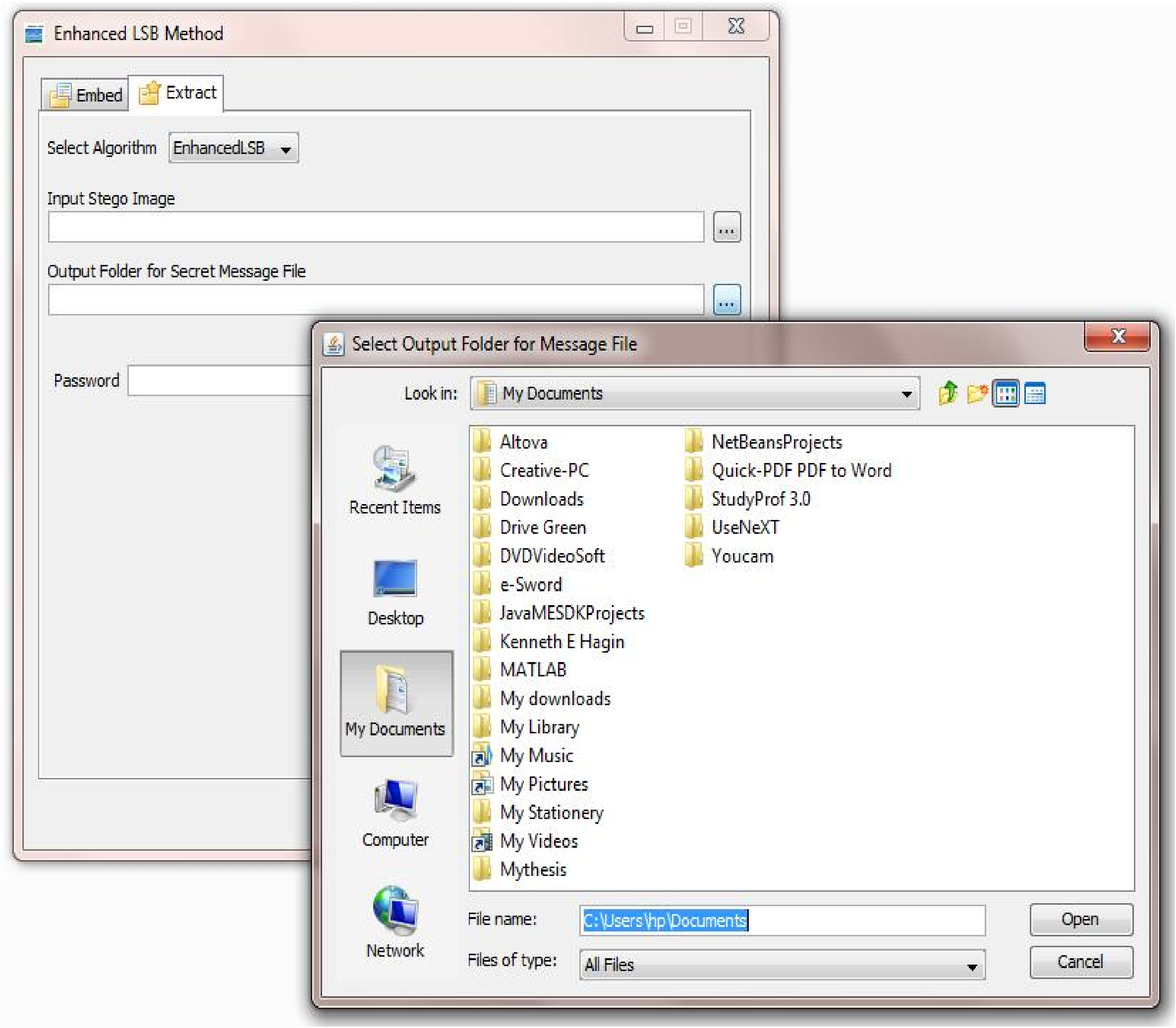

Select Algorithm: EnhancedLSB

Look in: My Documents

Files of type: Stego files (*.bmp, *.png)

Figure 38 Selecting the stego image containing the secret message

The user must then specify the output folder where to store the extracted secret message file. The file should contain the same exact filename that was used while embedding it. Figure 39 illustrates this process.



Enhanced LSB Method

Select Algorithm: EnhancedLSB

Input Stego Image

Output Folder for Secret Message File

Password

Select Output Folder for Message File

Look in: My Documents

File name: C:\Users\hp\Documents

Files of type: All Files

Figure 39 selecting the output folder where to store the extracted message

Lastly the user must provide the password that was used during the embedding process to be able to successfully extract the hidden information. The hushed value of this password is used to embed the information and therefore must be used to successfully extract the data.

5.4 Conclusion

The prototype for the proposed algorithm was successfully installed and implemented. It is advisable to use private original photographs taken with quality cameras as cover-images for this algorithm. Use of widely-available images like those downloaded from the Internet can make potential attackers to compare the stego-images to their original unprocessed copies available online thereby compromising security. The prototype is designed to carry out extensive analysis of the cover images during both embedding and extraction processes. Faster computers are therefore recommended for images with high resolution.

``

CHAPTER SIX

6.0 TESTING AND DISCUSSION OF RESULTS

6.1 Introduction

A steganographic system is said to fail if an attacker is able to detect the existence of a secret message in the cover medium or if the embedding technique used arouses the suspicions of a potential attacker. A steganographic system is therefore considered secure only when it is impossible for an attacker to detect the presence of hidden data in the cover file. The secret message therefore must be invisible both perceptually and statistically. The more identical the statistics of the cover file to those of the stego file, the more secure a steganographic system is.

Generally therefore, the statistical attributes of the cover medium should not be changed and clearly no distortions should be introduced in the cover medium during the embedding process (Venkatraman *et al*., 2004).

6.2 Testing

Chang *et al*. (2002) stated that "The better quality the stego image has, the more secure the steganography system will be". A secure steganographic system therefore refers to an imperceptible steganographic system (Cox *et al.,* 2008).This simply means that the hidden information cannot be perceived by the human visual system or by measuring statistical anomalies of the stego files. However embedding secret information in a digital image may introduce noise or modulate the cover image in some way (Venkatraman *et al.,* 2004). The important thing is to ensure that the introduced noise does not degrade the perceived quality of stego image in order to maintain the security of the steganographic system.

The steganographic prototype implemented in this study complies with the specifications that were initially set. It is able to embed and extract data from a cover image using both the traditional LSB algorithm and the enhanced LSB algorithm. Comparison and testing of the two algorithms is therefore possible.

Additionally the prototype employs a minimalistic user friendly interface displaying various messages to guide the user through the process of embedding and extracting data.

6.2.1 Objective testing

Statistical attacks are more powerful than visual attacks as they are able to reveal the tiniest modifications in the statistical properties of a image (Artz, 2001). However, one of the limitations of statistical analysis is that the statistics of a cover file may reveal that it has been modified in some way but can't tell which technique was used for modification (Watters *et al.,* 2005).

6.3 Perceptibility metrics

The following image quality metrics were employed for objective testing:

6.3.1 PSNR and MSE quality metrics

MSE measures the statistical difference in the pixel values between the original and the reconstructed image while PSNR measures the degree of similarity between two images (how two images are close to each other).

PSNR and MSE are defined as shown in equations (4) and (5) below respectively (Stoica *et al.,* 2003; Wang *et al.,* 2003):

$$MSE = \left(\frac{1}{MN} \right) \sum_{i=1}^{M} \sum_{j=1}^{N} \left(X_{ij} - \overline{X_{ij}} \right)^2 \qquad \textbf{(4)}$$

$$PSNR = 10.\log_{10} \frac{I^2}{MSE} db \qquad \textbf{(5)}$$

Where:

X_{ij} is the i^th row and the j^th column pixel in the original (cover) image,

$\overline{X}_{ij}$ is the i^th row and the j^th column pixel in the reconstructed (stego) image,

M and N are the height and the width of the image,

I is the dynamic range of pixel values, or the maximum value that a pixel can take, for 8-bit images: I=255.

The mean square error (MSE) represents the cumulative squared error between the original image and the stego-image. A lower MSE value means a better image quality ie lesser distortion in the cover image while the higher the PSNR value the better the degree of hidden message imperceptibility (Mei *et al.,* 2009).

6.3.2 Signal to noise ratio (SNR)

This is an objective quality evaluation metric whose measures are estimates of the quality of the stego image compared with the original image. The larger the value of SNR the higher the imperceptibility level of the embedding algorithm (Mei *et al.,* 2009). Equation (7) below is used to calculate the SNR between images:

$$SNR = 10 * Log_{10} \frac{\sum_{i-1}^{n} \sum_{j-1}^{m} (A_{ij})^2}{\sum_{i=1}^{n} \sum_{j=1}^{m} (A_{ij} - B_{ij})^2} \quad \textbf{(7)}$$

A_{ij} Represents one pixel in the original image (before embedding the data)

B_{ij} Represents one pixel in the stego image (after embedding the hidden data)

6.3.3 The average absolute difference (AAD)

Also referred to as mean absolute deviation, AAD was used to quantify the difference between original images and the stego images after the embedding process. A lower value of AAD gives a "cleaner" image as more noise is reduced. AAD is computed using Equation (8) below.

$$\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [f(i,j) - f'(i,j)] \qquad \textbf{(8)}$$

Where:

M and N are the height and the width of the image,

6.4 Metrics for testing robustness against steganalysis

Reed-solomon (RS) analysis

Fridrich and Goljan propose the method known as RS, for detecting the use of LSB steganography (Fridrich *et al.,* 2001). RS was used to measures the smoothness of the changes among pixels of both the cover and the stego images. The lower the value, the smoother the changes among them, or the lesser the noise of the image (Fridrich *et al.,* 2001).

Sample pair analysis (SPA)

This was used to establish the probabilities of transitions between sample pairs due to LSB embedding operations. Lower values of SPA indicate improved imperceptibility (Andrew, 2004).

Histogram analysis (First – order statistics)

The image histograms were used to help detect changes in frequency of appearance of the image colors between the original images and the stego images. More pronounced changes in the frequency of colors shows greater distortions of the original images.

6.5 Experiment and discussion of results

The experimental design

The experimental approach was used in this study to evaluate the performance of the proposed enhanced least significant bit steganoghraphy data hiding system. To test the performance of the proposed method, its efficiency in terms of imperceptibility and undetectability of hidden messages, an experiment was carried out to subject the stego images generated by the proposed system to both the objective and subjective evaluation tests discussed earlier. The results were then compared to those gotten from the traditional LSB method. In all cases the following constants were ensured:

i. All the experiments were implemented and run on a PC Pentium IV Duo core, 2.1 GHz with 2GB of RAM under the Windows 7 Home Edition operating system.

ii. Same images were used on both the methods

iii. Same secret information was embedded in each image ie equal payload

iv. Same evaluation metrics were used for each image

v. For subjective evaluation and analysis, fifteen individuals (5 digital photographers, 5 web developers and five constant web users) were used to detect differences between original cover images and the stego images generated by the enhanced least significant bit method.

vi. For objective evaluation and analysis, five digital images were used as test data files (cover images). Any digital image however can be used for this experiment provided it is original and of high quality. Table 5 shows the list of these digital images.

FILE NAME	DIMENSIONS	FILE SIZE
Banana.jpg	685 x 514 Pixels	157 Kilo Bytes
Dancers.jpg	685 x 457 Pixels	224 Kilo Bytes
Graduation.jpg	685 x 457 Pixels	161 Kilo Bytes
Office.jpg	685 x 457 Pixels	163 Kilo Bytes
zhbackground.bmp	685 x 610 Pixels	1.19 MB

Table 5 Test data images

These cover images are depicted in figures 40 to figure 44.0

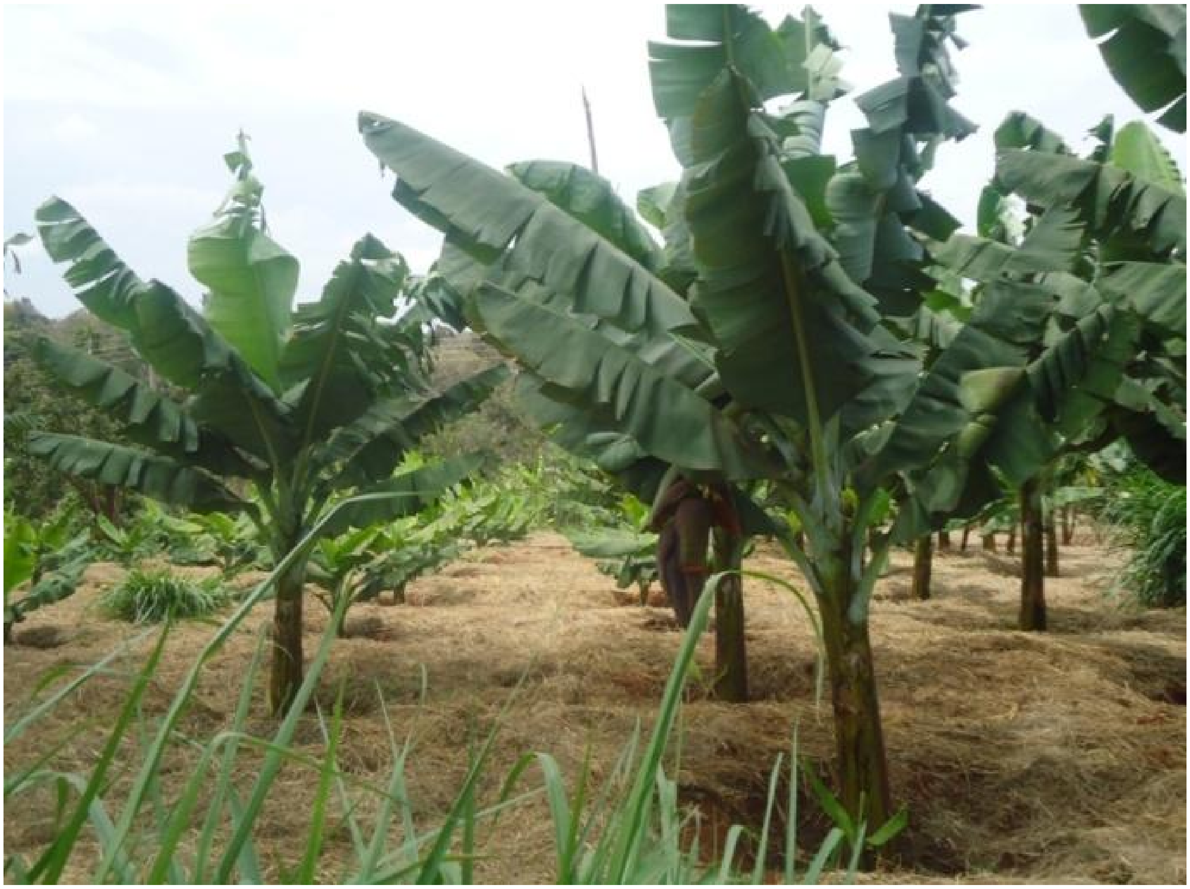


Figure 40 Banana.jpg

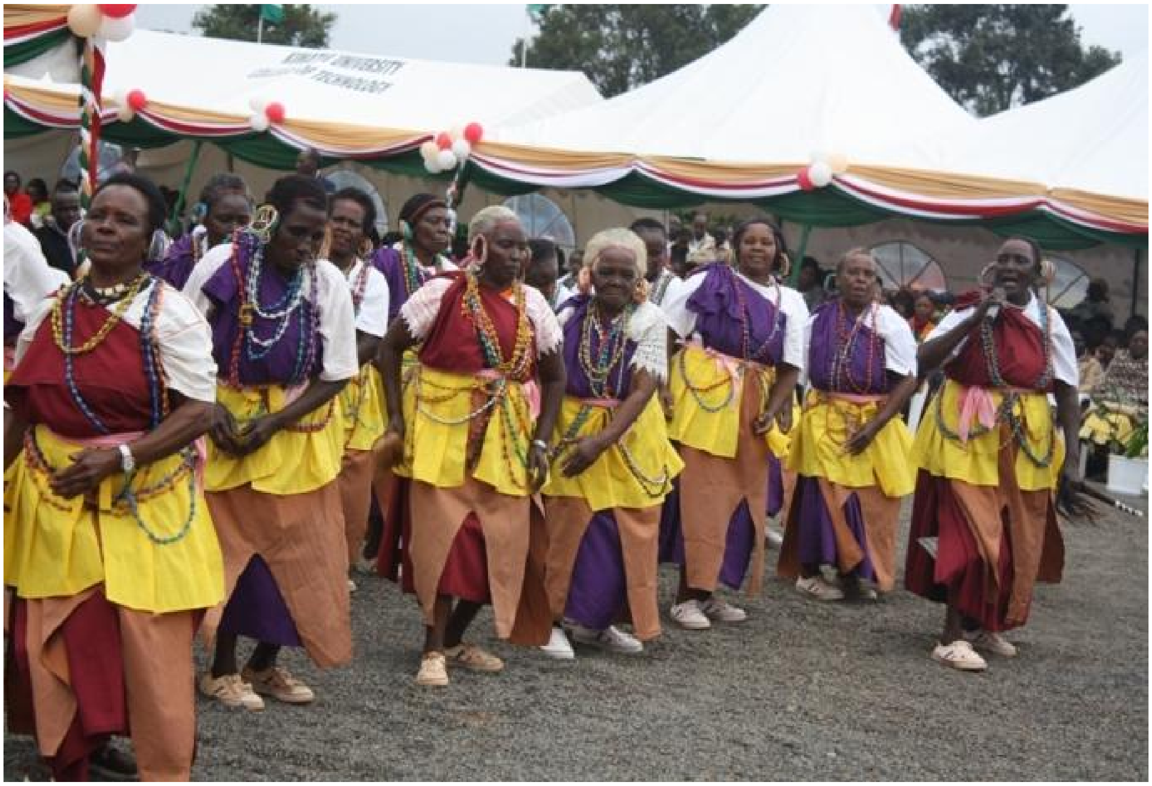


Figure 41 Dancers.jpg

Figure 42 Graduation.jpg

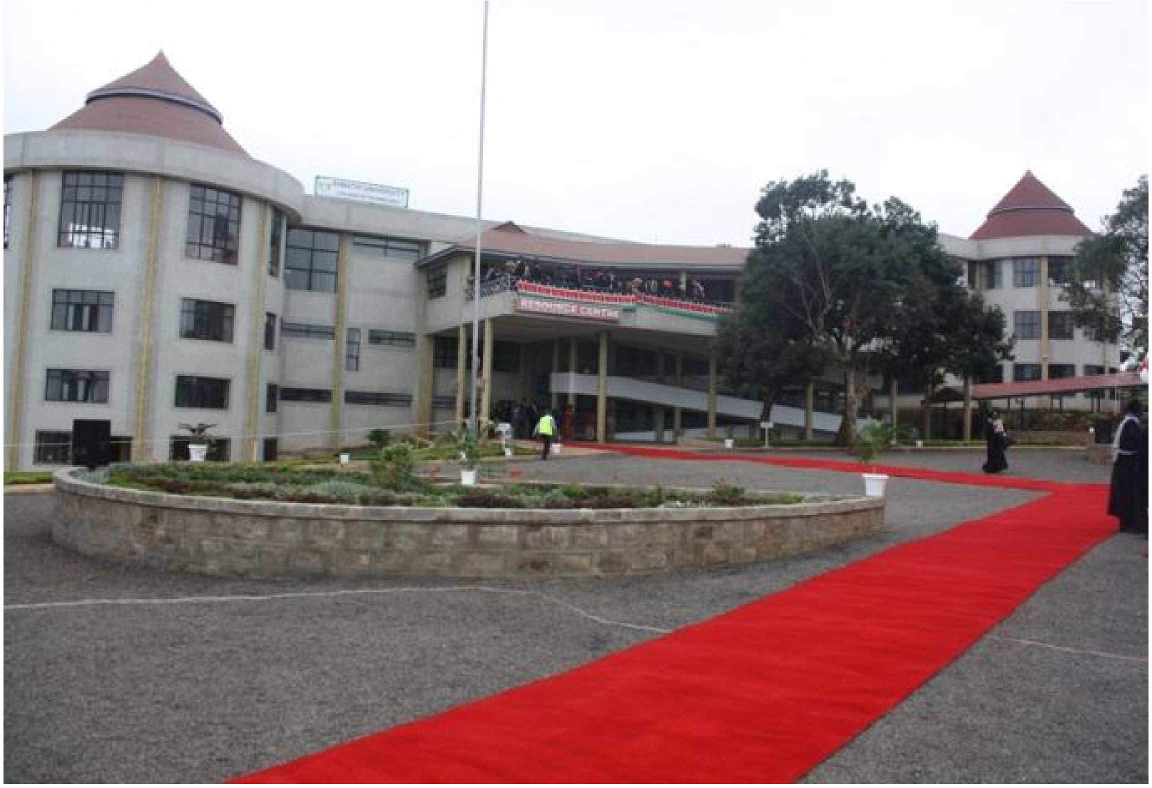


Figure 43 Office.jpg

Figure 44 zhbackground.bmp

In this experiment, the specific steganography method used to embed the secret data was assumed to be the independent variable (in this case the traditional least significant bit method and the proposed enhanced least significant bit method). In order to evaluate the efficiency of the proposed steganography method, eight objective evaluation metrics (dependent variables) all of which measure the imperceptibility level are considered. Accordingly, for each steganography method (the traditional least significant bit method and the proposed enhanced least significant bit method) and for each cover image, we measure the value of each dependent variable. Having just the values of the dependent variables of the proposed enhanced steganography method is not enough since we cannot evaluate these results. Therefore, we will compare the values of these dependent variables for the traditional least significant bit method with those of the proposed method in order to effectively evaluate it.

The *Steganography Studio* version 1.0 (SS) and the *Digital Invisible Ink Toolkit* 1.5 (DIIT) are the two steganalysis tools used for the objective evaluation tests in this study. These are among the latest readily available open source steganalysis software tools on the internet both of them having been released in the year 2011.

6.5.2 Results of perceptibility metrics

Peak signal to noise ratio (PSNR)

Table 6 shows a comparison of the PSNR values of the five stego images for both the traditional LSB and the Enhanced LSB methods.

Image	Traditional LSB (db)	Enhanced LSB (db)	Difference (db)
Banana.jpg	63.4	66.0	2.6
Dancers.jpg	62.8	65.4	2.6
Graduation.jpg	56.4	57.8	1.4
Office.jpg	62.9	65.5	2.6
zhbackground.bmp	57.6	59.4	1.8

Table 6 The PSNR - traditional method vs enhanced LSB method

These results are as illustrated in figure 45.

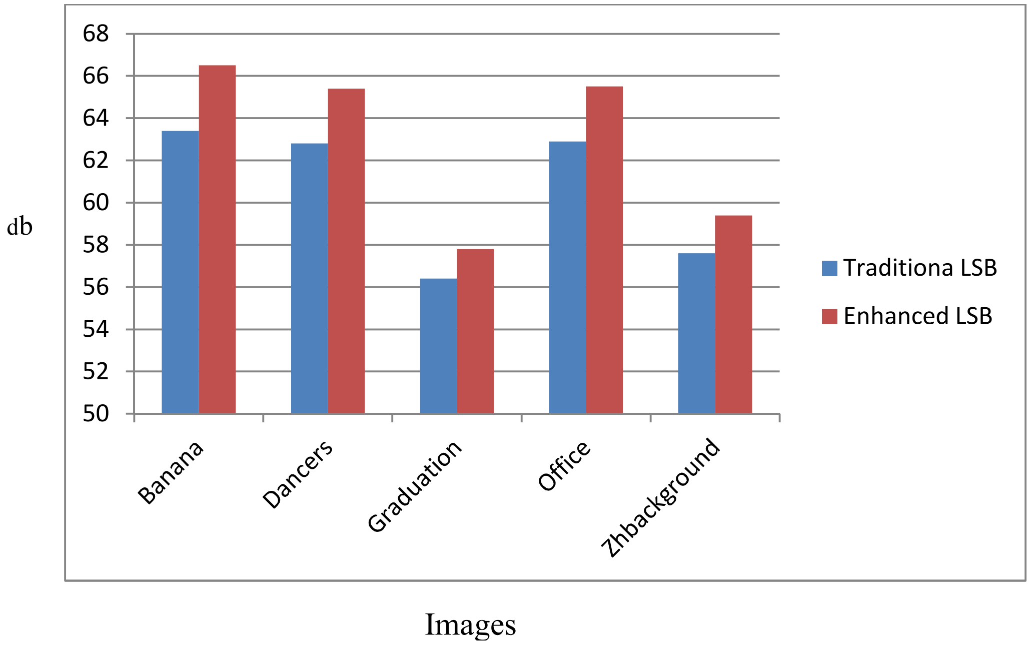


Figure 45 The PSNR of stego images - traditional vs enhanced LSB

Every image tested registered a higher PSNR when the enhanced LSB was used compared to when the Traditional LSB was used. These results therefore clearly shows that the enhanced LSB embedding method improves on imperceptibility of the hidden data since a higher Peak Signal to Noise Ratio (PSNR) always indicates improved imperceptibility (Mei Jiansheng *et al* .2009).

Signal to noise ratio (SNR)

Table 7 shows a comparison of the SNR values of the five stego images for both the traditional LSB and the Enhanced LSB methods.

Image	**Traditional LSB (db)**	**EnhancedLSB (db)**	**Difference (db)**
Banana.jpg	57.9	60.5	2.6
Dancers.jpg	57.3	59.9	2.6
Graduation.jpg	51.3	52.8	1.5
Office.jpg	58.4	61.0	2.6
zhbackground.bmp	48.8	50.6	1.8

Table 7 The SNR - traditional method vs enhanced LSB method

Again for all the seven test data images used, each registered a higher SNR when the enhanced LSB method was used compared to when the Traditional LSB method was used. For any embedding algorithm or method, the larger the value of SNR the higher the imperceptibility level (Mei Jiansheng *et al* .2009).

These results therefore establish the superiority of the enhanced LSB embedding method compared to the traditional LSB Method.

The results are further illustrated in figure 46.

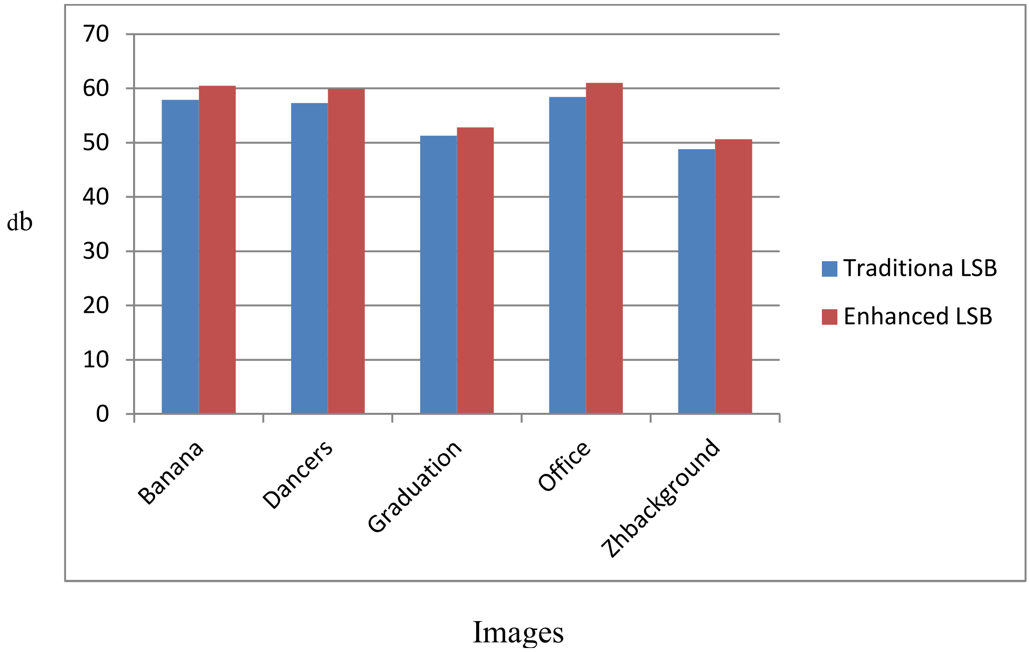


Figure 46 The SNR of stego images - traditional vs enhanced LSB

Mean square error (MSE)

Table 6.5.2.3 below is a summary of the MSE of the seven stego images after hiding data using the least significant bit method and the enhanced method as generated by the Digital Invisible Ink Toolkit. From all the five test data images used, each recorded a lower mean square error when the enhanced LSB was used as compared to the traditional LSB method. A lower MSE value means a better image quality ie lesser distortion in the cover image (Mei Jiansheng *et al* .2009). This means that stego images generated by the enhanced LSB method have lesser distortions compared to those generated by the traditional LSB method and hence improved imperceptibility. This is in table 8.

Image	Traditional LSB (db)	Enhanced LSB (db)	Difference (db)
Banana.jpg	0.268	0.146	0.122
Dancers.jpg	0.304	0.165	0.139
Graduation.jpg	1.34	0.961	0.379
Office.jpg	0.302	0.164	0.138
zhbackground.bmp	1.003	0.664	0.339

Table 8 The MSE - traditional method vs enhanced LSB method

This is further illustrated in figure 47.

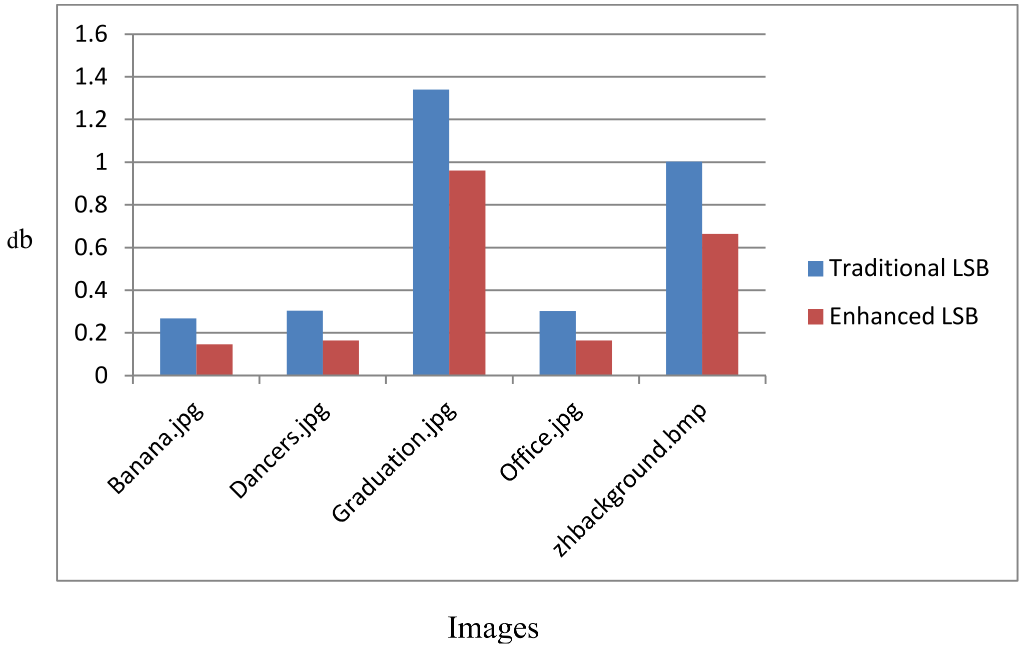


Figure 47 The MSE of stego images - traditional vs enhanced LSB

Average absolute difference (AAD)

Table 9 below is a summary of the AAD of the seven stego images after hiding data using the least significant bit method and the enhanced method as generated by the Digital Invisible Ink Toolkit.

Image	Traditional LSB (db)	EnhancedLSB (db)	Difference (db)
Banana.jpg	0.1341	0.134	0.0001
Dancers.jpg	0.1514	0.1501	0.0013
Graduation.jpg	0.6695	0.6649	0.0046
Office.jpg	0.151	0.1498	0.0012
zhbackground.bmp	0.5011	0.4992	0.0019

Table 9 The AD - traditional method vs enhanced LSB method

This is further illustrated in figure 48

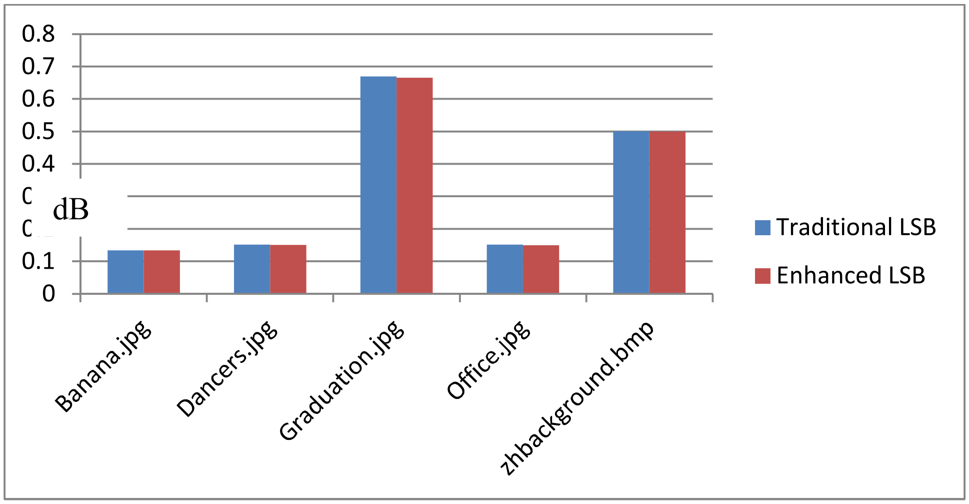


Figure 48 The AAD of stego images - traditional vs enhanced LSB

Images

The figures above clearly shows that the mean difference between the stego images created by the enhanced least significant bit method and their original cover images is smaller compared to that of stego images created by the traditional least significant bit method.

6.5.2 Results for robustness against steganalysis

High order statistics

Reed-solomon (RS) analysis

Table 10 is a summary of the RS of the five stego images after hiding data using the least significant bit method and the enhanced method as generated by the Digital Invisible Ink Toolkit.

Image	Traditional LSB (db)	Enhanced LSB (db)	Difference (db)
Banana.jpg	19.47	10.85	8.62
Dancers.jpg	16.22	10.37	5.85
Graduation.jpg	49.74	45.43	4.31
Office.jpg	17.46	10.17	7.29
zhbackground.bmp	39.33	36.37	2.96

Table 10 The RS - traditional vs enhanced LSB method

According to (Fridrich *et al.,* 2001) RS measures the smoothness of the changes among pixels (the lower the value, the smoother the changes among them, or the lesser the noise of the image). Stego images generated by the enhanced LSB method all have recorded lower values of RS compared to those generated by the traditional LSB method. Therefore these images have lesser noise and the information hidden in them more imperceptible.

The results are further illustrated in figure 49.

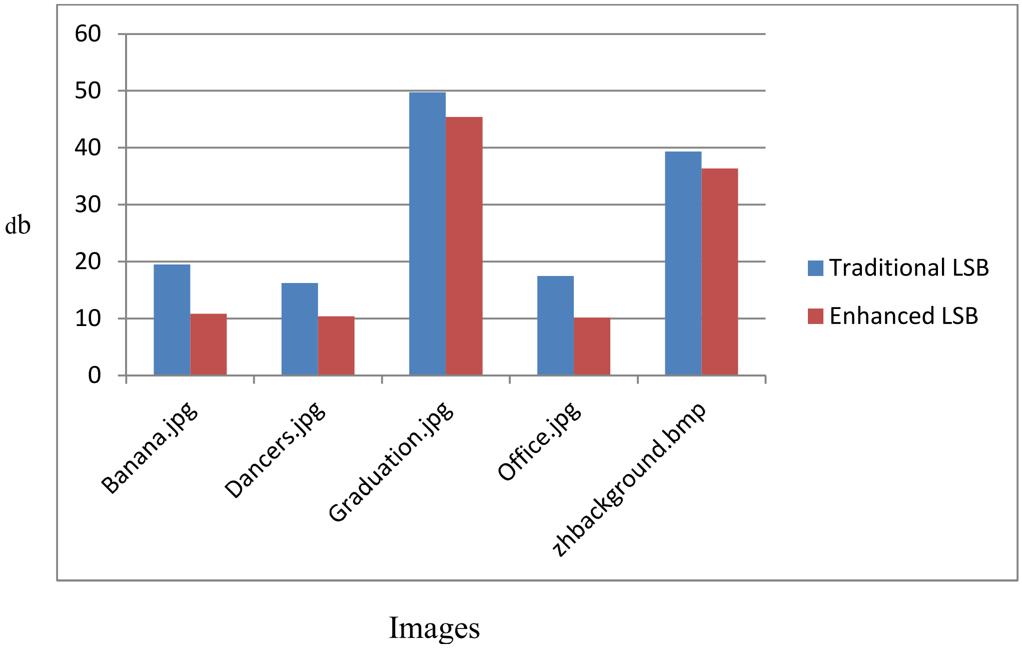


Figure 49 The RS of stego images -traditional vs enhanced LSB

Sample pair analysis (SPA)

Table 11 is a summary of the Sample Pair analysis of the five stego images after hiding data using the least significant bit method and the enhanced method as generated by the Digital Invisible Ink Toolkit.

SPA is based on probabilities of transitions between sample pairs due to LSB embedding operations. RS and SPA are the most reliable detectors of thinly-spread LSB steganography until now (Andrew, 2004). The lower the values of SPA, the higher the imperceptibility level for the embedding algorithm. This experiment therefore reveals that the enhanced LSB method is superior. These results are further illustrated in figure 50.

Image	Traditional LSB (db)	Enhanced LSB (db)	Difference (db)
Banana.jpg	19.54	9.51	10.03
Dancers.jpg	17.29	9.84	7.45
Graduation.jpg	51.38	45.07	6.31
Office.jpg	18.97	10.38	8.59
zhbackground.bmp	39.51	36.03	3.48

Table 11 The SPA - traditional vs enhanced LSB method

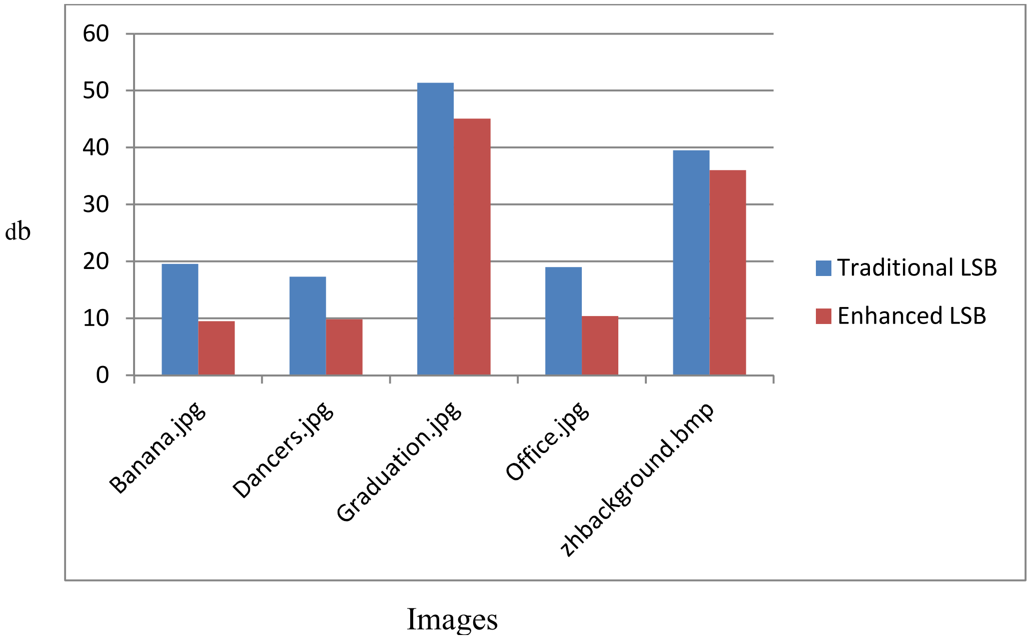
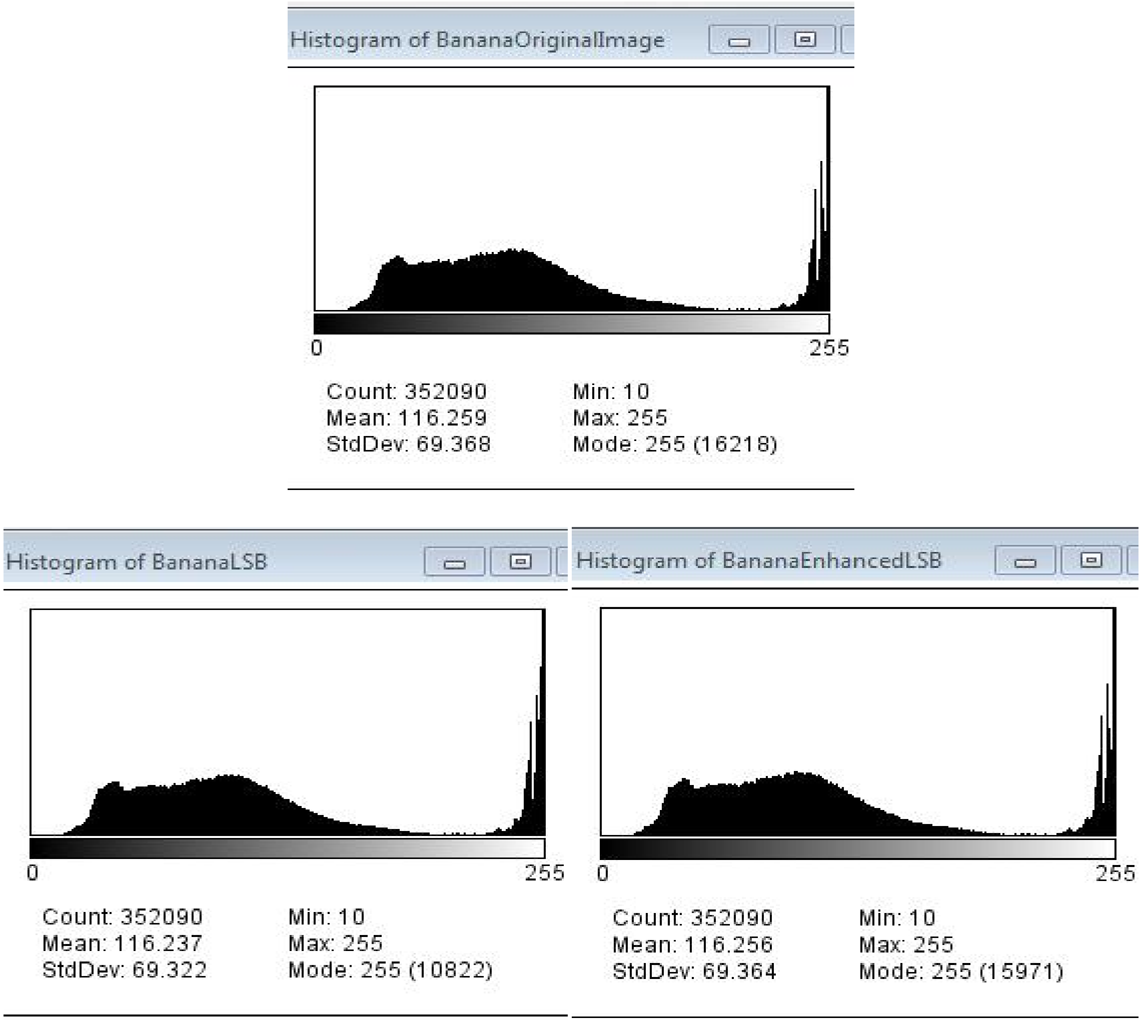


Figure 50 The SPA of stego images - traditional vs enhanced LSB

6.5.3 Histogram analysis

A histogram is an aggregation method that is used to illustrate data distribution in an image. The histogram is constructed by partitioning the data space into many small ranges, with each range corresponding to a bin. The height of an image histogram bin is then determined by the percentage of data points that fall in the corresponding range. This reveals the data density within each sub-range. (Jacobsen *et al.*, 2002)

Figure 51 below shows the comparison of histograms analysis for the image Banana.jpg. Compared to the original image, the standard deviation among pixels is least affected when the enhanced LSB method is used.



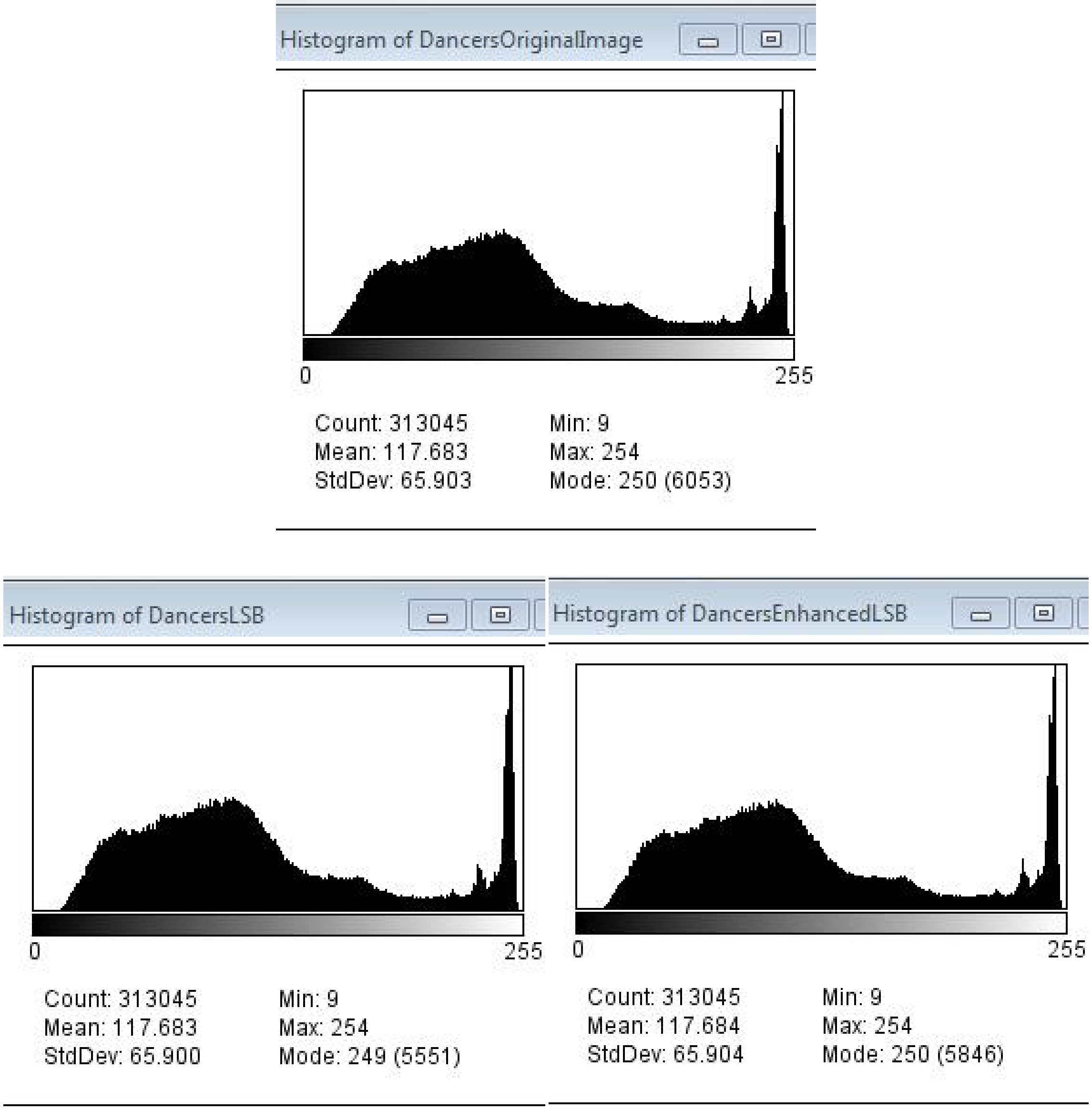
```
Histogram of BananaOriginalImage

Count: 352090        Min: 10
Mean: 116.259        Max: 255
StdDev: 69.368       Mode: 255 (16218)
```

```
Histogram of BananaLSB

Count: 352090        Min: 10
Mean: 116.237        Max: 255
StdDev: 69.322       Mode: 255 (10822)
```

```
Histogram of BananaEnhancedLSB

Count: 352090        Min: 10
Mean: 116.256        Max: 255
StdDev: 69.364       Mode: 255 (15971)
```

Figure 51 Banana.jpg histogram comparison

Figure 52 shows the comparison of histograms analysis for the image Dancers.jpg. Compared to the original image, the standard deviation among pixels is least affected when the enhanced LSB method is used.



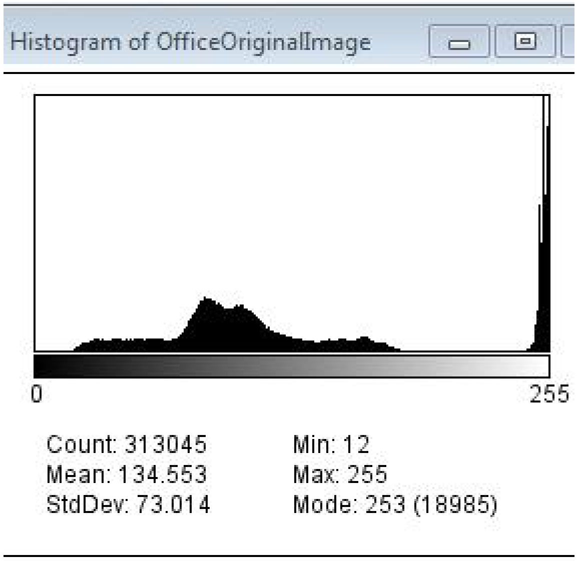
```
Histogram of DancersOriginalImage

Count: 313045     Min: 9
Mean: 117.683     Max: 254
StdDev: 65.903    Mode: 250 (6053)
```

```
Histogram of DancersLSB

Count: 313045     Min: 9
Mean: 117.683     Max: 254
StdDev: 65.900    Mode: 249 (5551)
```

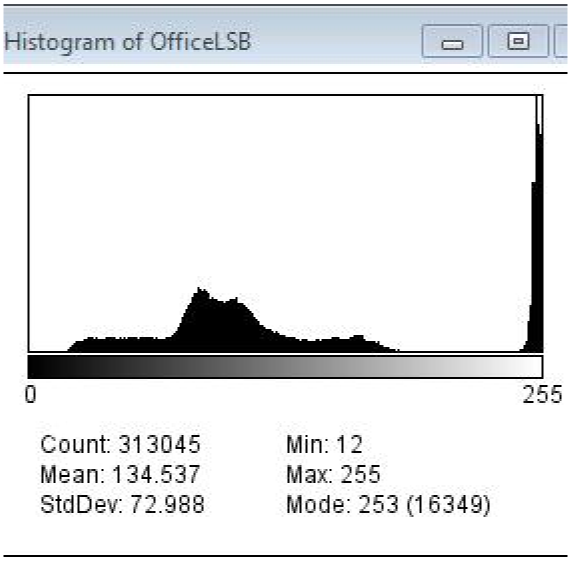
```
Histogram of DancersEnhancedLSB

Count: 313045     Min: 9
Mean: 117.684     Max: 254
StdDev: 65.904    Mode: 250 (5846)
```

Figure 52 Dancers.jpg histogram comparison

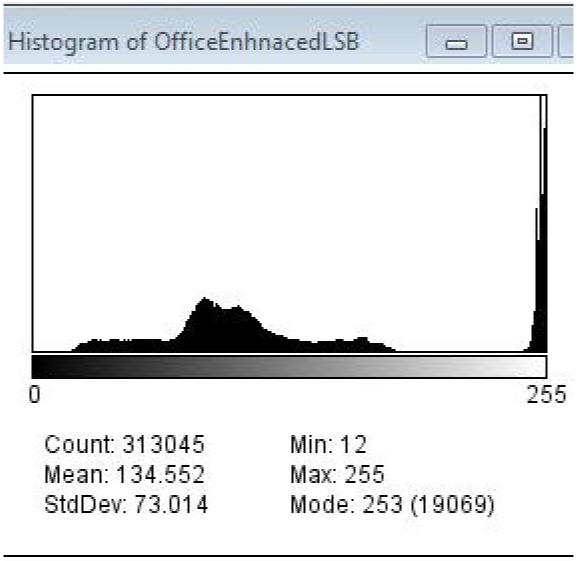
Figure 53 shows the comparison of histograms analysis for the image Office.jpg. Compared to the original image, the standard deviation among pixels is least affected when the enhanced LSB method is used.



```
Count: 313045        Min: 12
Mean: 134.553        Max: 255
StdDev: 73.014       Mode: 253 (18985)
```



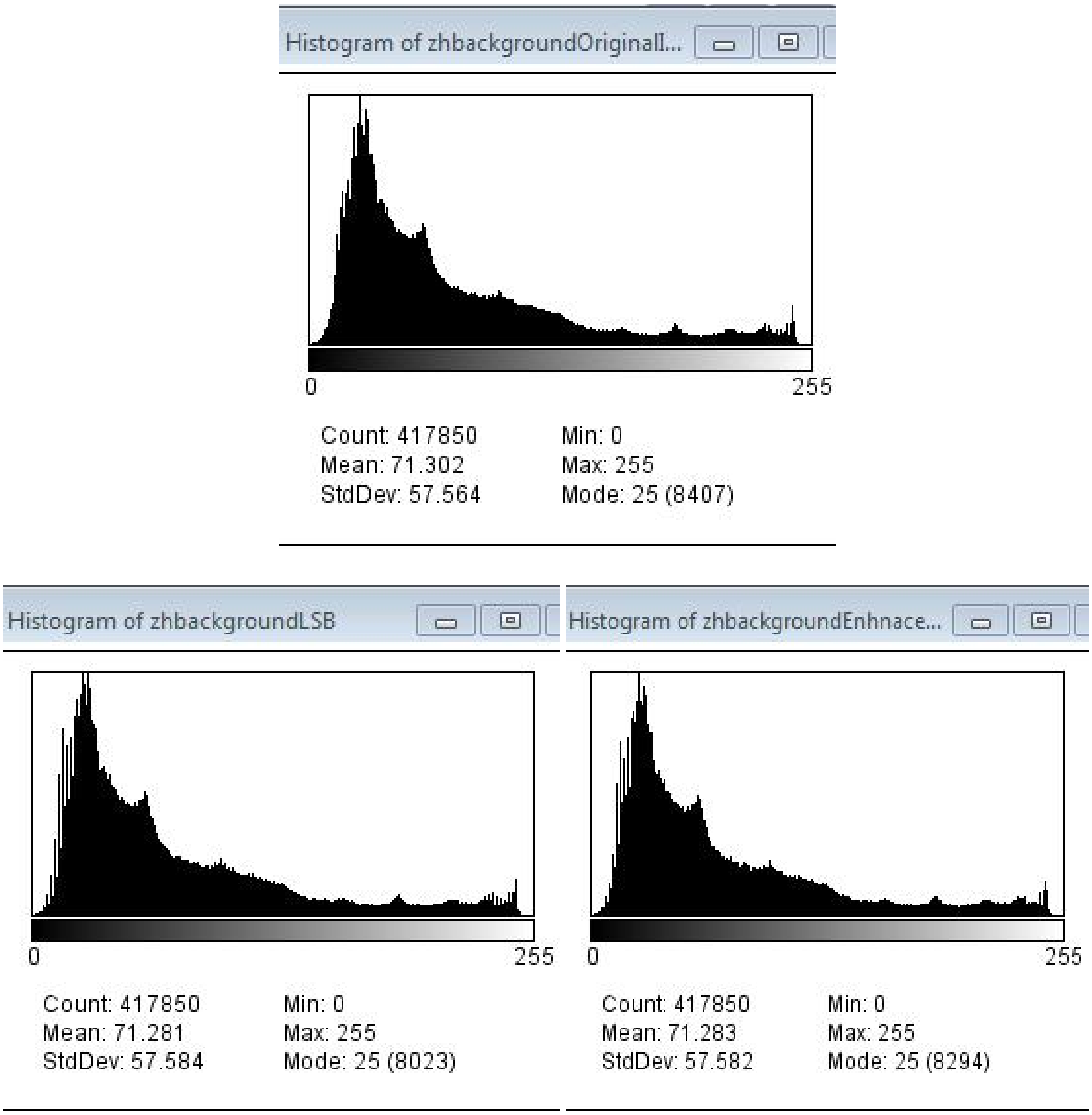
```
Count: 313045        Min: 12
Mean: 134.537        Max: 255
StdDev: 72.988       Mode: 253 (16349)
```



```
Count: 313045        Min: 12
Mean: 134.552        Max: 255
StdDev: 73.014       Mode: 253 (19069)
```

Figure 53 office.jpg histogram comparison

Figure 54 shows the comparison of histograms analysis for the image Zhbackground.jpg. Compared to the original image, the standard deviation among pixels is least affected when the enhanced LSB method is used.


Histogram of zhbackgroundOriginalI...
0
255
Histogram of zhbackgroundLSB
0
255
Histogram of zhbackgroundEnhnace...
0
255

```
Count: 417850        Min: 0
Mean: 71.302         Max: 255
StdDev: 57.564       Mode: 25 (8407)
```

```
Count: 417850        Min: 0
Mean: 71.281         Max: 255
StdDev: 57.584       Mode: 25 (8023)
```

```
Count: 417850        Min: 0
Mean: 71.283         Max: 255
StdDev: 57.582       Mode: 25 (8294)
```

Figure 54 zhbackground.bmp histogram comparison

As demonstrated by the above figures, the histograms generated by images created by the enhanced LSB method show a significant reduction in noise comparing the original and the stego images. This indicates less image distortion and hence improved imperceptibility.

Hiding capacity

In the traditional LSB method, the secret data is converted to a bit stream. Each bit of the message is then embedded into the LSB of the image's pixels. Therefore each pixel has a payload of 1 bit. The hiding capacity of a carrier file is defined as the maximum number of bits that can be hidden in it with an acceptable resultant stego-image quality.

The proposed enhanced LSB method presented in this research utilizes varied random bits per color channel. Each pixel in true color images has a total of 3 color channels. When more than one bit is use in each pixel color channel, more bits of secret data can be hid increasing the hiding capacity.

Table 12 is a summary of the perceptibility metrics for the enhanced LSB method in various images using the same payload (a 51 kilo byte Word document).

IMAGE	AAD (db)	MSE(db)	SNR(db)	PSNR (db)	RS(db)	SPA (db)
Banana.jpg	0.134	0.146	60.5	66.5	10.85	9.51
Dancers.jpg	0.1501	0.165	59.9	65.4	10.37	9.84
Graduation.jpg	0.6649	0.961	52.8	57.8	45.43	45.07
Office.jpg	0.1498	0.164	61.0	65.5	10.17	10.38
zhbackground.bmp	0.4992	0.664	50.6	59.4	36.37	36.03

Table 12 Summary of LSB perceptibility metrics

6.5.4 Subjective testing

For the subjective test, fifteen individuals (five digital photographers, five constant web users and five web developers) were used to identify the differences between original cover images and the stego images generated by the enhanced least significant bit method. In the first question, the respondents were exposed to both the original cover image and the stego images and asked to indicate whether the images were identical. The results of their responses are tabulated in table 13.

	Strongly Agree	**Agree**	**Disagree**	**Strongly Disagree**	**Do not Know**
Digital Photographers	3	2	0	0	0
Web Users	3	1	0	0	1
Web Developers	3	2	0	0	0

Table 13 Subjective tests for quality

In the first second question, the respondents were exposed to both the original cover image and the stego images and asked to indicate whether either of the images had clearly visible distortions. The results of their responses are tabulated in table 14.

Group	**Strongly Agree**	**Agree**	**Disagree**	**Strongly Disagree**	**Do not Know**
Digital Photographers	0	0	4	1	0
Web Users	0	0	3	2	1
Web Developers	0	0	3	1	1

Table 14 Subjective tests for image distortion

The general results of this survey indicates that the proposed enhanced least significant method improves levels of imperceptible to visual steganalysis ie the human visual system cannot detect the difference between the original image and the image used to hide the secret information.

An example using one of the images – Banana.jpg illustrates this in figure 55. The image is shown below in its original form without any secret information hidden in it.

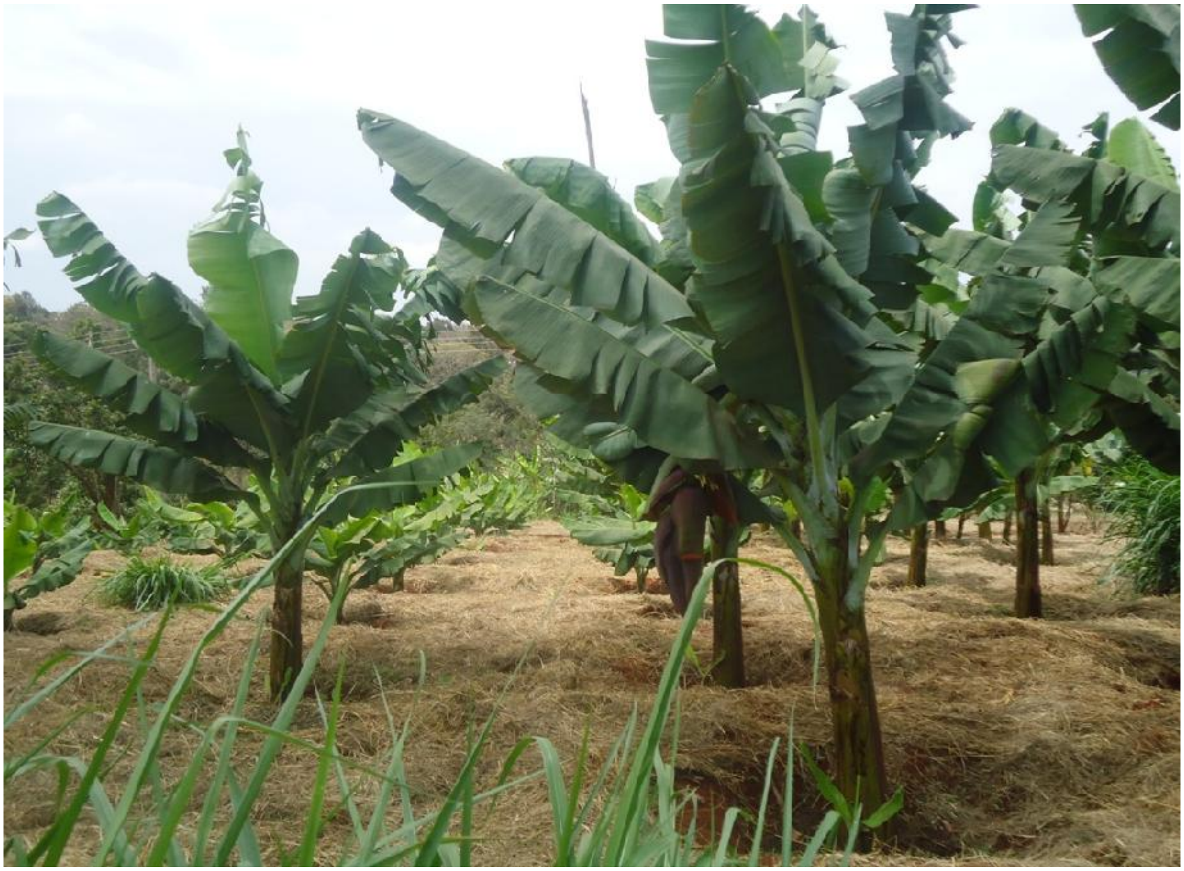


Figure 55 Banana.jpg without data

Figure 56 however shows the same image with secret information i.e a 51 kilo byte word document hidden in it using the enhanced LSB method.

Figure 56 Banana.jpg with hidden data

6.6 Comparative objective analysis between the Enhanced LSB Method and other LSB applications.

For a comparative analysis of objective metrics between the enhanced LSB Method and other steganographic applications discussed earlier, a single bitmap image (**zhbackground.bmp)** was used as a cover image. The same document (a 51 kb Word document) was embedded in the image using the different applications. The *Steganography Studio* version 1.0 (SS) and the *Digital Invisible Ink Toolkit* 1.5 (DIIT) steganalysis tools are then used to measure the image statistical characteristics. The results are as tabulated in table 15.

Applications	**AAD**	**MSE**	**SNR**	**PSNR**	**RS**	**SPA**
EyeMage	0.1155	0.3599	53.2	62.0	11.4644	10.9577
Invisible Secrets	1.5006	3.0025	44.0	52.8	103.8117	96.4961
The Third Eye	0.5727	0.8978	49.2	58.1	33.1363	33.9031
Hermetic Stego	1.8592	4.3366	42.4	51.2	86.59	77.02
Enhance LSB	**0.500**	**0.665**	**50.54**	**59.42**	**11.2639**	**10.8034**

Table 15 Enhanced LSB metrics vs other applications

6.7 Interpretation of the results

Average absolute difference (AAD)

The Enhanced LSB Method produced the least average absolute difference between the original image and the stego image indicating the least distortion. This is illustrated in figure 57.

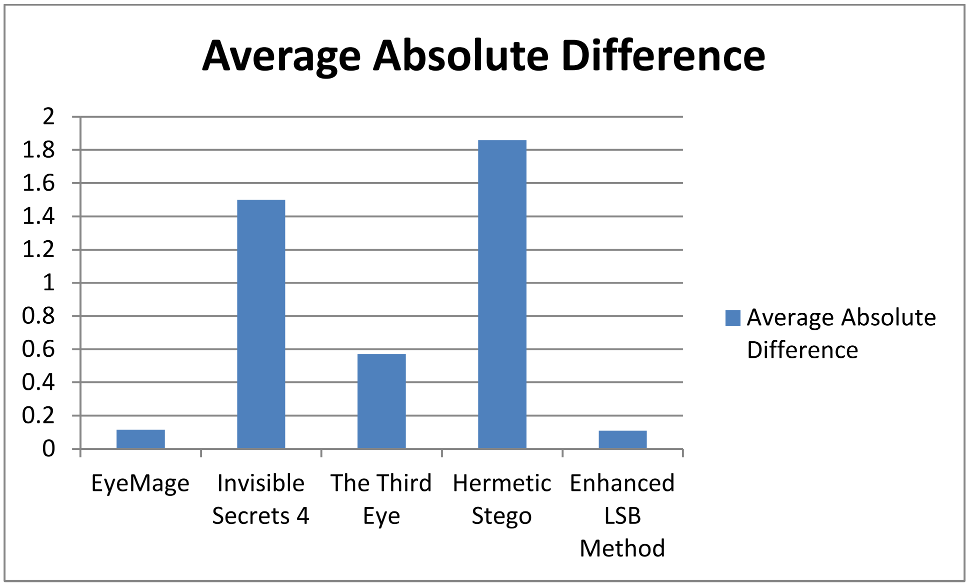


Figure 57 Comparison of AAD between enhanced LSB and other applications

Mean squared error

According the analysis above stego images generated by the enhanced LSB have the least MSE compared to those generated by the other applications and hence improved imperceptibility (Mei *et al.,* 2009). This is illustrated in figure 58.

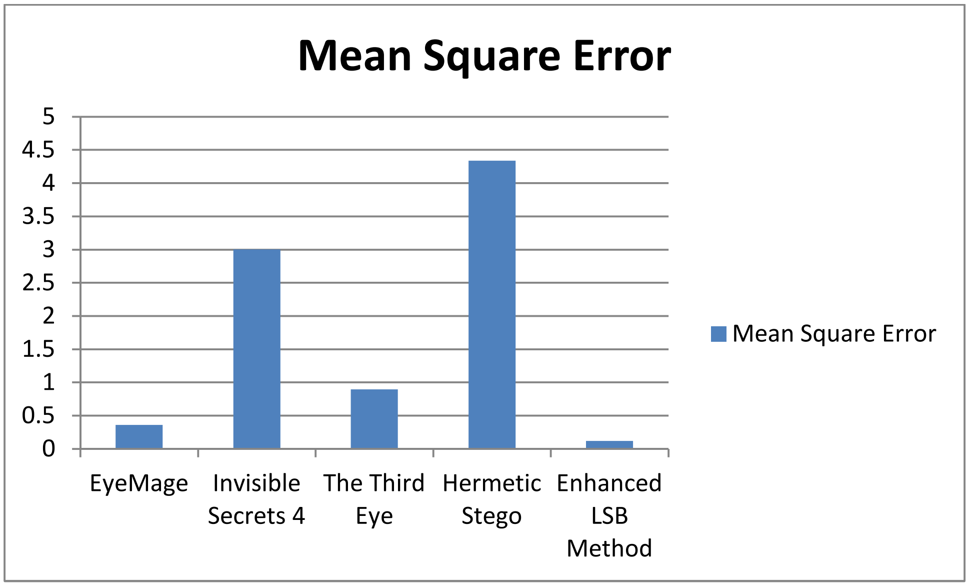


Figure 58 Comparison of MSE between enhanced LSB and other applications

Signal to noise ratio (SNR)

The Enhanced LSB Method recorded the highest SNR, displaying the best imperceptibility level among the other applications. Figure 59 illustrates the results of SNR testing. The higher the SRN of a stego image the better the hiding technique (Jiansheng *et al.,* 2009).

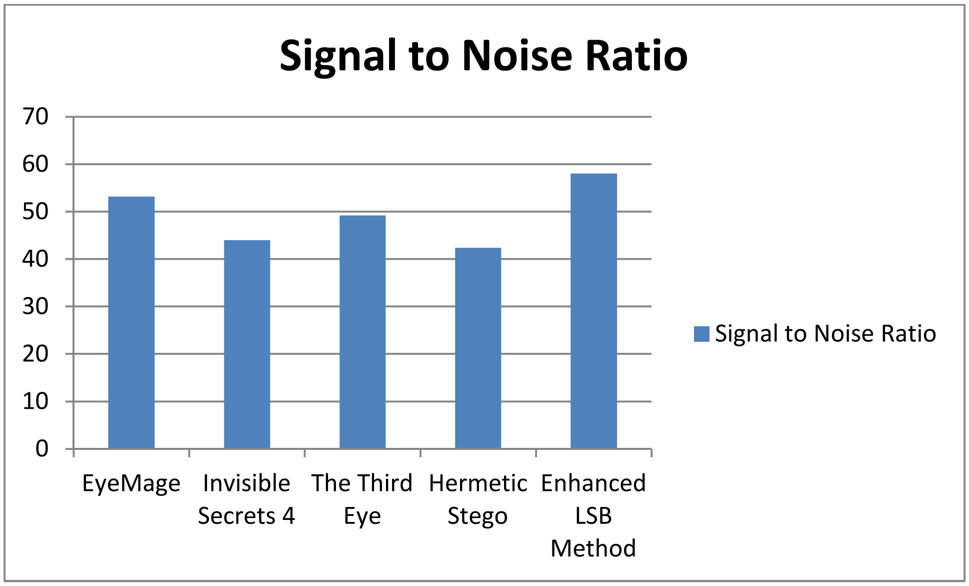


Figure 59 Comparison of SNR between enhanced LSB and other applications

Peak signal to noise ratio (PSNR)

The stego image registered a higher PSNR when the Enhanced LSB method was used compared to when the other applications were used. The analysis as illustrated in figure 60 shows that the enhanced LSB embedding method significantly improves on imperceptibility since a higher Peak Signal to Noise Ratio (PSNR) indicates improved imperceptibility (Jiansheng *et al.,*2009).

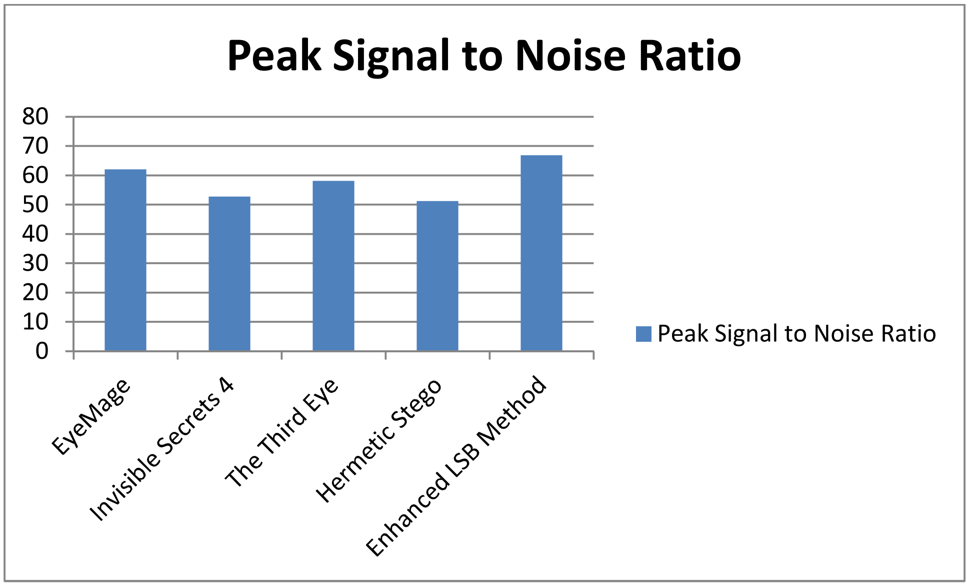


Figure 60 Comparison of PSNR between enhanced LSB and other applications

Reed-Solomon (RS) analysis

The Stego image generated by the Enhanced LSB Method recorded the lowest value of RS compared to those generated by the other applications. Therefore the stego image produced by the Enhanced LSB Method has lesser noise and the information hidden more imperceptible. This is as illustrated in figure 61.

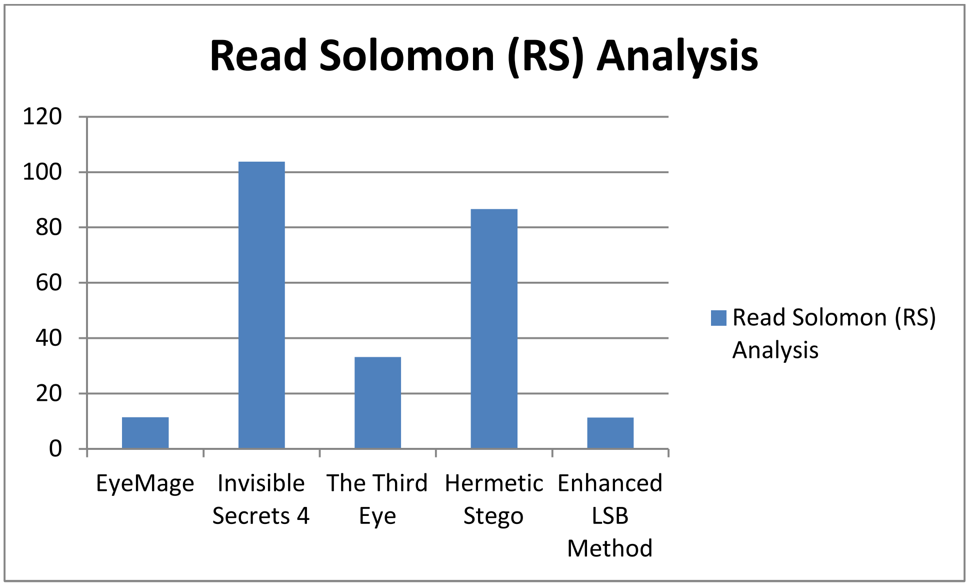


Figure 61 Comparison of RS between enhanced LSB and other applications

Sample pair analysis (SPA)

SPA is based on probabilities of transitions between sample pairs due to LSB embedding operations. The lower the values of SPA, the higher the imperceptibility level for the embedding algorithm. This experiment therefore reveals that the enhanced LSB method is superior to other applications tested.

Compared to the other already available steganographic applications as discussed above, Enhanced LSB Method has the least SPA value.

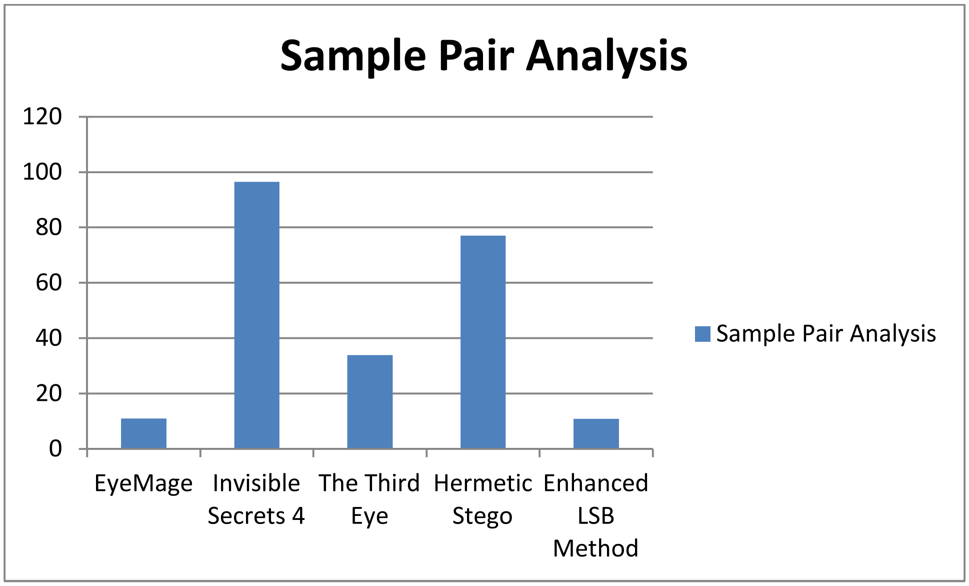


Figure 62 Comparison of SPA between enhanced LSB and other applications

6.7 Conclusion

The comprehensive and extensive experiments carried out during the testing of the proposed prototype showed improvement in the hiding capacity ranging between 1percent to 5 percent for the various metrics tested. The stego images produced by the proposed algorithm showed less levels of distortion compared to those produced by the traditional method and other related applications. The payload capacity also slightly increased without compromising on the quality of the stego image. This was clearly validated by the both the subjective and the objective experiments carried out on the stego images from both the algorithms.

CHAPTER SEVEN

7.0 CONCLUSION AND RECOMMENDATIONS

7.1 Introduction

This chapter summarizes findings and conclusions. It also pinpoints the main research contribution to knowledge in the study area. The limitations of the study and research are also outlined and directions for further research suggested.

7.2 Research findings

In this research an enhancement of the LSB steganographic information hiding method is presented in an attempt to provide better means of secure covert communication. Research in steganographic techniques is geared towards making sure that the method used provides for the maximum possible payload while ensuring that the hidden information is imperceptible. Though capacity, robustness and imperceptibility contend with each other, a good embedding method attempts to make the best trade-off between them. These most important components of any steganographic data hiding application were studied and analyzed and attempts made to improve on them in the traditional LSB method.

The proposed method utilizes specific and varied pixels, color channels and individual bits to hide secret data in a true color image. To identify the target image bits during both the embedding and the extraction processes, the LCG number generator was used. The research clearly established that such an embedding process improves on imperceptibility of the hidden information compared to the traditional embedding method. Use of varied bits in each color channel also improves on the payload within acceptable distortion levels. This was validated by both the subjective and the objective test experiments carried out on the stego images. In every case, lesser statistical differences were observed. Comparison of the stego images first order characteristics using histogram analysis also revealed lesser noise on images produced by the proposed method.

7.3 Research contributions

Digital image steganography is certainly gaining relevance by the day in various applications of data security. In covert communication, it is used to conceal the fact that communication is taking place between two parties. In medical imagery, clinical specialists can encode a patient's information in a carrier image to reduce possibility of wrong diagnosis and fraud. In copy write materials, digital signatures can be used to certify authenticity of documents and digital media.

Due to its simplicity, ease of application and less perceptual impact on cover images, the LSB method is one of the most popular algorithms in the world of digital steganography. An improvement on this method therefore makes it more robust and imperceptible thereby enhancing the security of the embedded data.

This research has presented an improvement on the imperceptibility and the hiding capacity of the traditional LSB Method. The visual and statistical analysis of the stego images produced by the proposed method registered improvement in each case compared to those produced by the LSB method and other applications that utilize the LSB method embedding process. By using varied and random image pixels, color channels and bits, information is not only well spread across the image, but the complexity of the embedding process is increased making decoding more difficult and thereby enhancing security.

By using varied number of bits per color channel, the prototype presented in this research improves on the hiding capacity or payload of the carrier image.

7.4 Recommendations and future work

In comparison to the traditional least significant bit algorithm, the enhanced least significant bit algorithm presented in this study was proved to demonstrate increased imperceptibility to both visual and statistical steganalysis attacks. However this method is recommended for communication purposes since it is designed for such applications only as more permanent aspects of steganography like watermarking are not included.

As is the practice with all image steganographic applications, the recommended carrier images are original photographs taken from high quality digital cameras. The recommended mode of transmission of the stego images is through web postings, email attachments, electronic bulletin boards and by file transfer such as file transfer protocol (FTP) (Cole, 2003). Whichever method of transmission is chosen, it should be one that prevents tracing by potential attackers.

The future work in relation to this research should centre on development of stronger embedding algorithms whose output can survive image manipulations and those that can make use of more permanent embedding procedures. This will facilitate the use of steganography in more sensitive application areas like computer digital forensics and enhance security in electronic commerce and trading applications. Research along these lines will also help in ensuring a permanent solution to the issues of plagiarism of copy write digital content materials.

7.5 Conclusion

Digital steganography is a rapidly growing and increasingly interesting field of research for information hiding and data security. It is currently playing a vitally important role in defense and civil applications. In future, we are bound to see more of data security applications based on this technology.

Though it cannot be said to replace cryptography, steganography has its place in data security and certainly it supplements cryptography. Its use for example in finger printing and watermarking for detection of plagiarism is continually being deployed and researched. Also, in countries and places where cryptography and data encryption is outlawed, steganography can still be used for secret data hiding and communication.

REFERENCES

Amin, M.M. (2003). Information hiding usingsteganography in Telecommunication Technology, . NCTT *Proceedings. 4th National Conference* pp. 21-25.

Anderson, R.J. and Petitcolas, F.A.P. (May 1998). On the limits of Steganography. *IEEE Journal of selected Areas in Communications.*

Andrew, D.K. (2004). Improved detection of LSB steganography in grayscale images, Proc. *of the 6th Information Hiding Workshop, Springer LNCS*, vol.3200, pp.97-115.

Artz, D. (2001) "Digital steganography: hiding data within data", *Internet Computing, IEEE*, vol. 5, Issue: 3, pp. 75-80

Bailey, K. and Curran, K. (2006) An Evaluation of Image Based Steganography Methods Using Visual Inspection and Automated Detection Techniques. *Multimedia Tools and Applications*, 31(2): 55-88.

Bender, W. (1996). 'Techniques for Data Hiding', *IBM Systems Journal*, 35(3&4), pp 313-336.

Cachin, C. (2008). "An Information-Theoretic Model for Steganography", *in proceeding 2nd Information Hiding Workshop*, vol. 1525, pp. 306-318.

Chan, and Cheng, L.M. (2004). Hiding data in images by simple LSB substitution. *Computer Journal of Pattern Recognition Letters*, vol. 37, no. 3, pp. 469-474

Chandramouli, R. and Memon, N. (2001) "*Analysis of LSB Based Image Steganography Techniques*", IEEE pp. Springer Verlag, 347-350

Chang, C.C., Chen, T.S. and Chung, L.-Z. (2002) A Steganographic Method Based Upon JPEG and Quantization Table Modification. *Information Sciences*, 141, 123-138.

Clair, B. (2001). Steganography: *How to Send a Secret Message*. Saint Louis University.

Cole, E. (2003) *Hiding in plain sight: Steganography and the art of Covert Communication,*

Indiana, John Wiley & Sons Inc.

Cox, I., (2009). *Information hiding, watermarking and steganography*. Public Lecture. Londonderry: University of Ulster at Magee Intelligent Systems Research Centre.

Cox, I. J., Miller, M. L., Bloom, J. A., Fridrich, J. & Kalker, T. (2008) *Digital Watermarking and Steganography-Second Edition*, Burlington, MA, USA, Elsevier Inc.

Currie, D.L. and Irvine, C.E. (1996). Surmounting the effects of lossy compression on Steganography. *19th National Information Systems Security Conference*. Londonderry.

Denton, J.A. (2002), "Biography of Jeremiah A. Denton" Accessed on 16th January 2012. Available in (http://www.nff.org/Jeremiah%20Denton.htm

Dumitrescu, S., Wu, X. and Wang, Z. (2003) Detection of LSB steganography via sample pair analysis, *IEEE Transactions on Signal Processing*, vol.51, no.7, pp.1995-2007.

Fridrich, J., Goljan, M. and Du, R. (2001). Reliable Detection of LSB Steganography in Color and Grayscale Images. *IEEE Multimedia*. 8, 22-28.

Fridrich, J., Goljan, M. & Hogea, D. (2002) Attacking the OutGuess. *The ACM Workshop on Multimedia and Security.*

Fridrich, J., Du R. and Long, M. (2000). '*Steganalysis of LSB Encoding in Color Images'*, Air Force Research Laboratory, Air Force Material Command, USAF.

Ghrare, S., Ali, M., Ismail M. and Jumari K. (2008). Diagnostic Quality of Compressed Medical Images: Objective and Subjective Evaluation, *IEEE Computer Society*.

Hinkelmann, K. and Kempthorne, O. (2008) Design *and Analysis of Experiments: Introduction to Experimental Design*, John Wiley & Sons, Inc., Hoboken,New Jersey.

Huaiqing,W.(2004). "*Cyber Warfare:* Steganography vs. Steganalysis," Vol 47, *Communications of the ACM*, October 2004.

Hull, T.E and A.R. Dobell, A.R. (1962) "Random Number Generators.", SIAM Review, vol.4, No.3,1962,pp.230-254.

Isbell, R.A. (2002). *Steganography: Hidden Menace or Hidden Savoir*. Steganography White Paper.

Ismail, A. (2003), "Steganalysis Using Image Quality Metrics," *IEEE transactions on Image processing,* vol. 12, no. 2.

Jacobsen, N., Solanki, K., Madhow, U., Manjunath B.S. and Chandrasekaran, S. (2002) "Image-Adaptive high –volume data hiding based on scalar quantization," *In proceedings of IEEE military communication conference (MILCOM), Anaheim CA, USA.*

Jessica Fridrich and Miroslav Goljan.(2002) Practical steganalysis of digital images-state of the art. In *Proceedings of SPIE*, pages 1–13.

Kiran, P.S. (2002). 'Project Third Eye', *Webkclub.com*. Accessed on 14th January 2012. Available on http://www.webkclub.com/tte/.

Knuth, D.E. (1997) " *The Art of Computer Programming*": Seminumerical. Algorithms (Vol 2, 3rd Ed, 1997), Addison-Wesley.

Krenn, J.R., (2004) "*Steganography and Steganalysis*". Accessed on 2nd March 2012. Available on http://www.krenn.nl/univ/cry/steg/article.pdf

Katzenbeisser, S and Petitcolas, F.A.P (2000) *Information Hiding Techniques for Steganography and Digital Watermarking* pp 80, Artech House

Kendall, K. (2008). 'Systems Analysis and Design'. Seventh Edition, Pretice Hall

Law, A. M. and Kelton, W.D. (2000). Simulation Modeling and Analysis, Third ed, New York: McGraw-Hill.

Lee, Y.K., Bell, G., Huang, S.Y., Wang, R.Z. and Shyu, S.J. (2009). An Advanced Least-Significant-Bit Embedding Scheme for Steganographic Encoding. Advances in Image and Video Technology. Berlin / Heidelberg: Springer, 349-360.

Lee, K. W. (2006). Attack for LSB Steganalysis of JPEG images. Digital Watermarking (5th International Workshop). London: Springer-Verlag, 35-48.

Lin E. and Delp E. (1999). '*A Review of Data Hiding in Digital Images'*, Video and Image Processing Laboratory, School of Electrical and Computer Engineering, Purdue University.

Lee, Y. K. and Chen L. H. (2000) "High Capacity Image Steganographic Model". *IEEE Proceedings Vision, Image and Signal Processing*, pp. 288-294.

Larry, L. C. and Lucy, A. D. L. (1999) Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design. Addison-Wesley Professional. ACM Press series

Marvel, L.M., Boncelet, C.G. and Retter, C.T (1999). 'Spread Spectrum Image Steganography', *IEEE Transactions on Image Proces*sing, 8(8), pp 1075-1083.

Mei, J., Li, S. and Tan, X. (2009) "A Digital Watermarking Algorithm Based on DCT and DWT", *in Proceedings of the 2009 International Symposium on Web Information Systems and Applications (WISA'09) Nanchang*, P. R. China, pp. 104-107.

Mohammad, F. and Abdallah, M. (2008). "*A Steganographic Data Security Algorithm with Reduced Steganalysis Threa*t," Birzeit University, Birzeit.

Morkel, T., Eloff, J.H.P. and Olivier, M.S. (2005). An Overview of Image Steganography. Delp, E. J. III and Wong, P.W. (Ed.). *Proceedings of the Fifth Annual Information Security South Africa Conference*. South Africa: Sandton.

Moulin, P. and Koetter, R. (2005). Data-Hiding Codes (tutorial paper), *Proceedings IEEE*, Vol. 93, No. 12, pp. 2083—2127.

Neil F. J. (1998), "*Steganalysis of Images Created Using Current Steganography Software*," Center for Secure Information Systems, George Mason University, Fairfax, Virginia.

NeoByte Solutions, Invisible Secrets 4.6.4 last update 2007-04-01. Accessed on 12th January 2012. Available on http://www.neobytesolutions.com/invsecr/.

Neil, F. J. and Jajodia, S. (1998). 'Exploring Steganography: Seeing the Unseen', *IEEE Computer*, 31(2), pp 26-34.

Park, S.K. and Miller, K.W.(1988). Random Number Generators: Good Ones are Hard to Find. *Commun. ACM* 31(11): 1192-1201

Pressman, R.S. (2008). Software Engineering: A Practitioner's Approach, 3rd ed, International ed, Singapore: McGraw-Hill.

Petitcolas, F.A.P.(2000) . *Information hiding techniques for steganography and digital watermarking*. Norwood:Artech House, INC.

Petitcolas, F.A., Anderson, R.J and Kuhn M.G. (1999). 'Information Hiding–A Survey', *Proceedings of the IEEE*, 87(7), pp 1062-1078.

Pfitzmann, B. (1996). 'Information Hiding Terminology', In:Information Hiding*: First International Workshop* (R Anderson, ed), *Lecture Notes in Computer Science* 1174, pp 347-350, Berlin: Springer-Verlag.

Provos, N. and Honeyman, P. (2003) Hide and Seek: An Introduction to Steganography. *IEEE Security & Privacy Magazine,* 1, 32-44.

Provos, N. and Honeyman, P.(2001). *Detecting steganographic content on the Internet*. Technical report, University of Michigan.

Popa, R.(2008) "*An Analysis of Steganographic System", The "Politehnica*" University of Timisoara, Faculty of Automatics and Computers, Department of Computer Science and Software Engineering.

Sellars D. (2006) "An Introduction to Steganography," http ://www. totse.com /en/privacy/ encryption/ 163947. Html

Shuttleworth, M. (2008). Experiment Resources. URL:http://www.experiment-resources.com. Accessed on 17th January 2012.

Sousa, G. (2004) "*Supporting Separation of Concerns in Requirements Artifacts*". First Brazilian Workshop on Aspect- Oriented Software Development (WASP'04), 2004, Brazil.

teganos GmbH (2002). 'Steganos Security Suite 4', *Steganos GmbH Home.* http://www.steganos.com/en/sss/index.htm. Accessed on 13th January 2012

Stoica, A., Vertan, C. and Fernandez-Maloigne, C. (2003) Objective and subjective color image quality evaluation for JPEG 2000 compressed images. *International Symposium on Signals, Circuits and Systems, SCS 2003*, 1,137-140.

Tomas K. and Pavel S. (2005) '*Utilization of MATLAB for picture quality evaluation'*, Institute of Radio Electronics, Brno University of Technology, Czech. Republic.

Tseng, H.-W. and Chang, C.C. (2004). Steganography Using JPEG-Compressed Images. *The Fourth International Conference on Computer and Information Technology, CIT '04,* 12-17.

Umamaheswari, M., Sivasubramanian, S. and Pandiarajan, S. (2010). Analysis of Different Steganographic Algorithms for Secured Data Hiding. *International Journal of Computer Science and Network Security*, VOL.10 No.8 154-160

Venkatraman, S., Abraham, A. and Paprzycki, M. (2004) Significance of Steganography on Data Security. *The International Conference on Information Technology: Coding and Computing. ITCC* 2, 347-351.

Wang, Z., Sheikh, H. R. and Bovik, A. C. (2003) Objective Video Quality Assessment. *The Handbook of Video Databases: Design and Applications*.CRC Press

Wang, Z., Bovik, A. C. and Lu, L. (2002a). Why is image quality assessment so difficult? *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '02)*, 4, 3313-3316.

Wang, Z., Sheikh, H. R. and Bovik, A. C. (2002b). No-reference perceptual quality assessment of JPEG compressed images. *Proceedings of the International Conference on Image Processing*, 1, 477-480.

Wang, H. and Wang, S. (2004) Cyber Warfare: Steganography vs. Steganalysis. *Communications of The ACM*, 47, 76-82.

Watters, P. A., Martin, F. and Stripf, S. H. (2005) Visual Steganalysis of LSBEncoded Natural Images. *The Third International Conference on Information Technology and Applications*, ICITA 1, 746-751.

Wu, D.C. and Tsai, W.H. (2003). A steganographic method for images by pixel value differencing. *Pattern Recognition Letters*. Vol. 24 (9-10), 1613-1626.

Wu, N.I. and Hwang, M.S. (2007). Data Hiding: Current Status and Key Issues. *International Journal of Network Security*. Vol.4 (1), 1–9.

Yu, Y.H., Chang, C.C. and Lin, I.C. (2007) A new steganographic method for color and grayscale image hiding. *Computer Vision and Image Understanding*, 107, 183-194.

APPENDICES

APPENDIX I

SOFTWARE AND HARDWARE RESOURCES USED FOR THE PROJECT

- Desktop PC Pentium IV Duo core, 2.1 GHz, 120GB, 2GB of RAM
- Windows 7 Home Edition Operating system
- JDK 6.25
- JCreator 4.0 LE 4.0
- ArgoUML 3.2
- Digital Invincible Ink Toolkit 1.5
- Steganography Studio 1.0
- Fast Picture Viewer 1.6.0
- Easy Model 2.0
- ImageJ