

DATA ENCRYPTION AND HIDING IN DIGITAL IMAGE THROUGH STEGANOGRAPHIC TECHNIQUE

**A Thesis Submitted
in Partial Fulfilment of the Requirements
for the Degree of**

MASTER OF TECHNOLOGY

in

Software Engineering

by

**Abhay Deep Singh
(11404491030)**

**Under the Supervision of
Mr. Akhilesh Yadav
Assistant Professor
B. B. D. U., Lucknow**

to the

School of Engineering

**BABU BANARASI DAS UNIVERSITY
LUCKNOW**

May, 2016

CERTIFICATE

It is certified that the work contained in this thesis entitled “**Data Encryption and Hiding in Digital Image through Steganographic technique**”, by **Abhay Deep Singh** (Roll No. 1140449001), for the award of **Master of Technology** from Babu Banarasi Das University has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature

Mr. Akhilesh Yadav

Assistant Professor

B.B.D.U. Lucknow (U.P.)

Signature

Dr. Reena Srivastava

Head of Dept. (CS)

B.B.D.U. Lucknow (U.P.)

Date:

ABSTRACT

With the widespread use of Internet and wireless networks, and the blooming growth in consumer electronic devices and advances in multimedia compression techniques, multimedia streams are easily acquired nowadays. In an attempt to ensure protection of the aforementioned multimedia contents and effective hiding of additional data into such digital content, several techniques emerged. Steganographic techniques are a very important part of the future of Internet security and privacy on open systems such as the Internet because important data can be hidden inside a cover medium so that only the parties intended to get the message knows that a secret message exists. A cover medium acts as a carrier to embed messages into. Many different medium have been employed to embed messages for example images, audio, and video as well as file structures. The resulting media after the text message has been hidden in cover medium is called stego object (Anderson and Petitcolas 1998). The mostly used medium include: text, video, audio and image. Despite availability of several steganographic techniques, they are prone to visual, structural and statistical attacks. In relation to text steganography, texts with hidden data are expected to have higher entropy than those without.

Most of the research has been done in the area of steganography which is the act of hiding valuable information into the transmission medium in such a way that it would not make attention that there are hidden information into transmitted media.

ACKNOWLEDGMENT

I would like to express my sincere gratitude to **Mr. Akhilesh Yadav (Assistant Professor, B.B.D.U., Lucknow)** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

I would like to express my gratitude towards my parents & member of BBD University for their kind co-operation and encouragement which helped me in completion of this project.

I would like to express my special gratitude and thanks to industry persons for giving me such attention and time.

My thanks and an appreciation also goes to my colleague in developing the project and people who have willingly helped me out with their abilities.

Abhay Deep Singh

Roll Number -1140449001

TABLE OF CONTENTS

	Page No.
Certificate	ii
Abstract	iii
Acknowledgement	iv
List of Figures	v
CHAPTER 1 : INTRODUCTION	1-15
1.1 Cryptography	1
1.2 Cipher Types	3
1.3 Security Attack	6
1.4 Steganography	7
1.5 Steganography and Cryptography	9
1.6 The Steganographic Framework	11
1.7 Steganography Application	14
1.8 Objectives of Steganography	14
1.9 Organization of the Thesis	15
CHAPTER 2 : LITERATURE SURVEY	16-27
2.1 Existing Steganographic Techniques	16
2.2 Existing Attacks	18
2.3 Characteristics feature of Data Hiding Techniques	23
2.4 Image Compression	23
2.5 Comparative chart	23
2.6 Summary	27
CHAPTER 3 : SYSTEM ARCHITECTURE IMPLENTATION	28-35
3.1 Introduction	28
3.2 Concept of Data Hiding	28
3.3 Pseudo-Random Encoding Technique	33
CHAPTER 4 : SYSTEM IMPLEMENTATION	36-48
4.1 System Architecture	36
4.2 Sender End	36
4.2.1 Encryption Algorithm	39

4.2.2 Embedding Process	39
4.2.3 Mutation Process	41
4.2.4 Pixel Selection Process	41
4.3 Receiver End	42
4.3.1 Pixel Selection Process to extract	45
4.3.2 Reverse Mutation Process	45
4.3.3 Decryption process	46
CHAPTER 5 : SYSTEM IMPLEMENTATION AND RESULTS	49-55
5.1 Sender End	49
5.2 Receiver End	50
5.3 Result	52
5.4 Images of Form	54
CHAPTER 6 : CONCLUSION AND FUTURE SCOPE	
6.1 Future Scope	56
REFERENCES	57-60
CURRICULUM VITAE	

LIST OF FIGURES

Figure 1.1 Interception Attack	6
Figure 1.2 Modification Attack	7
Figure 1.3 Fabrication Attack	7
Figure 1.4 Tradeoff between embedding capacity, undetectability and robustness in data hiding.	10
Figure 1.5 A generalized steganographic framework	11
Figure 2.1 Flipping of set cardinalities during embedding	20
Figure 3.1 LSB Encryption technique	31
Figure 3.2 LSB decryption Algorithm	31
Figure 4.1 Sender End Procedure	37
Figure 4.2 Encryption Process	38
Figure 4.3 Embedding Process	40
Figure 4.4 Mutation Process	41
Figure 4.5 Pixel Selection Process to embed	42
Figure 4.6 Receiver End Procedure	44
Figure 4.7 Pixel Selection Process to extract	45
Figure 4.8 Reverse Mutation Process	46
Figure 4.9 Decryption Process	47
Figure 5.1 Results on Field Image	52
Figure 5.2 Results on Water Filter Image	53
Figure 5.3 Main Form	54
Figure 5.4 Encryption Form	54
Figure 5.5 Decryption Form	55

CHAPTER 1

INTRODUCTION

1.1 Cryptography

The term Cryptography comes from the Greek word “*kryptos*”, which means hidden. The origin of cryptography is usually dated from about 2000 BC, The first known use of a modern cipher was by Julius Caesar (100 BC to 44 BC), who did not trust on his messengers when communicating with his officers. For this reason, he created a system in which each character in his messages was replaced by a character three positions ahead of it in the Roman alphabet.

Prior to the modern age the Cryptography was known as encryption, the conversion of information from a readable state to unreadable state. The creator of an encrypted message share the decoding technique only to the intended user needed to recover the original information, thereby prevent unwanted persons to decode information.

Cryptography is a method of encryption and decryption process and transmitting data in a particular way so that only those for whom they can process and read it. Some of techniques of Cryptography are microdots, merging words within images. However, in today's computer centric world, cryptography is most often associated with transforming ordinary text into encrypted text, then at the receiver end do reverse process of encryption (known as decryption), who works this are known as cryptographers.

1.1.1 Methods of Cryptography

There are basically two types of method for cryptography.

1. **Substitution** In substitution technique the letters of message are replaced by other letters/numbers/symbols. If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with cipher text bit patterns.
2. **Transposition** In Transposition method the elements of the message are rearranged. The position of letters of message is changed in this method.

1.1.2 Terms Used in Cryptography

- **Plaintext** The original intelligible message that user wants to send to intended recipient. This plaintext has meaningful information.
- **Ciphertext** The transformed message that cannot be understood by others without the knowledge of key and algorithm. This cipher text has meaningless information.
- **Cipher** Cipher is an algorithm that is used by sender and receiver for converting message from one form to other form by using key that is plaintext to ciphertext or ciphertext to plaintext.
- **Encipher** The process of converting plaintext to ciphertext using a cipher and a key
- **Decipher** The process of converting ciphertext back into plaintext using a cipher and a key
- **Key** Some critical information used by the cipher to transform the message from one form to other, which is only known by the sender & receiver to encrypt or decrypt information.

1.1.3 Number of keys

1. Symmetric Key
 2. Asymmetric Key
-
1. **Symmetric key (or) single key** Sender and receiver uses same key for encryption and decryption. e.g. Digital Encryption Standard (DES), Triple-DES (3DES), IDEA, and BLOWFISH.
 2. **Asymmetric key** Sender and receiver use different key. In other words sender use one key for encryption that is known as public key and receiver use another key for decryption that is only known by the receiver. This is also known as public-private key. For example

RSA Asymmetric algorithm Rivest-Shamir-Adleman is the most commonly used asymmetric algorithm (public key algorithm). It can be used both for encryption and for digital signatures. The security of RSA is generally considered equivalent to factoring, although this has not been proved. RSA computation occurs with integers modulo $n = p * q$, for two large secret primes p, q . To encrypt a message m , it is exponentiated with a small public exponent e . For decryption, the recipient of the ciphertext $c = m^e \pmod{n}$ computes the multiplicative reverse $d = e^{-1} \pmod{(p-1)*(q-1)}$ (we require that e is selected suitably for it to exist) and obtains $cd = m^{e*d} = m \pmod{n}$. The private key consists of n, p, q, e, d (where p and q can be omitted). The public key contains only n and e . The problem for the attacker is that computing the reverse d of e is assumed to be no easier than factorizing n . The key size should be greater than 1024 bits for a reasonable level of security. Keys of size, say, 2048 bits should allow security for decades. There are actually multiple incarnations of this algorithm; RC5 is one of the most common in use, and RC6 was a finalist algorithm for AES.

1.1.4 Cryptanalysis The study of principles and methods of converting an unintelligible message back into an intelligible message without the knowledge of the key. Also called code breaking

1.2 Cipher Types

- I. Monoalphabetic Cipher
- II. Polyalphabetic Cipher

1.2.1 Monoalphabetic Cipher Monoalphabetic cipher is a substitution cipher in which for a given key, the cipher alphabet for each plain alphabet is fixed throughout the encryption process. For example, if character 'A' is encrypted as 'C', for any number of occurrence of character in that plaintext, 'A' will always get encrypted to 'C'. For example Caesar Cipher.

Caesar Cipher It is a mono-alphabetic cipher where in each letter of the plaintext is substituted by another letter to form the ciphertext. It is a simplest form of substitution cipher scheme. This cryptosystem is generally referred to as the Shift Cipher. The concept is to replace each alphabet by another alphabet which is ‘shifted’ by some fixed number between 0 and 25.

For this type of scheme, both sender and receiver agree on a ‘secret shift number’ for shifting the alphabet. This number which is between 0 and 25 becomes the key of encryption.

1.2.2 Polyalphabetic Cipher Polyalphabetic Cipher is a substitution cipher in which the cipher alphabet for the plain alphabet may be different at different places during the encryption process. For examples, playfair and Vigenere Cipher.

Playfair Cipher In this scheme, pairs of letters are encrypted, instead of single letters as in the case of simple substitution cipher. In playfair cipher, initially a key table is created. The key table is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet is omitted from the table as we need only 25 alphabets instead of 26. If the plaintext contains J, then it is replaced by I.

The sender and the receiver decide on a particular key, say ‘tutorials’. In a key table, the first characters (going left to right) in the table is the phrase, excluding the duplicate letters. The rest of the table will be filled with the remaining letters of the alphabet, in natural order. The key table works out to be

T	U	O	R	I
A	L	S	B	C
D	E	F	G	H
K	M	N	P	Q
V	W	X	Y	Z

Vigenere Cipher This scheme of cipher uses a text string (say, a word) as a key, which is then used for doing a number of shifts on the plaintext.

For example, let's assume the key is 'first'. Each alphabet of the key is converted to its respective numeric value: In this case,

f -> 6, i -> 9, r -> 18, s -> 19, and t -> 20.

Thus, the key is: 6 9 18 19 20.

- The sender and the receiver decide on a key. Say 'first' is the key. Numeric representation of this key is '6 9 18 19 20'.
- The sender wants to encrypt the message, say 'attack from south east'. He will arrange plaintext and numeric key as follows:

a	t	t	a	c	k	f	r	o	m	s	o	u	t	h	e	a	s	t
6	9	18	19	20	6	9	18	19	20	6	9	18	19	20	6	9	18	19

- He now shifts each plaintext alphabet by the number written below it to create ciphertext as shown below:

a	t	t	a	c	k	f	r	o	m	s	o	u	t	h	e	a	s	t
6	9	18	19	20	6	9	18	19	20	6	9	18	19	20	6	9	18	19
G	C	L	T	W	Q	O	J	H	G	Y	X	M	M	B	K	J	K	M

- Here, each plaintext character has been shifted by a different amount – and that amount is determined by the key. The key must be less than or equal to the size of the message.
- For decryption, the receiver uses the same key and shifts received ciphertext in reverse order to obtain the plaintext.

G	C	L	T	W	Q	O	J	H	G	Y	X	M	M	B	K	J	K	M
6	9	18	19	20	6	9	18	19	20	6	9	18	19	20	6	9	18	19
a	t	t	a	c	k	f	r	o	m	s	o	u	t	h	e	a	s	t

Vigenere Cipher was designed by tweaking the standard Caesar cipher to reduce the effectiveness of cryptanalysis on the ciphertext and make a cryptosystem more robust. It is significantly more secure than a regular Caesar Cipher.

1.3 Security Attack

There are basically four general categories of attack.

1. Interruption
2. Interception
3. Modification
4. Fabrication

1. Interruption In this type of attack, an asset of the system is destroyed or becomes unavailable or unusable. This is an attack on availability For example destruction of piece of hardware, cutting of a communication line or Disabling of file management system.

2. Interception In this type of attack, an unauthorized party gains access to the asset. This is an attack on confidentiality. Unauthorized party could be a person, a program or a computer. For example wire tapping to capture data in the network, illicit copying of files

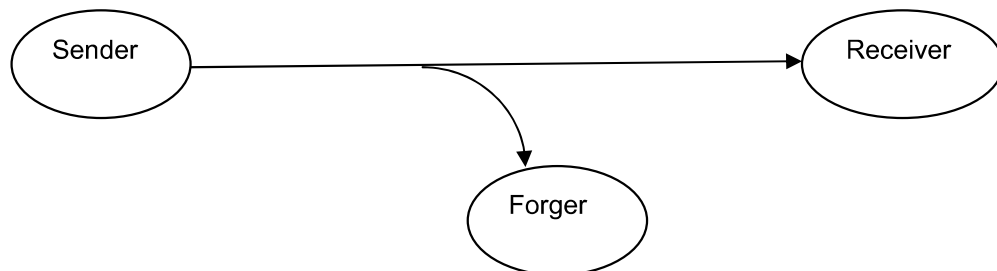


Fig. 1.1 Interception Attack

3. Modification In this type of attack, an unauthorized party not only gains access to the assets but tampers with it. This is an attack on integrity. For

example changing values in data file, altering a program, modifying the contents of messages that is being transmitted over the network.

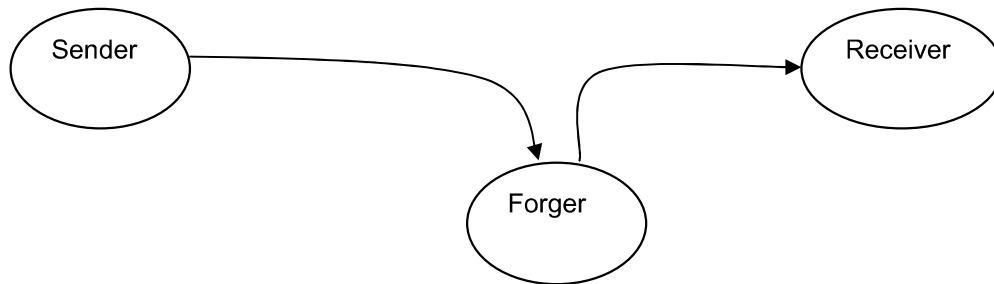


Fig. 1.2 Modification Attack

- 4. Fabrication** In this type of attack, an unauthorized party inserts counterfeit objects into the system. This is an attack on authenticity. For example insertion of spurious message in the network or addition of records to a file.

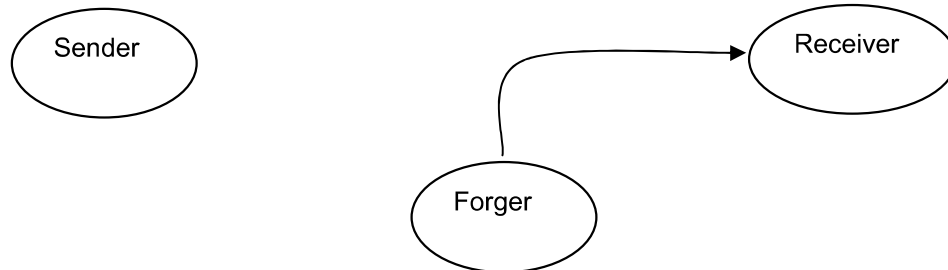


Fig. 1.3 Fabrication Attack

1.4 Steganography

The steganography is the art and science of secret communication. It is the art of hiding information or data imperceptibly in a cover medium. The word “Steganography” is derived from two Greek word “stegos” and “grafia”. “stegos” means ”covered” and “grafia” means “writing” collectively known as “covered writing” or “hidden writing”. In the image steganography the information is hidden exclusively in images. The main purpose of steganography is to hide the existence of the message in the cover medium. The Steganography includes a large array of methods of secret communication that concealed the very existence of hidden information and data. The Traditional techniques include use of invisible inks,

microdots etc. The original files can be referred to as cover image, cover text, or cover audio. After inserting the secret message it is referred to as stego-medium. A stego-key is used for the hiding/encoding process to restrict the detection or extraction of the embedded data. Modern day, steganographic techniques used to exploit the digital media images, audio files, video files etc. The Cryptography scrambles a message by using a certain cryptographic algorithms for converting the hidden data into unintelligible form. On the other hand, Steganography hides the message so that it cannot be seen. Steganography and cryptography are cousins in the spy craft family. The message in the ciphertext might arouse suspicion on the part of the recipient while an “invisible” message created with steganographic techniques will not. If anyone engaging in the secret communication can always apply a cryptographic algorithm to data before embedding it in, to achieve the additional security.

Another form of data hiding in digital images is Watermarking. Digital watermarking is the process of embedding auxiliary information into a digital cover signal with the aim of providing authentication information. Watermarking and fingerprinting related to steganography are basically used for intellectual property protection. A digital watermark is a kind of marker covertly embedded in a noise-tolerant signal such as audio or image data. It is typically used to identify ownership of the copyright of such signal. A watermark is called robust with respect to a class of transformations if the embedded information can reliably be detected from the marked signal even if degraded by any transformation within that class. Typical image degradations are JPEG compression, rotation, cropping, additive noise and quantization.

A Steganography and watermarking differ in a number of ways including purpose, specification and detection/extraction methods. The most fundamental difference is that the object of communication in watermarking is the host signal, with the embedded data providing copyright protection. In any case, once the presence of hidden information is revealed or even suspected, the purpose of steganography is defeated, even if the message content is not extracted or deciphered. According to [1], “Steganography’s niche in security is to supplement

cryptography, not replace it. If a hidden message is encrypted, it must also be decrypted if discovered, which provides another layer of protection.”

1.5 Steganography and Cryptography

Steganography differs from cryptography

- (i) **Steganography** Hide the messages inside the Cover medium, Many Carrier formats.
- (ii) **Steganalysis** Breaking of Steganography is known as Steganalysis.
- (iii) **Cryptography** Encrypt the message before sending to the destination, no need of carrier/cover medium.
- (iv) **Cryptanalysis** Breaking of cryptography is known as Cryptanalysis.

The Watermarking and fingerprinting related to steganography are basically used for intellectual property protection. The embedded information in a watermarked object is a signature refers the ownership of the data in order to ensure copyright protection. A digital watermark is a kind of marker covertly embedded in a noise-tolerant signal such as audio or image data. In this case, it becomes easy for the property owner to find out such customers who give themselves the right to violate their licensing agreement when they illegally transmit the property to other groups. It is typically used to identify ownership of the copyright of such signal. In fingerprinting, different and specific marks are embedded in the copies of the work that different customers are supposed to get.

In steganography the object to be transmitted is the embedded message, and the cover signal serves as an innocuous disguise chosen fairly arbitrarily by the user based on its technical suitability. The difference between Steganography and Watermarking with respect the three parameters of payload, un-detectability and robustness can be understood from Figure 1.4.

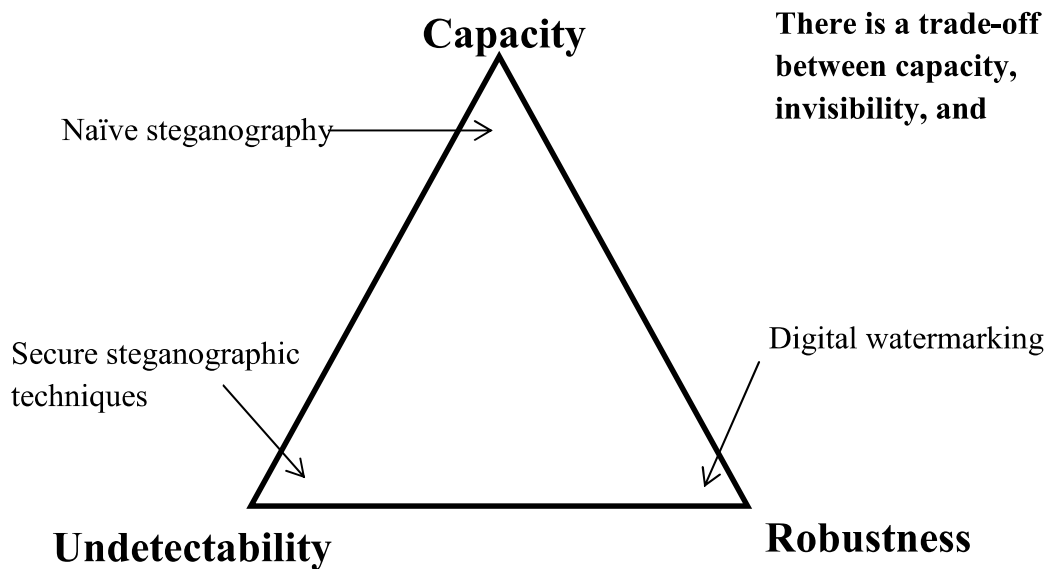


Fig. 1.4 Tradeoff between embedding capacity, undetectability and robustness in data hiding.

The Steganalysis is the art of science of detecting the hidden messages using Steganography. This is analogous to the cryptanalysis applied to cryptography. The goal of steganalysis is to identify suspected packages, determine whether or not they have a payload encoded into them, and if possible, recover that payload. Hence, the major challenges of effective steganography are

- 1. Size of Payload** Unlike watermarking, which needs to embed only a small amount of copyright information, steganography aims at hidden communication and therefore usually requires sufficient embedding capacity. Requirements for higher payload and secure communication are often contradictory. Depending on the specific application scenarios, a tradeoff has to be sought.
- 2. Security of Hidden Communication** In order to avoid raising the suspicions of eaves droppers, while evading the meticulous screening of algorithmic detection, the hidden contents must be invisible both perceptually and statistically.

1.6 The Steganographic Framework

For a steganographic algorithm having a stego-key, given any cover image the embedding process generates a stego image. Any steganographic system can be studied as shown in Figure 1.2. The extraction process takes the stego image and using the shared key applies the inverse algorithm to extract the hidden message. The stego object is then sent through the public channel. In a pure steganographic framework, the technique for embedding the message is unknown to Rahul and shared as a secret between Ramesh and Suresh. This system can be explained using the ‘prisoner’s problem’, where Ramesh and Suresh are two inmates who wish to communicate in order to hatch an escape plan. However the communication between them is examined by the warden Rahul. To send the secret message to Suresh, Ramesh embeds the secret message ‘m’ into the cover object ‘c’, to obtain the stego object ‘s’. In the private key steganography Ramesh and Suresh share a secret key which is used to embed the message. The secret key, for example, can be a password used to seed a pseudo-random number generator to select pixel locations in an image cover-object for embedding the secret message. In the public key steganography, Ramesh and Suresh have private-public key pairs and know each other’s public key. In this thesis we confine ourselves to private key steganography only.

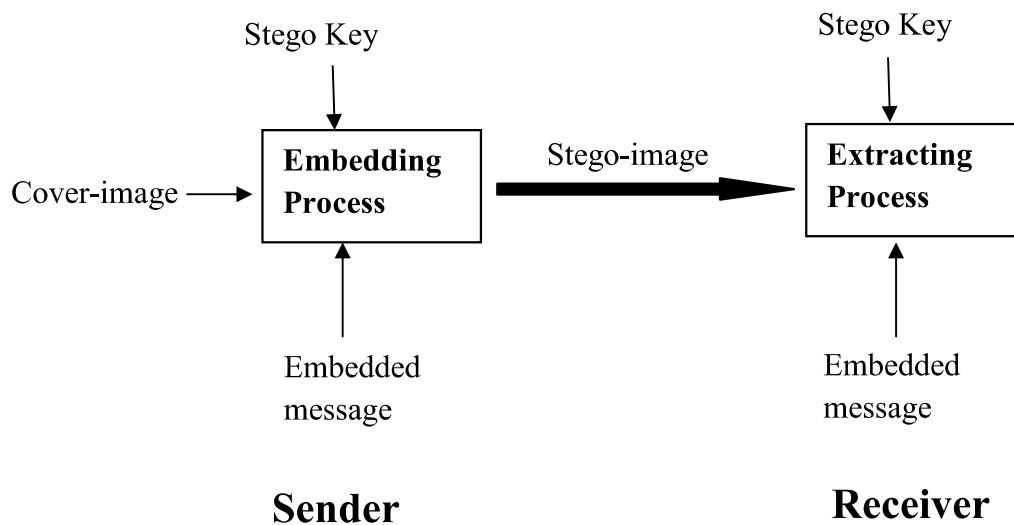


Fig. 1.5 A generalized steganographic framework

The term steganography came into use in 1500s after the appearance of Trithemius book on the subject Steganographia.[3]

1. 6.1 Past

The word Steganography technically means covered or hidden writing. Its ancient origins can be traced back to 440 BC. Although the term steganography was only coined at the end of the 15th century, the use of steganography dates back several millennia. In ancient times, messages were hidden on the back of wax writing tables, written on the stomachs of rabbits, or tattooed on the scalp of slaves. Invisible ink has been in use for centuries for fun by children and students and for serious undercover work by spies and terrorists [9].

1. 6.2 Present

The majority of today's steganographic systems uses multimedia objects like image, audio, video etc., as cover media because people often transmit digital pictures over email and other Internet communication. Modern steganography uses the opportunity of hiding information into digital multimedia files and also at the network packet level[4].

Hiding information into a medium requires following elements [2]

1. The cover medium (C) that will hold the secret message.
2. The secret message (M) may be plain text, digital image file or any type of data.
3. The steganographic techniques.
4. A stego-key (K) may be used to hide and unhide the message.

In modern approach, depending on the cover medium, steganography can be divided into five types:

1. Text Steganography
2. Image Steganography
3. Audio Steganography
4. Video Steganography
5. Protocol Steganography

- **Text steganography** Hiding information in text file is the most common method of steganography. The method was to hide a secret message into a text message. After coming of Internet and different type of digital file formats it has decreased in importance. Text stenography using digital files is not used very often because the text files have a very small amount of excess data.
- **Image steganography** Images are used as the popular cover medium for steganography. A message is embedded in a digital image using an embedding algorithm, using the secret key. The resulting stego-image is send to the receiver. On the other side, it is processed by the extraction algorithm using the same key. During the transmission of stego- image unauthenticated persons can only notice the transmission of an image but can't see the existence of the hidden message.
- **Video steganography** Video Steganography is a technique to hide any kind of files in any extension into a carrying Video file.
- **Audio steganography** Audio steganography is concerned with embedding information in an innocuous cover speech in a secure and robust manner. Communication and transmission security and robustness are essential for transmitting vital information to intended sources while denying access to unauthorized persons. An audible, sound can be inaudible in the presence of another louder audible sound .This property allows to select the channel in which to hide information[2]. Existing audio steganography software can embed messages in WAV and MP3 sound files.
- **Protocol steganography** The term protocol steganography is to embedding information within network protocols such as TCP/IP. We hide information in the header of a TCP/IP packet in some fields that can be either optional or are never used.

1.7 Steganography Application

- Secret Communications
- Feature Tagging Elements
- Copyright Protection

- (i) **Secret Communications[13]** The use of the steganography does not advertise the secret communication and therefore it avoids scrutiny of the sender, message, and recipient. The trade secret, the blueprint, or other sensitive information can be sent without alerting the potential attackers.
- (ii) **Feature Tagging Elements** It can be embedded inside the image, such as names of individuals in a photo or locations in the map. The Copying the stego-image also copies all of the embedded features and only parties who possess the decoding stego-key will be able to extract and view the features.
- (iii) **Copyright Protection** The Copy protection mechanisms that prevents the data, usually the digital data, from being copied. The insertion and analysis of watermarks image to protect copyrighted material is responsible for the recent rise of the interest in the digital steganography and in the data embedding.[16,17]

1.8 Objectives of Steganography

The project is carried out with the following objectives[2]

- a. The primary motivation of my current work is to increase PSNR(peak signal to noise ratio) of the stego image.
- b. To hide the message or a secret data into an image which acts as a cover medium, using LSB technique and pseudo random technique.

1.9 Organization of the Thesis

This thesis is organized as follows, In Chapter 2, “Literature Survey”, a background of the existing state of the steganographic research, is presented. The main categories of steganographic algorithms covered till date although the survey is not exhaustive and may have missed out some of the algorithms. In Chapter 3, the motivation for working on this approach in steganography and present some of the existing algorithms based on this approach, is presented. Chapter 4 gives a detailed analysis of the steps used to implement this research work. Chapter 5 contains implementation results followed by conclusion and future scope in Chapter 6.

CHAPTER 2

LITERATURE SURVEY

In this chapter, we are discussing the necessary background required for this thesis. In section 2.1, a brief overview of some of the existing steganographic techniques is provided. In the section 2.2, some of the common steganalytic attacks proposed till date as the counter measure to the steganographical attack, algorithms are described.

2.1 Existing Steganographic Techniques

A steganographic algorithms discussed in literature can be classified into following categories

1. Spatial Domain Techniques
2. Transform Domain Techniques
3. Masking and Filtering

Each of these techniques are discussed in detail in the next two subsections.

2.1.1 Spatial Domain

This technique uses the pixel gray levels and the color values directly for encoding the message bits. The major drawback of this method is amount of additive noise that creeps in the image which directly affects the “Peak Signal to the Noise Ratio” and the statistical properties of the image. These techniques are some of the simplest schemes in terms of embedding and extraction complexity. The most common algorithm belonging to this class of techniques is the “Least Significant Bit” (LSB) Replacement technique in which the least significant bit of the binary representation of the pixel levels is used to represent the message bit. Moreover these embedding algorithms are applicable mainly to lossless image compression schemes like TIFF images. For lossy compression schemes like JPEG, some of the message bits get lost during the compression step.

There are many versions of spatial steganography, all directly change some bits in the image pixel values in hiding data. Least significant bit (LSB)-based

steganography is one of the simplest techniques that hides a secret message in the LSBs of pixel values without perceptible distortions. To our human eye, changes in the value of the LSB are imperceptible. Embedding of message bits can be done either simply or randomly[4]. Least Significant Bit (LSB) replacement technique, Matrix embedding are some of the spatial domain techniques. This kind of embedding leads to an addition of a noise of $0.5p$ on average in the pixels of the image where p is the embedding rate in bits/pixel. This technique is popularly known as LSB Matching. It can be observed that even this kind of embedding adds a noise of $0.5p$ on average.

To further reduce the noise, [2] have suggested the use of a binary function of two cover pixels to embed the data bits. The embedding is performed using a pair of pixels as a unit, where the LSB of the first pixel carries one bit of information, and a function of the two pixel values carries another bit of information. It has been shown that embedding in this fashion reduces the embedding noise introduced in the cover signal.

Advantages of spatial domain LSB technique are

1. Degradation of the original image is not easy.
2. Hiding capacity is more i.e. more information can be stored in an image.

Disadvantages of LSB technique are

1. Robustness is low
2. Hidden data can be destroyed by simple attacks.

2.1.2 Transform Domain Technique

These techniques try to encode the message bits in the transform domain coefficients of the image. Data embedding performed in the transform domain is widely used for robust watermarking. Similar techniques can also realize large-capacity embedding for steganography. Candidate transforms include discrete cosine Transform (DCT), discrete wavelet transform (DWT), and discrete Fourier transform (DFT). In the Frequency domain, the message is inserted into transformed coefficients of image giving more information hiding capacity and more robustness

against attacks. Transform domain embedding can be termed as a domain of embedding techniques for which a number of algorithms have been suggested[3]. Most of the strong steganographic systems today operate within the transform domain. Transform domain techniques have an advantage over LSB techniques as they hide information in areas of the image that are less exposed to compression, cropping, and image processing. Some transform domain techniques do not seem dependent on the image format and they may out run lossless and lossy format conversions.

Transform domain techniques are of different types[3]

1. Discrete Fourier transformation technique (DFT).
2. Discrete cosine transformation technique (DCT).
3. Discrete Wavelet transformation technique (DWT).

2.1.3 Masking and Filtering

The Masking and Filtering is a steganography technique which can be used on the gray- scale images. The Masking and filtering is similar to placing watermarks on a printed image. Watermarking techniques can be applied without the fear of image destruction due to lossy compression as they are more integrated into the image [5]. These techniques embed the information in the more significant areas than just hiding it into the noise level.

Advantages of Masking and filtering Techniques

This method is much more robust than LSB replacement with respect to compression.

Disadvantages

Techniques can be applied only to gray scale images and restricted to 24 bits.

2.2 Existing Attacks

The steganalytic attacks proposed till date can be divided into visual and statistical attacks.

The statistical attacks can further be classified as

1. Targeted Attacks
2. Blind Attacks

Each of these classes of attacks is covered in detail in the next two subsections along with several examples of each category.

2.2.1 Targeted Attacks

These attacks are designed for keeping a particular steganographic algorithm in mind. A particular steganographic algorithm imposes a specific kind of the behaviour on the image features. These attacks are based on the image features which get modified by a particular kind of steganographic embedding. This specific kind of behaviour of the image statistics is exploited by the targeted attacks. Some of the targeted attacks are as follows

1. Histogram Analysis

In the histogram analysis method, it exploits the asymmetry introduced by LSB replacement. It has been observed statistically that in natural images (cover images), the number of odd pixels and the number of even pixels are not equal. For higher embedding rates of LSB Replacement these quantities tend to become equal. The main idea is to look for statistical artifacts of embedding in the histogram of a given image. So, based on this artifact a statistical attack based on the Chi-Square Hypothesis Testing is developed to probabilistically suggest one of the following two hypotheses.

- **Null Hypothesis H₀** The given image contains steganographic embedding
- **Alternative Hypothesis H₁** The given image does not contain steganographic embedding

The decision to accept or reject the Null Hypothesis H₀ is made on the basis of the observed confidence value p .

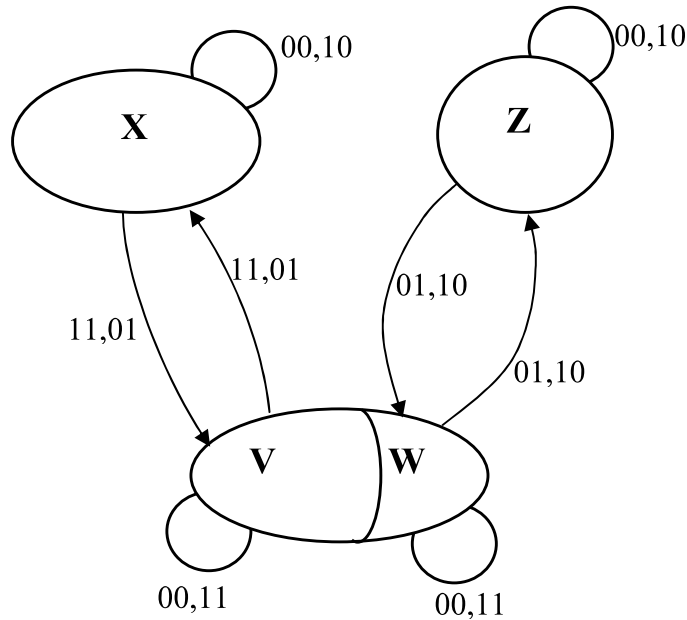


Fig. 2.1 Flipping of set cardinalities during embedding

2. **Sample Pair Analysis** A Sample Pair Analysis is another LSB steganalysis technique that can detect the existence of hidden messages that are randomly embedded in the least significant bits of natural continuous-tone images. The key to this methods success is the formation of 4 subsets of pixels (X, Y , U and V) whose cardinalities change with LSB embedding (as shown in Figure 2.1), and such changes can be precisely quantified under the assumption that the embedded bits are randomly scattered. It can precisely measure the length of the embedded message, even when the hidden message is very short relative to the image size. A detailed analysis on Sample Pair technique can be found in. Another attack called RS Steganalysis based on the same concept has been independently proposed.

3. **HCF-COM based Attack** HCF-COM based Attack was first proposed is based on the Center of Mass (COM) of the Histogram Characteristic Function (HCF) of an image. This attack was further expanded for LSB Matching. This attack observes the COM of a cover/stego-image ($C(H_c)/C(H_s)$) and its

calibrated version obtained by down sampling the image ($C(H_c)/C(H_s)$). It has been proved empirically that

$$C(H_c) \approx C(H_{\hat{c}}) \quad (2.1)$$

$$C(H_c) - C(H_s) > C(H_{\hat{c}}) - C(H_{\hat{s}}) \quad (2.2)$$

From Equations 2.1 and 2.2, a dimensionless discriminator for classification can be obtained as $C(H_s)/C(H_{\hat{s}})$. (2.3)

By estimating suitable threshold values of the discriminator from a set of training data, an image can be classified either as cover or stego.

2.2.2 Blind Attacks

The blind approach to steganalysis is similar to the pattern classification problem. The pattern classifier, in our case a Binary Classifier, is trained on a set of training data. Many of the blind steganalytic techniques often try to estimate the cover image statistics from stego image by trying to minimize the effect of embedding in the stego image. The training data comprises of some high order statistics of the transform domain of a set of cover and stego images and on the basis of this trained dataset the classifier is presented with images for classification as a non-embedded or an embedded image. This estimation is sometimes referred to as “Cover Image Prediction”. Some of the most popular blind attacks are defined next.

I. Wavelet Moment Analysis (WAM) Wavelet Moment Analyzer (WAM) is the most popular Blind Steganalyzer for Spatial Domain Embedding. WAM uses a denoising filter to remove Gaussian noise from images under the assumption that the stego image is an additive mixture of a non-stationary Gaussian signal (the cover image) and a stationary Gaussian signal with a known variance (the noise). As the Many of the blind steganalytic techniques often try to estimate the cover image statistics from stego image by trying to minimize the effect of embedding in the stego image.

WAM is based on a 27 dimension feature space. It then uses a Fisher Linear Discriminant (FLD) as a classifier. It must be noted that WAM is a state of the art

steganalyzer for Spatial Domain Embedding and no other blind attack has been reported which performs better than WAM.

II. Farid's Wavelet Based Attack It is based on the features drawn from the wavelet coefficients of an image. The second set of statistics is based on the errors in an optimal linear predictor of coefficient magnitude. It is from this error that additional statistics i.e. the mean, variance, skewness, and kurtosis are extracted thus forming a " $24 \times (n - 1)$ " dimensional feature vector. This attack first makes an n level wavelet decomposition of an image and computes four statistics namely Mean, Variance, Skewness and Kurtosis for each set of coefficients yielding a total of " $12 \times (n - 1)$ " coefficients. For implementation purposes, n is set to 4 i.e. four level decomposition on the image is performed for extraction of features. The source code of this attack is available. After extraction of features, a Support Vector Machine (SVM) is used for classification. We would like to mention that although a SVM has been used for classification we have used the Linear Discriminant Analysis for classification.

III. Calibration Based Attacks The calibration based attacks estimate the cover image statistics by nullifying the impact of embedding in the cover image. These attacks were first proposed by [14] and are designed for JPEG domain steganographic schemes. They estimate the cover image statistics by a process termed as Self Calibration. This calibration is done by decompressing the stego JPEG image to spatial domain and cropping 4 rows from the top and 4 columns from the left and recompressing the cropped image as shown in Figure 2.2. The cropping and subsequent recompressions produce a "calibrated" image with most macroscopic features similar to the original cover image. The steganalysis algorithms based on this self calibration process can detect the presence of steganographic noise with almost 100% accuracy even for very low embedding rates [14, 28].

2.3 Characteristics feature of Data Hiding Techniques

- **Robustness** to attacks can embedded data exist manipulation of the stego medium in an effort to destroy, or change the embedded data.
- **Perceptibility** does embedding message distort cover medium to a visually unacceptable level.
- **Tamper Resistance** Beyond robustness to destruction, tamper-resistance refers to the difficulty for an attacker to alter a message once it has been embedded in a stego-image.[13]
- **Capacity** how much information can be hidden with relative to the change in perceptibility.

2.4 Image Compression

In image compression there are two types of compression, that is lossy compression and lossless compression. In lossless compression, every single bit of data that was originally in the file remains after the file is uncompressed. In this case the resulting image is expected to be something similar to the original image, but not the same as the original. An example of an image format that uses this compression technique is JPEG (Joint Photographic Experts Group) [11]. All of the information is completely restored. The most popular image formats that use lossless compression is GIF (Graphical Interchange Format) and BMP (bitmap file). Lossy compression reduces a file by permanently eliminating certain information, especially redundant information. When the file is uncompressed, only a part of the original information is still there.

2.5 Comparative chart

Table 2.1 Comparative chart

Paper Name	Domain	Input	Method
J. K. Mandal, A. Khamrui “A Data Embedding Technique for Gray scale Image	Spatial	Text or Image	<ul style="list-style-type: none">• Large volume of message/ image is embedded in spatial domain using 3 x 3 masks from the source image.• Four bits of the secret message /

Using Genetic Algorithm (DEGGA)”			<p>image is embedded per byte of the source image onto the rightmost 4 bit of each pixel.</p> <ul style="list-style-type: none"> • Mutation is applied on the embedded image. Also, a method of bit handling is applied to keep the fidelity high. Reverse process is followed during decoding. Genetic algorithm is used to enhance a security level.
Mr. Vikas Tyagi “Data Hiding in Image using least significant bit with cryptography”	Spatial	Text	<ul style="list-style-type: none"> • Message character position in alphabet series are stored in an array and same for key. • Then encrypts message by adding message array value and key array value with 1. • Converts message array value in corresponding ASCII value. Each bit of message are stored in host image pixel’s last bit.
Rehana Begum R.D, Sharayu Pradeep “Best Approach for LSB Based Steganography Using Genetic Algorithm and Visual Cryptography for Secured Data Hiding and Transmission over	Spatial	Text	<ul style="list-style-type: none"> • First eight bit key value is obtained by XOR operation of key. • Then each message character ASCII value are XORED with key value. Encrypted message are stored in LSB of host image. • The pixel values of the stego-image are modified by the genetic algorithm. This module is used to

Networks ”			<p>change the pixel positions of the stego image, which is another protection lock for the secret message and image. Using Genetic Algorithm’s cross-over concept, the column pixel shuffling happen first and then the row pixel shuffling.</p> <ul style="list-style-type: none"> • This stego-image is input for visual cryptography which uses Color Image Visual Cryptography algorithm to divide the input image into two secret shares.
<p>Samir Kumar Bandyopadhyay, Tuhin Utsab Paul, Avishek Raychoudhury “GENETIC ALGORITHM BASED SUBSTITUTION TECHNIQUE OF IMAGE STEGANOGRAPHY”</p>	Spatial	Text	<ul style="list-style-type: none"> • Convert the Target Text Message to their 8 bit binary representation from their corresponding ASCII code. • Read the 1st pixel value of the Cover Image & 1st bit of the generated bit stream. • Use a intelligent genetic function $f(r, c)$, which outputs a integer value from 0 to 7. The function may return NULL, in that case consider the next pixel of the host image. • Change the 8 bit binary representation of the pixel value by updating (pos+1)th bit with the last read bit from the generated bit

			<p>stream.</p> <ul style="list-style-type: none"> • Store the new generated pixel value in the corresponding position.
<p>K. Jyothsana, V. Lokeswara Reddy “Dithering Technique for Digital Image Steganography”</p>	Spatial	Image	<ul style="list-style-type: none"> • Dithering technique is used, which is a process of creating factory of colors from RGB set of colors. • Rotation the secret image for ‘r’ number of times. • Convert the secret image to binary equivalent. • Get the RGB value in pixel of cover image. • Store secret image binary value as 3 bits in R, 3 bits in G and 2 bits in B of LSB.
<p>Hadhoud, M. M. Ismail, N. A. Shawkey, W. & Mohammed, A.Z. “Secure perceptual data hiding technique using information theory”</p>	Spatial	Text	<ul style="list-style-type: none"> • Technique is based on entropy calculation; Data embedding is done on the basis of entropy. • If the entropy is > 2 then it inserts ‘4’ bits into the ‘4’ LSBs, if not then the entropy of the ‘5’ MSBs is calculated. • If it is > 2 then it inserts ‘3’ bits into the ‘3’ LSBs, if not then it inserts ‘2’ bits into ‘2’ LSBs.

2.6 Summary

In this chapter, we have discussed some of the necessary background needed for the rest of the thesis. Some other concepts and definitions may be used from time to time and they shall be explained as and when it is needed.

CHAPTER 3

SYSTEM ARCHITECTURE IMPLEMENTATION

3.1 Introduction

The information hiding system has been developed for keeping important information and data confidentially. However, in this chapter, we will study an image file as a carrier to hide messages or information. Therefore, the carrier will be known as cover-image, while the stego-object known as stego-image. The implementation of the system will only focus on Least Significant Bit (LSB) as one of the steganography techniques as mentioned in below [14].

3.2 Concept of Data Hiding

The least significant bit (LSB) (in other words, the 8th bit) of some or all the bytes inside an image is changed to a bit of the secret message. Digital images are mainly of two types

- (i) 24 bit images and
- (ii) 8 bit images.

In 24 bit images we can embed three bits of information in each pixel, one in each LSB position of the three eight bit values. Increasing or decreasing the value by changing the LSB does not change the appearance of the image much so the resultant stego image looks almost same as the cover image. In 8 bit images, one bit of information can be hidden.

The Least Significant Bit (LSB) Replacement technique is that in which the least significant bit of the binary representation of pixel gray levels is used to represent the message bit. This kind of embedding leads to the addition of a noise of $0.5p$ on average in the pixels of the image where p is the embedded rate in bits/pixel. This kind of embedding also leads to an asymmetry and a grouping in the pixel gray values $(0,1);(2,3);.....(254,255)$. To overcome this undesirable asymmetry, the decision of changing the least significant bit is randomized i.e. if the message bit does not match the pixel bit, then pixel bit is either increased or decreased by 1. It

can be observed that even this kind of embedding adds a noise of 0:5p on average. To further reduce the noise, [2] have suggested the use of a binary function of two cover pixels to embed the data bits. This asymmetry is exploited in the attacks developed for this technique as explained further in section 2.2. This technique is popularly known as LSB Matching. The embedding is performed by using a pair of pixels as a unit, where the LSB of the first pixel carries one bit of message, and a function of the two pixel values carries another bit of message. It has been shown that embedding in this fashion reduces the embedding noise introduced in the covered signal.

In [4], a multiple base number system has been employed for embedding data bits. While embedding, the human vision sensitivity has been taken care of. A similar kind of algorithm based on human vision sensitivity has been proposed by [5] by the name of Pixel Value Differencing. The variance value for a block of pixels is used to compute the number base to be used for embedding. According to [20], “For a given medium, the steganographic algorithm which makes fewer embedding changes or adds less additive noise will be less detectable as compared to an algorithm which makes relatively more changes or adds higher additive noise.” This approach is based on adding more amount of data bits in the high variance regions of the image for example near “the edges” by considering the difference values of two neighboring pixels. This approach has been improved further by clubbing it with least significant bit embedding in [6]. Following the same line of thought Crandall [7] have introduced the use of an Error Control Coding technique called “Matrix Encoding”.

In Matrix Encoding, q message bits are embedded in a group of $(2^q - 1)$ cover pixels while adding a noise of $(1 - 2^{-q})$ per group on average. The maximum embedding capacity that can be achieved is $q/(2^q - 1)$. For example, 2 bits of secret message can be embedded in a group of 3 pixels while adding a noise of 0:75 per group on average.

If the LSB of the pixel value of cover image $C(i,j)$ is equal to the message bit m of secret message to be embedded, $C(i,j)$ remain unchanged; if not, set the LSB of $C(i, j)$ to m . The message embedding procedure is given below

$$S(i,j) = \{C(i,j) - 1\}, \text{ if LSB } (C(i,j)) = 1 \text{ and } m = 0$$

$$S(i,j) = C(i,j), \text{ if LSB } (C(i,j)) = m$$

$$S(i,j) = \{C(i,j) + 1\}, \text{ if LSB } (C(i,j)) = 0 \text{ and } m = 1$$

Where $\text{LSB } (C(i, j))$ stands for the LSB of cover image $C(i,j)$ and m is the next message bit to be embedded, $S(i,j)$ is the stego image. As we already know each pixel is made up of three bytes consisting of either a 1 or a 0.

For example, Let us suppose one can hide a message in three pixels of an image (24-bit colors). Suppose the original 3 pixels are:[16]

(11101010 11101000 11001011)

(01100110 11001010 11101000)

(11001001 00100101 11101001)

A steganographic program could hide the letter "J" which has a position 74 into ASCII character set and have a binary representation "01001010", by altering the channel bits of pixels.

(11101010 11101001 11001010)

(01100110 11001011 11101000)

(11001001 00100100 11101001)

In this case, only four bits needed to be changed to insert the character successfully.

The advantage of LSB embed technique is its simplicity and many techniques uses these methods [10]. LSB embedding also allows high perceptual transparency. The figure 3.1 and 3.2 show the LSB encryption and decryption process flow diagrams

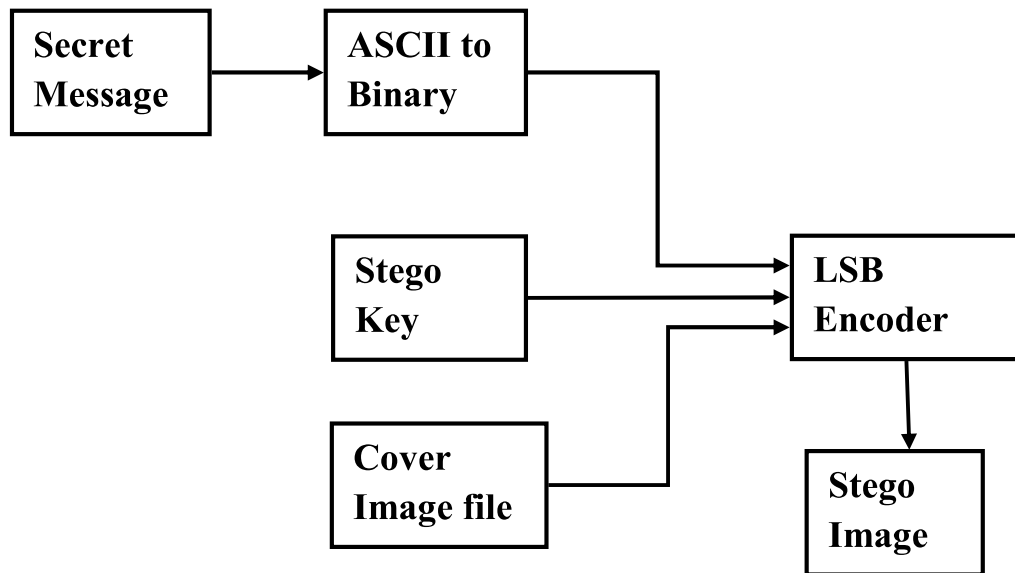


Fig. 3.1 LSB Encryption technique

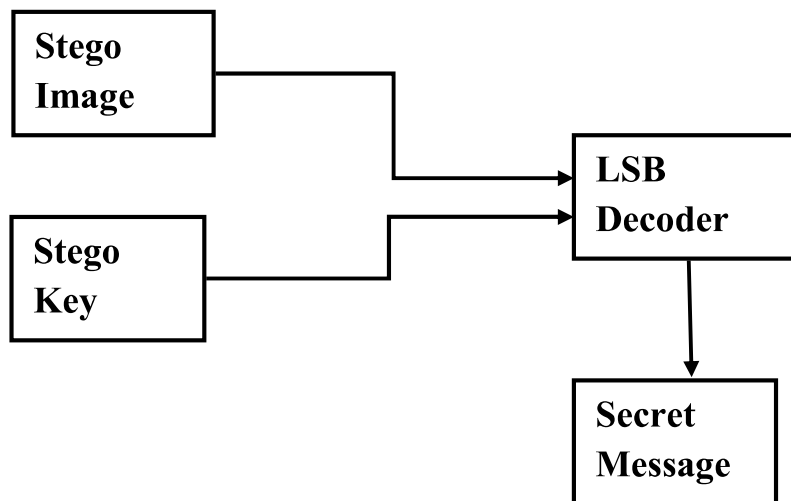


Fig. 3.2 LSB Decryption Technique

3.2.1 Data Embedding

The embedding process is as follows.

Inputs Cover image, stego-key and the text file

Output stego image

Procedure

Step 1 Extract the pixels of the cover image.

Step 2 Extract the characters of the text file.

Step 3 Extract the characters from the Stego key.

Step 4 Choose first pixel and pick characters of the Stego key and place it in first component of pixel.

Step 5 Place some terminating symbol to indicate end of the key. 0 has been used as a terminating symbol in this algorithm.

Step 6 Insert characters of text file in each first component of next pixels by replacing it.

Step 7 Repeat step 6 till all the characters has been embedded.

Step 8 Again place some terminating symbol to indicate end of data.

Step 9 Obtained stego image.[17]

3.2.2 Image Encoding Algorithm

Inputs Image file, stego key and image file

Output Stego image

Procedure

Step 1 The cover and secret images are read and converted into the unit8 type.

Step 2 The numbers in secret image matrix are conveyed to 8-bit binary. Then the matrix is reshaped to a new matrix a.

Step 3 The matrix of the cover image is also reshaped to matrix b

Step 4 Perform the LSB technique described above

Step 5 The stego-image, which is very similar to the original cover image, is achieved by reshaping matrix b.

Step 6 While extracting the data, the LSB of the stego image is collected and they are reconstructed into the decimal numbers. The decimal numbers are reshaped to the secret image[10].

3.2.3 Data Extraction

The extraction process is as follows.

Inputs Stego-image file, stego-key

Output Secret text message

Procedure

Step 1 Extract the pixels of the stego image.

Step 2 Now, start from first pixel and extract stego key characters from first component of the pixels.

Step 3 To terminating symbol, otherwise follow step 4.

Step 4 If this extracted key matches with the key entered by the receiver, then follow Step 5, otherwise terminate the program.

Step 5 If the key is correct, then go to next pixels and extract secret message characters from first component of next pixels. Follow Step 5 till up to terminating symbol, otherwise follow step 6.

Step 6 Extract secret message[18,20]

3.3 Pseudo-Random Encoding Technique

In this technique, a random key is used to choose the pixels randomly and embed the information. This will make the message bits more difficult to find and hopefully reduce the realization of patterns in the image [9]. Data can be hidden in the LSB of a particular colour plane (Red plane) of the randomly selected pixel in the RGB colour space[19].

3.3.1 Extraction of Hidden Message

In this process of extraction, the process first takes the key and then random-key. These keys takes out the points of the LSB where the secret message is randomly distributed [18]. In decoding algorithm the random-key must match i.e. the random-key which was used in encoding should match because the random key sets the hiding points of the message in case of encoding. Then receiver can extract the embedded messages exactly using only the stego-key. Decoding process searches the

hidden bits of a secret message into the least significant bit of the pixels within a cover image using the random key.

3.3.2 Embedding Algorithm

In this process of encoding method, a random key is used to randomised the cover image and then hide the bits of a secret message into the least significant bit of the pixels within a cover image. The transmitting and receiving end share the stego key and random-key. The random-key is usually used to seed a pseudo-random number generator to select pixel locations in an image for embedding the secret message[3].

Inputs Cover image, stego-key and the message

Output Stego image

Procedure

Step 1 Read character from text file that is to be hidden and convert the ASCII value of the character into equivalent binary value into an 8 bit integer array.

Step 2 Read the RGB colour image(cover image) into which the message is to be embedded.

Step 3 Read the last bit of red pixel.

Step 4 Initialize the random key and randomly permute the pixels of cover image and reshape into a matrix.

Step 5 Initialize the stego-key and XOR with text file to be hidden and give message.

Step 6 Insert the bits of the secret message to the LSB of the Red plane's pixels.

Step 7 Write the above pixel to Stego Image File [19].

3.3.3 Message extraction algorithm

Inputs Stego-image file, stego-key, random key

Output Secret message

Procedure

Step 1 Open the Stego image file in read mode and from the Image file, read the RGB colour of each pixel.

Step 2 Extract the red component of the host image.

Step 3 Read the last bit of each pixel.

Step 4 Initialize the random-key that gives the position of the message bits in the red pixel that are embedded randomly.

Step 5 For decoding, select the pixels and Extract the LSB value of red pixels.

Step 6 Read each of pixels then content of the array converts into decimal value that is actually ASCII value of hidden character.

Step 7 ASCII values got from above is XOR with stego-key and gives message file, which we hide inside the cover image[19].

CHAPTER 4

SYSTEM IMPLEMENTATION

In the previous chapter a discussion on two prominent techniques of data encryption i.e. the LSB technique and the pseudo random technique was presented. The current chapter now discusses the implementation work which is the basis of this research work.

The steganography technique in this research work is applied to embed the secret encrypted data into the cover image and mutation process to enhance the security. The system implementation flow diagram is shown in 4.1.

4.1 System Architecture

The system architecture consists of four main stages viz. encryption, embedding, extraction and decryption. The encryption and embedding procedure is performed at the sender end and extraction and decryption process at receiver end.

4.2 Sender End

At the sender or transmitter end the message to be send and the secret key is generated and then forwarded to the encryption block. The encryption block generates the encrypted message by combining the message and the key using the encryption algorithm, the flowchart of which is shown in figure 4.2 and is described later in the chapter. This encrypted message is then combined with the cover image and sent to the embedding engine to embed it along with the encrypted image. The procedure for embedding uses a mutation process which is discussed later in the chapter. The result of the embedding process is a stego image which appears similar to the original image and contains the hidden encrypted message, in its pixel value information. The whole process is shown in the below flowchart in Figure 4.1. This completes the operation at the sender end and the stego image is then transmitted.

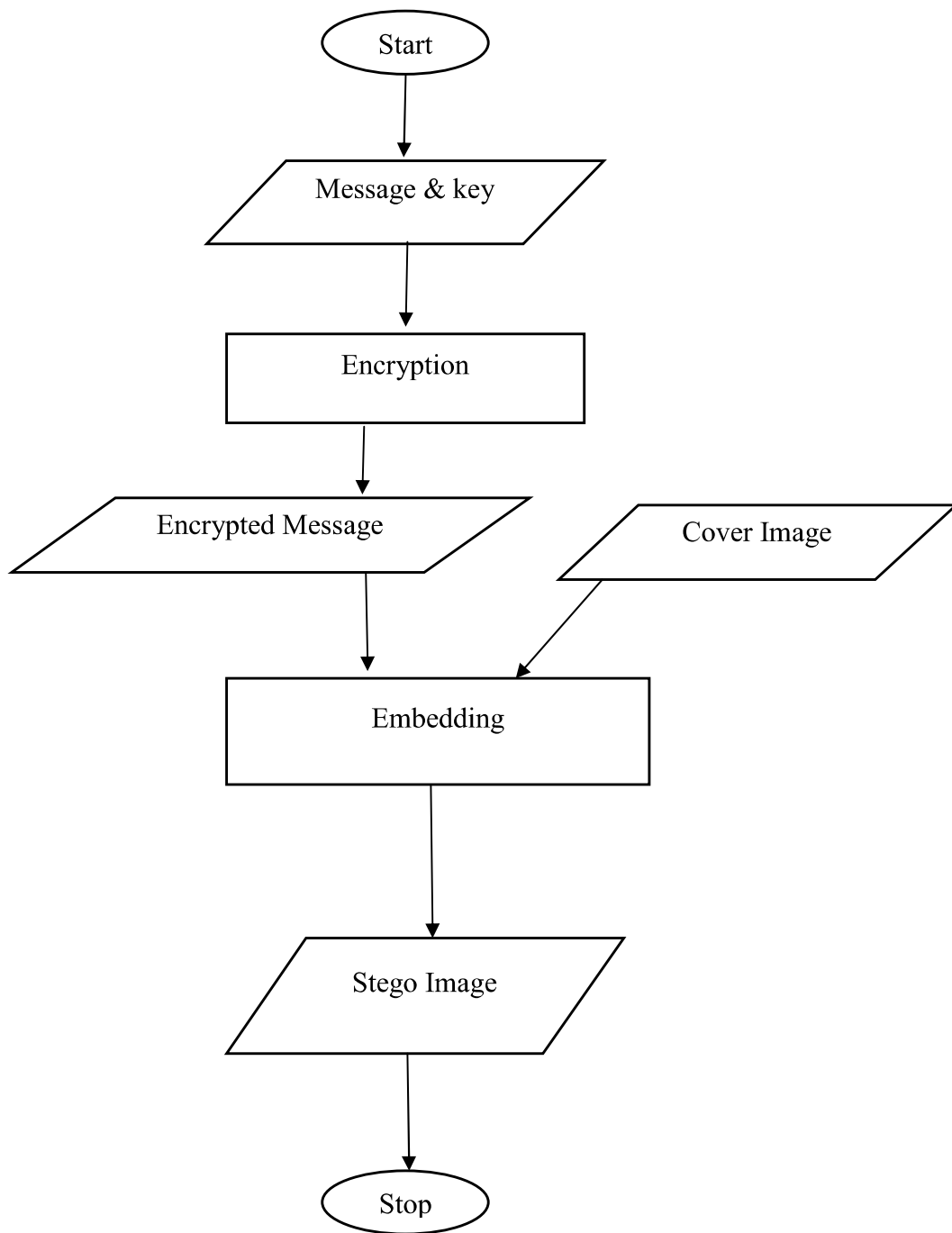


Fig. 4.1 Sender End Procedure

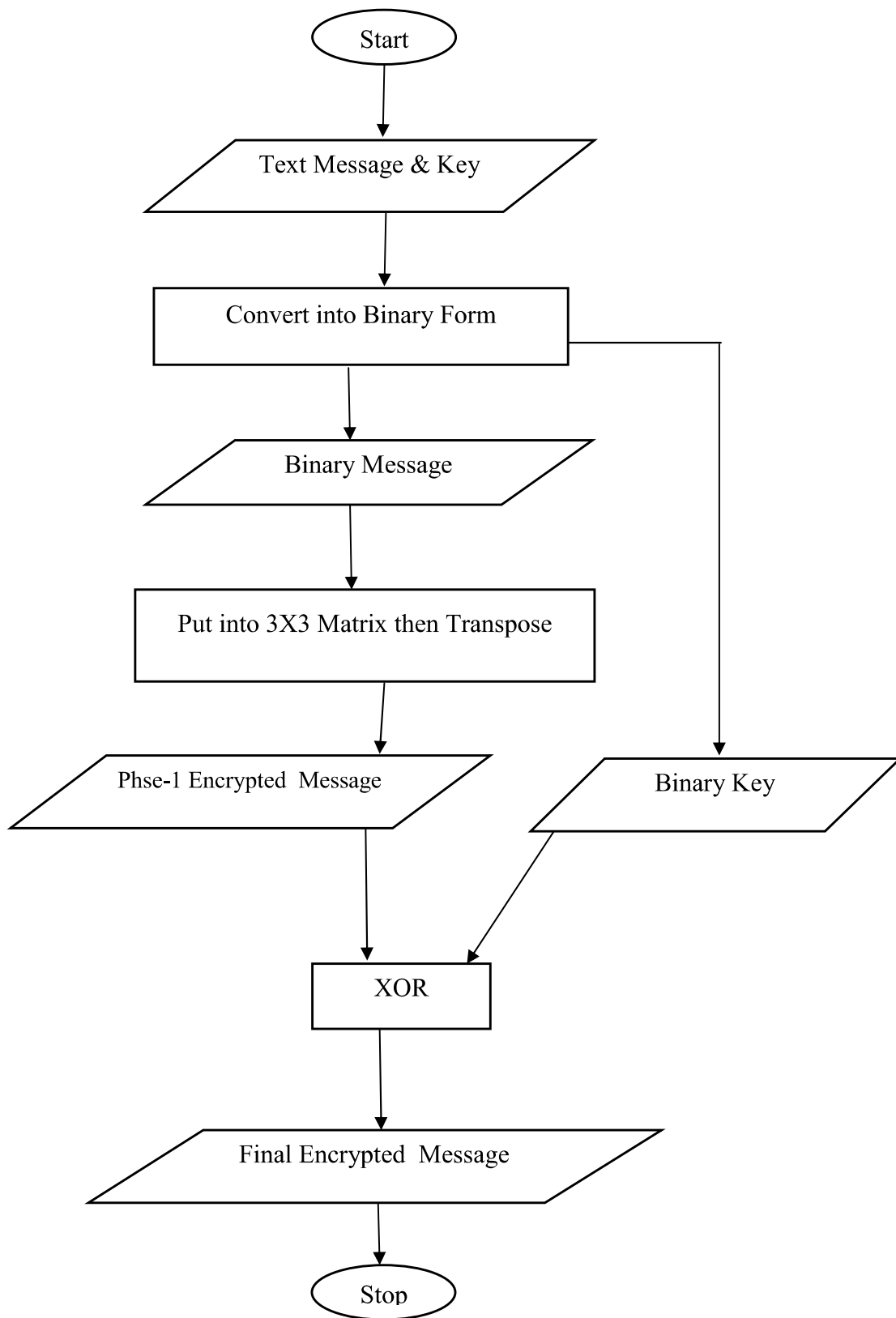


Fig 4.2 Encryption Process

4.2.1 Encryption Algorithm

Figure 4.2 shows the implementation algorithm for the encryption process.

Step 1 The text message that is to be encrypted and the key are input to the system.

Step 2 The input is then converted to the binary form.

Step 3 The binary message of the text message is then written into a 3x3 matrix and then transposed. This gives a level1 encrypted message.

Step 4 The Level 1 encrypted message obtained above is then XORed with the key binary obtained in Step 2 to get the final encrypted message.

4.2.2 Embedding Process

The encrypted message is now available and is now required to embed within the cover image inside its pixel's individual color information. The following steps are used to implement the embedding process as shown in the process flow diagram in the figure 4.3

Step 1 Input the cover image

Step 2 Check the selection counter for pixel of the image to be selected for embedding the message.

Step 3 Calculate the factory color binary of selected pixel i.e. binary form.

Step 4 Embed the 4 bits of encrypted message into the factory color LSB of the of the selected pixel as 2 bit in Red, 1 bit in Green and 1 bit in Blue factory color of selected pixel.

Step 5 Perform Mutation operation to enhance the security.

Step 6 Put this stego pixel into the the cover image corresponding position.

Step 7 Perform steps 1 through 6 until the entire encrypted message have been embedded.

After all the bits of the encrypted message are embedded into the cover image then the process completes.

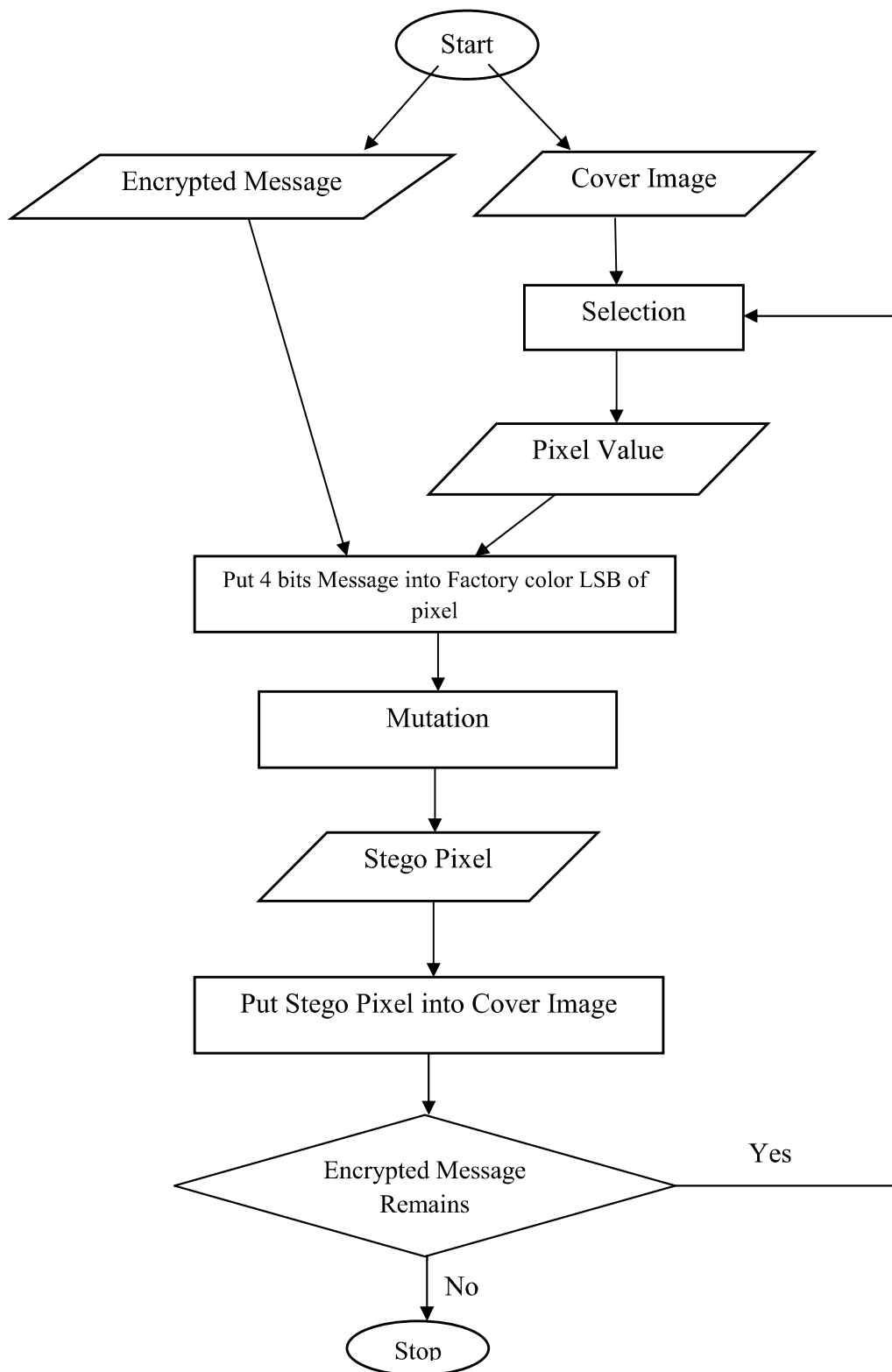


Fig. 4.3 Embedding Process

4.2.3 Mutation Process

As shown in the Step 5 a mutation process is followed to enhance the security. Generate the stego pixel from the encrypted message bits and the pixels of cover image. The mutation is a simple algorithm in which the two LSB of pixel's factory color of the cover image is simply XORed with the corresponding right most two MSB(Most Significant Bit) of the same pixel, to give the stego pixel value.

The process flow diagram for the Mutation process is as shown below in Figure 4.4 Mutation process can be understood from below process flow diagram.

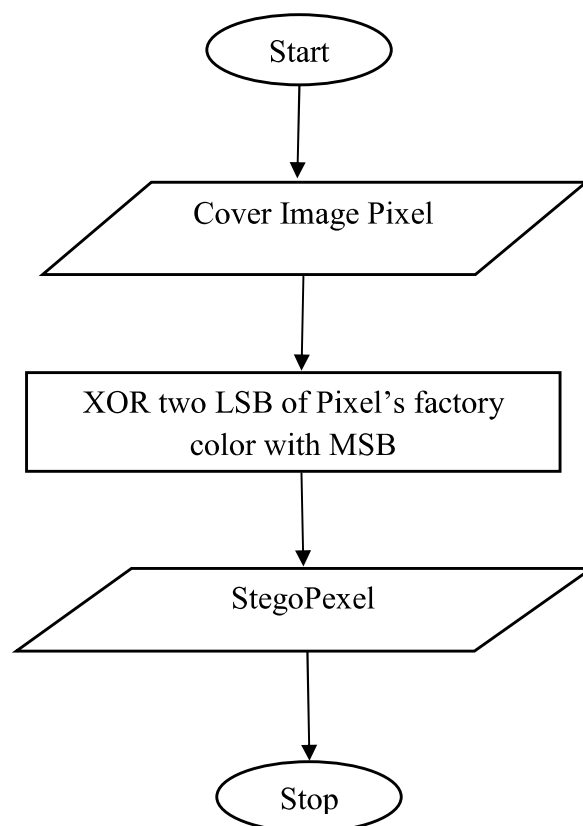
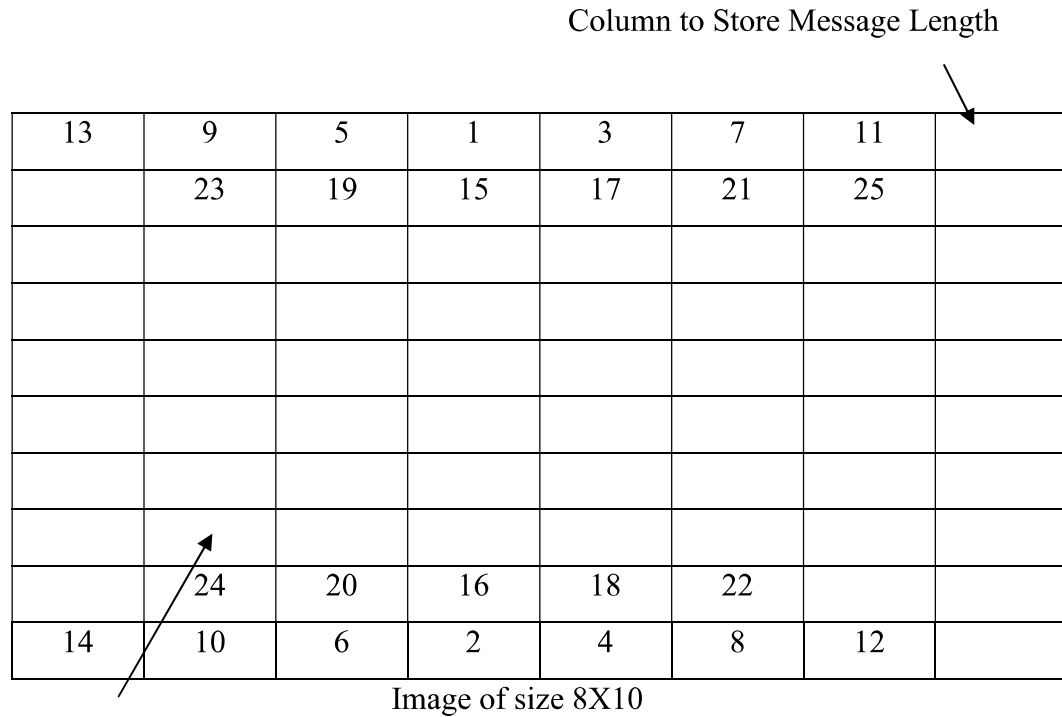


Fig. 4.4 Mutation Process

4.2.4 Pixel Selection Process

After encrypting the message, the message bits are to be embed into cover image. This process is shown below figure in fig. 4.5.



Cover Image Pixel

Fig. 4.5 Pixel Selection Process to embed

Pixels are selected from cover image as first middle top then middle bottom then top middle right then bottom middle right then top middle left then bottom middle left and so on. Last column of cover image is used for embedding the message length.

4.3 Receiver End

At the receiver end the steago-image is received. The decryption block generates the decrypted message by combining the message and the key using the decryption algorithm, the flowchart of which is shown in figure 4.6 and is described later in the chapter.

The below process flow diagram shows that the task accomplished at the receiver end of the proposed architecture. The stego image transmitted from the sender is received at the receiver end and then it is again processed through reverse mutation process to generate the original image bit.

This steago image and key sent to the Selection process, which selects the pixel from the steago image and extract the length of the message. After selection of the pixel, reverse mutation process is applied which output unmuted pixel LSB. This unmuted pixel LSB are stored into an array. This array has extracted encrypted message.

The complete step by step procedure at receiver end is as described below.

Step 1 Receive the stego image.

Step 2 Check the selection counter for pixel of the image to be selected for embedding the message.

Step 3 Apply reverse mutation process and find out the un-mutated LSB values from this.

Step 4 Store these values in an array form.

Step 5 Check for no. of selection pixels. If $2 \times \text{Message Length} = \text{No. of Selection pixels}$ then continue step 2 Otherwise go to Step 6

Step 6 This is the decryption phase. The key that is used for encryption is the input to the decryption procedure and the result is the original message after decryption.

The shared key cryptography is being used as is clear from the above discussion. The key is being shared between the sender and receiver and the same key is being used for both the encryption and the decryption process.

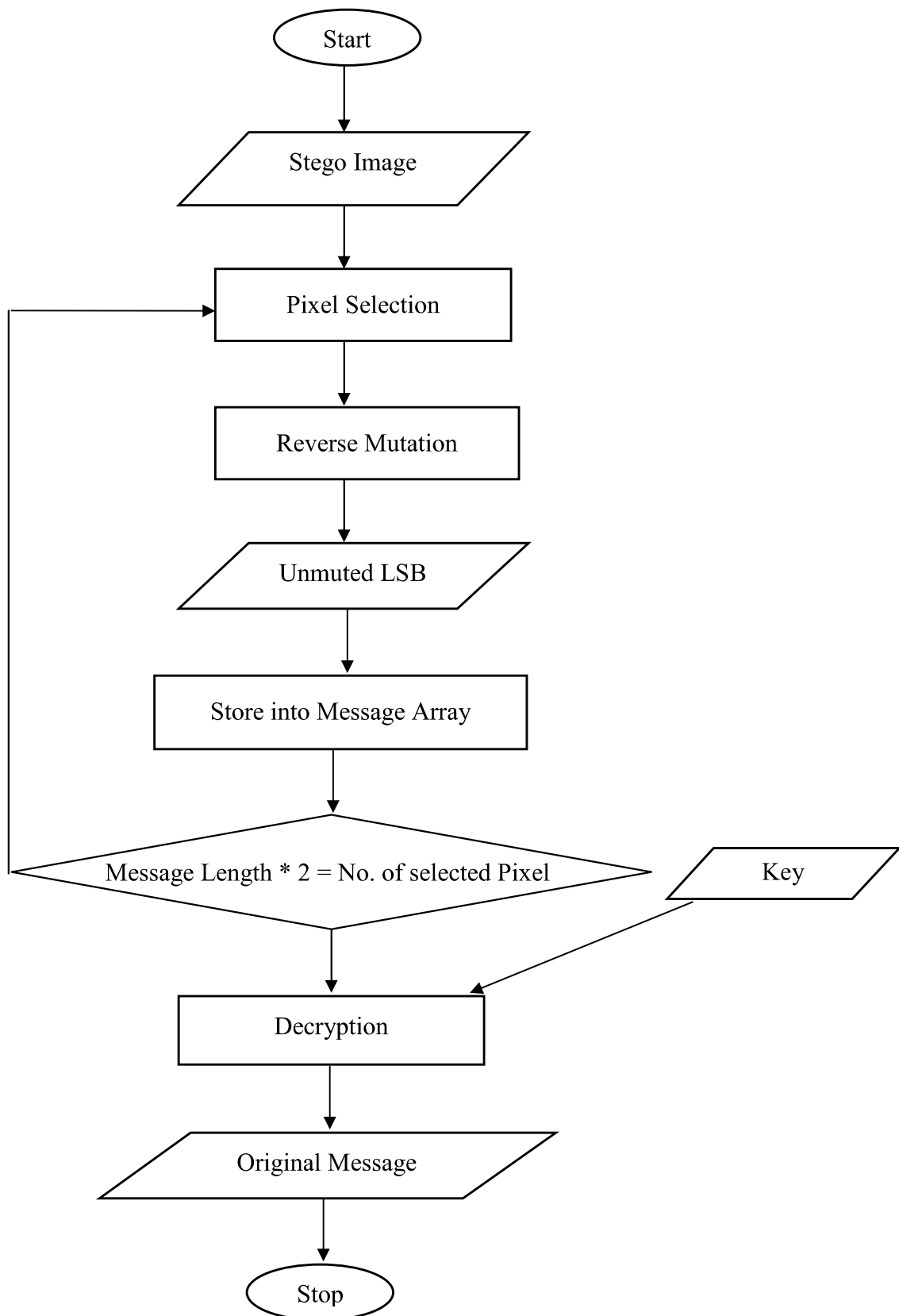


Fig. 4.6 Receiver End Procedure

4.3.1 Pixel Selection Process to extract

To understand the pixel selection from steago image and how it helps in getting encrypted message that steago image contain can be understood from the below table.

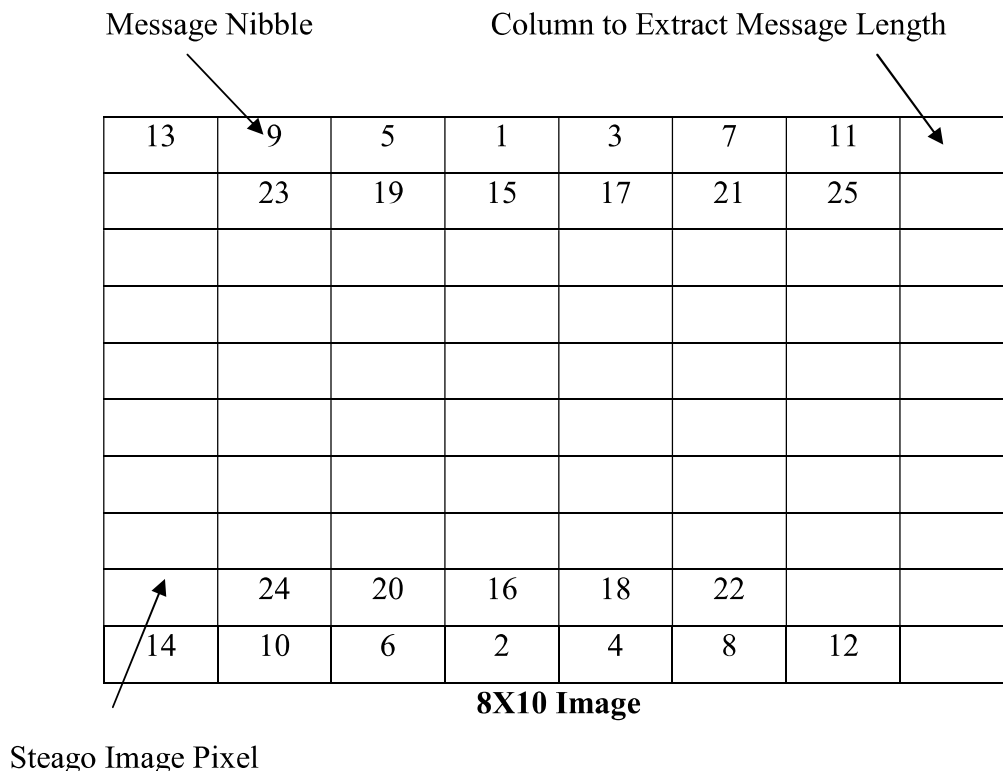


Fig. 4.7 Pixel Selection Process to extract

4.3.2 Reverse Mutation Process

The detailed description of the reverse mutation and pixel selection is shown by the below diagrams. The reverse mutation process is similar to the mutation process as is clear from the above diagram. The reverse mutation is a simple algorithm which in each step the steago image's factory color 2 bit LSB is simply XORed with the corresponding 2 bit MSB(Most Significant Bit) of the same pixel, to give the un-mutated pixel value of the LSB as shown in below figure.

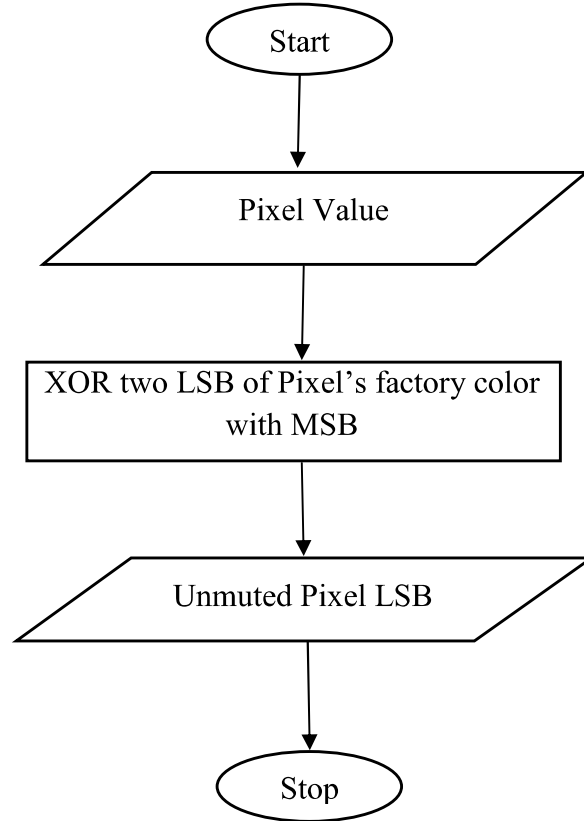


Fig. 4.8 Reverse Mutation Process

4.3.3 Decryption process

The message decryption process to remove the message bits from the stego image and retrieve the original message from the stego image is shown in the process flow diagram appearing in figure 4.10. The step by step description of the process is as below:

Step 1 The shared key is given as input and converted to its binary form.

Step 2 This binary key is then XORed with the Message Array formed after the reverse mutation process as shown in the process flow diagram in figure 4.6. This gives the Level 1 decrypted message.

Step 3 This decrypted message is put in a 3x3 matrix and transposed. This step is similar to that done in encryption that is matrix transposition. This gives the final decrypted message.

Step 4 In this step, the decrypted message obtained above is converted from the binary to character form and thus the message is retrieved.

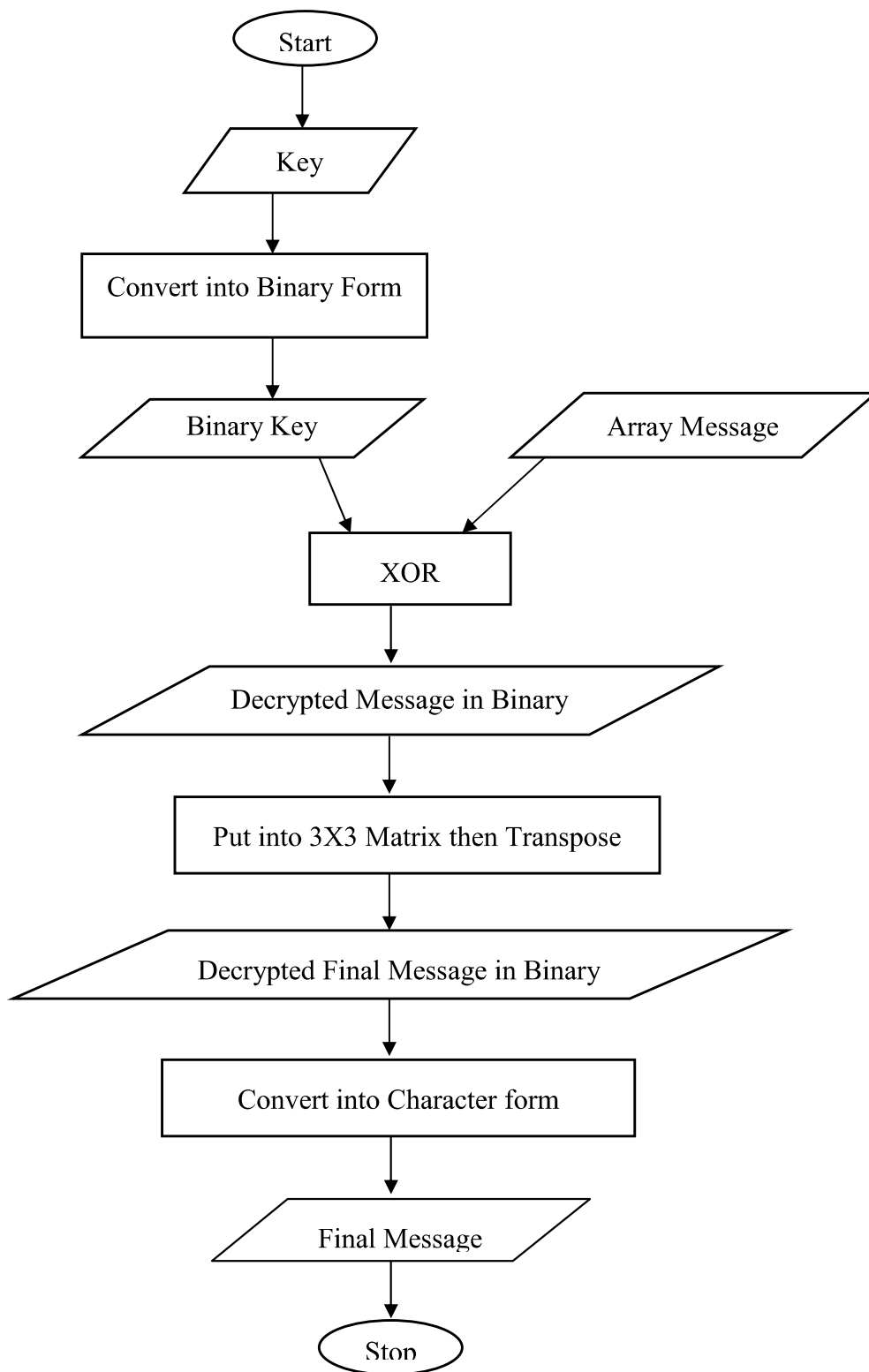


Fig. 4.9 Decryption Process

The complete system architecture used in this research work has been described in detail in this chapter. The next chapter contains some test and simulation results on various images.

CHAPTER 5

SYSTEM IMPLEMENTATION AND RESULTS

This chapter shows the results of the implementation of the processes described in previous chapters. The message which is used is a text message which can be changed and the encryption key is a one character message .the implementation steps and corresponding results of the sender and receiver end is as shown below:

5.1 Sender End

Message- My name is Abhay

Key- Hello

Step 1 Convert the message into binary.

01001101 01111001 00100000 01101110 01100001 01101101 01100101 00100000
01101001 01110011 00100000 01000001 01100010 01101000 01100001 01111001

Step 2 Put message into 3X3 Matrix then transpose. After transpose matrix write down matrix value in row major order.

00011101 01111001 00100001 00101110 11000110 00011100 01011100 01100000
10000111 01110110 00001000 01000001 01101100 10001010 00010001 01111001

Step 3 Convert Key in Binary

01001000 01100101 01101100 01101100 01101111

Step 4 Do XOR operation between transpose binary and key binary.

Encrypted Message in Binary after XOR:

01010101 00011100 01001101 01000010 10101001 01010100 00111001 00001100
11101011 00011001 01000000 00100100 00000000 11100110 01111110 00110001

Encrypted Message UMB)T9k@\$ f~1

Embed the encrypted message binary and length of message into cover image pixel's factory colour LSB as 2 bit in red 1 bit in green and 1 bit in blue colour. Selection of pixel from cover image is defined above in flow chart.

Final output Stego image.

5.2 Receiver End

Extract the encrypted message binary and message length from stego image pixel's factory color LSB as 2 bit from red, 1 bit from green and 1 bit from blue colour. Selection of pixel from steago image is defined above in Chapter 4.

Key- Hello

Step 1 Receive Encrypted Message in Binary from cover image

01010101 00011100 01001101 01000010 10101001 01010100 00111001 00001100
11101011 00011001 01000000 00100100 00000000 11100110 01111110 00110001

Step 2 Convert Key "Hello" in Binary

01001000 01100101 01101100 01101100 01101111

Step 3 Do XOR operation between encrypted message binary and key binary

After XOR in Binary

00011101 01111001 00100001 00101110 11000110 00011100 01011100 01100000
10000111 01110110 00001000 01000001 01101100 10001010 00010001 01111001

Step 4 Put XOR binary into 3X3 matrix then transpose the matrix. After transpose write matrix value in row major order.

Decrypted Message in Binary after Transposition

01001101 01111001 00100000 01101110 01100001 01101101 01100101 00100000
01101001 01110011 00100000 01000001 01100010 01101000 01100001 01111001

Step 5 Convert the decrypted message binary into text form.

Obtained Message- My name is Abhay

5.3 Result

Message- My name is Abhay

Key- Hello



Cover Image	
Steago Image	

Fig. 5.1 Results on Field Image

Message- My name is Abhay

Key- Hello

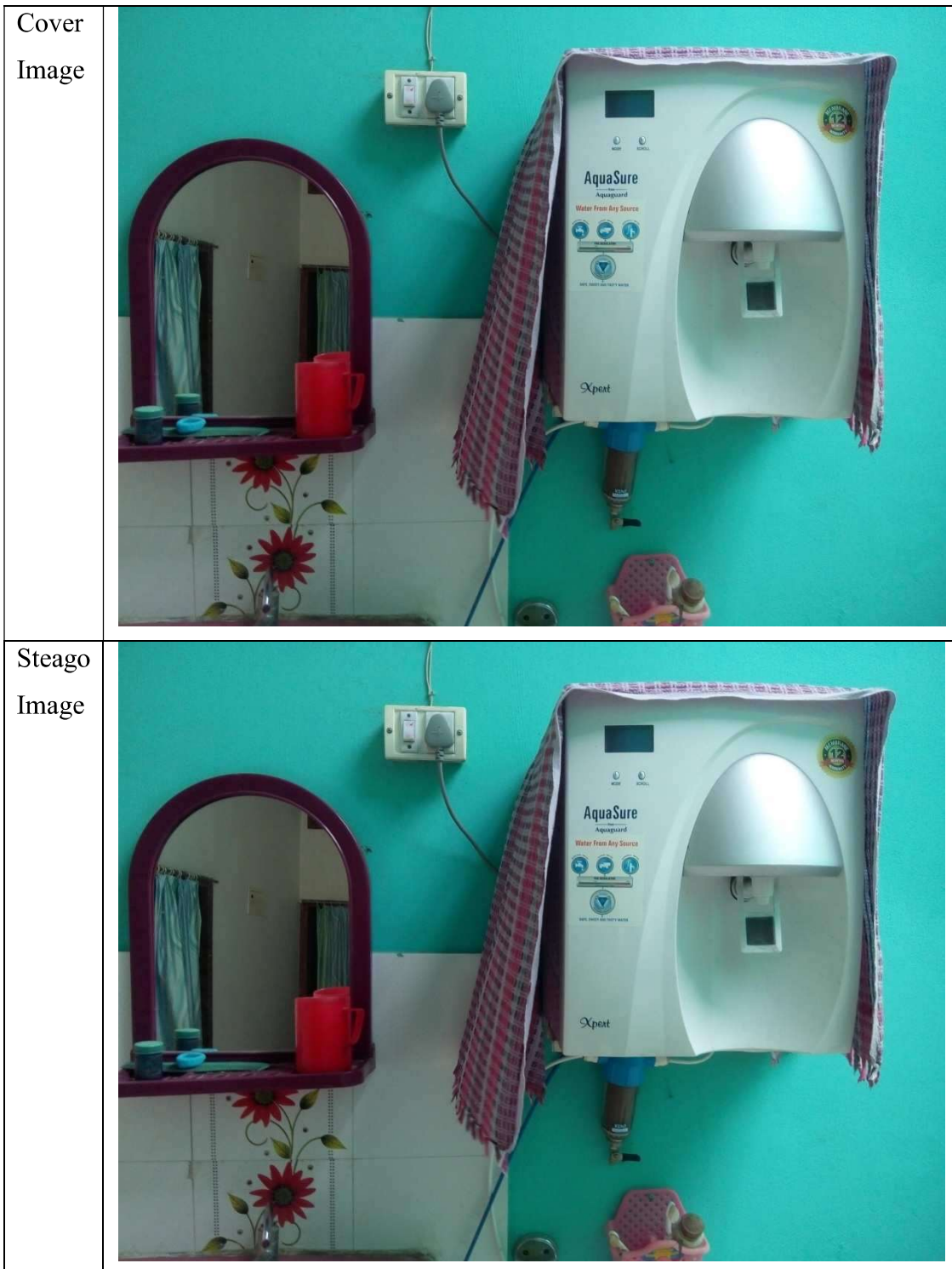


Fig. 5.2 Results on Water Filter Image

5.4 Images of Form

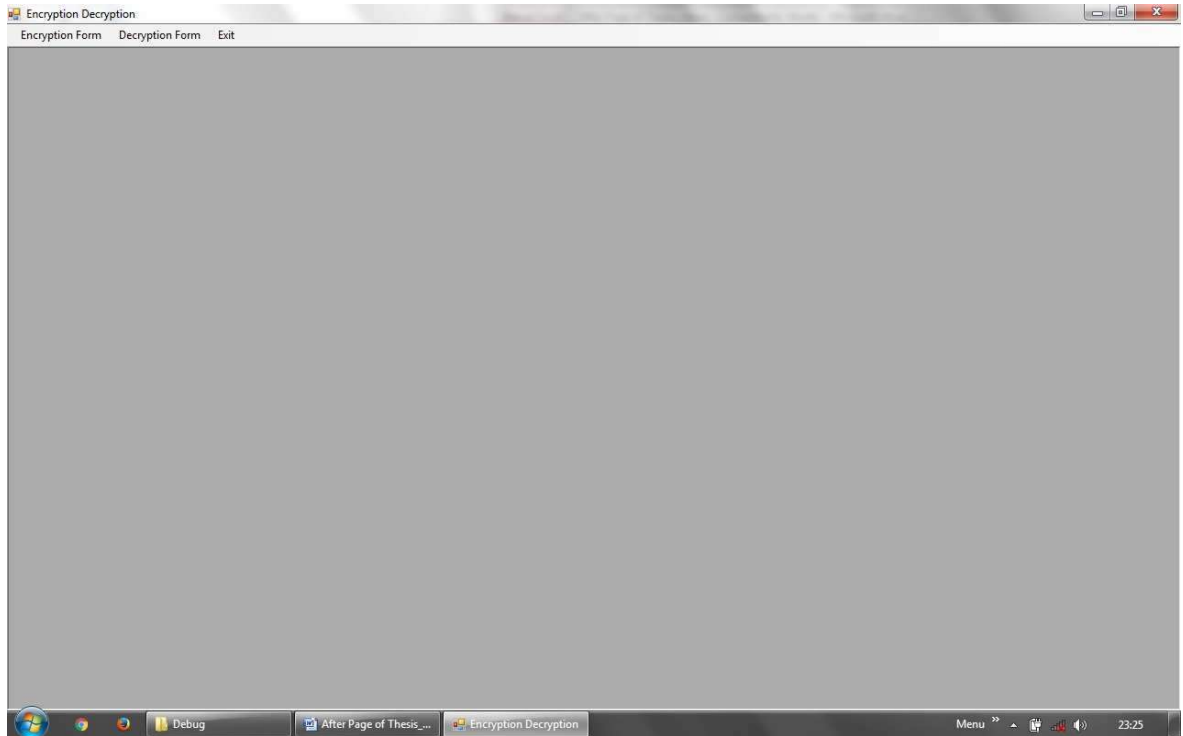


Fig. 5.3 Main Form

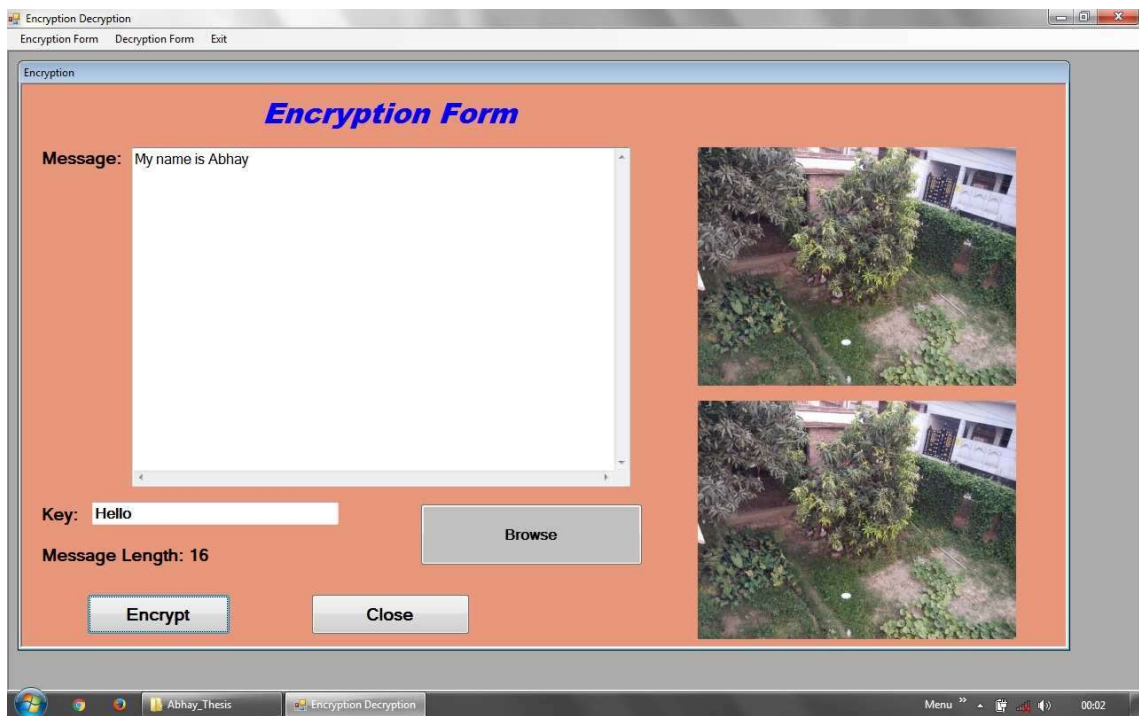


Fig. 5.4 Encryption Form

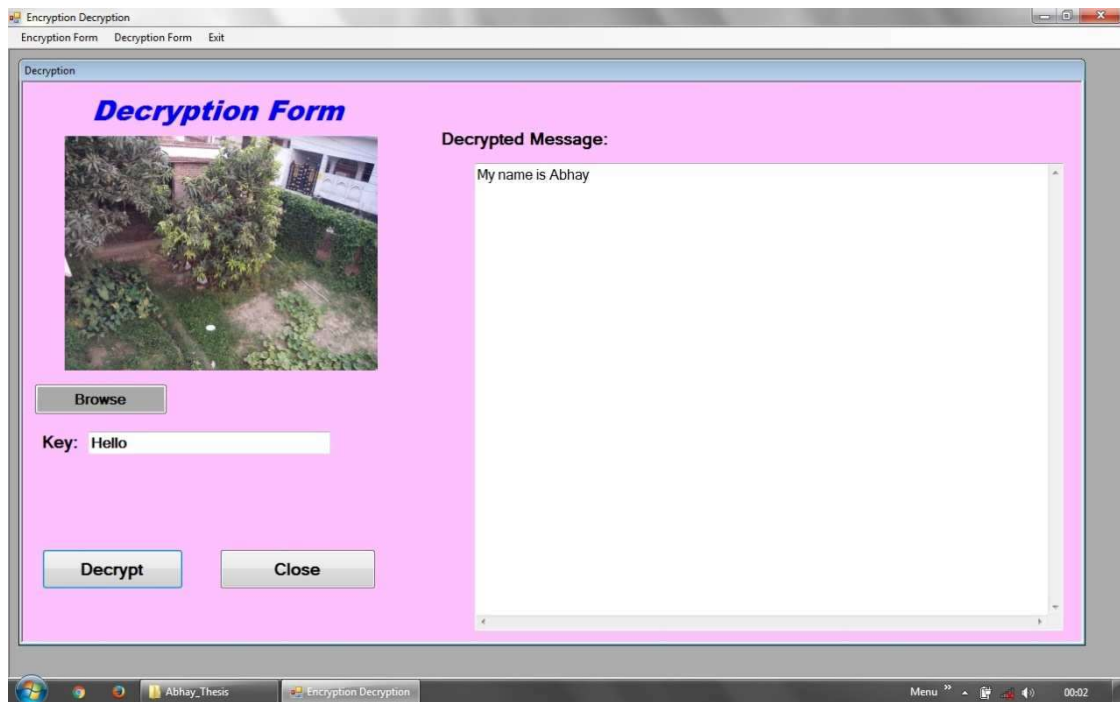


Fig. 5.5 Decryption Form

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The thesis presents a detailed study of the various terms and methods related to data hiding, steganography and watermarking. Digital data especially the image data is prone to several kinds of attacks during transmission and reception from one end to end. Thus encryption and steganographic techniques are used to prevent the originality of the message intact, under a possibility of an attack. In this research work a mutation based technique has been used to perform encryption of a hidden message using a private key which is shared between the sender and the receiver. The complete work is divided into various stages. At the sender end message encryption and embedding is performed. The encrypted message is embedded in the LSB of the cover image using the mutation process. The final outcome of the embedding process at the sender end is a stego image. At the receiver image de-embedding and encryption process is performed on the stego image. Reverse mutation process is being used along with pixel value selection. This gives the original image along with the encrypted message. Thus, the image as well as the message is kept in its original form.

6.1 Future Scope

The research work presents a new way of message encryption inside an image. The work can be further extended to examine its feasibility with various kinds of images and message patterns. The algorithm used can be easily tested along with different image types in future research works.

REFERENCES

- [1] J. K. Mandal, A. Khamrui, 2011. A Data Embedding Technique for Gray scale Image Using Genetic Algorithm (DEGGA), *International Conference on Electronic Systems (ICES-2011)*
- [2] Rehana Begum R.D, SharayuPradeep, 2014. Best Approach for LSB Based Steganography Using Genetic Algorithm and Visual Cryptography for Secured Data Hiding and Transmission over Networks, *International journal of Advanced Research in Computer Science and Software Engineering*.
- [3] Mr. VikasTyagi, 2012. Data Hiding in Image using least significant bit with cryptography, *International journal of Advanced Research in Computer Science and Software Engineering*
- [4] Samir Kumar Bandyopadhyay, TuhinUtsab Paul, AvishekRaychoudhury, 2010. GENETIC ALGORITHM BASED SUBSTITUTION TECHNIQUE OF IMAGE STEGANOGRAPHY, *Journal of Global Research in Computer Science (Volume 1, No. 5, December 2010)*
- [5] K. Jyothsana, V. Lokeswara Reddy, 2005. Dithering Technique for Digital Image Steganography, *International journal of Computer Applications (0975-8887) Volume 123-No.5, August 2015*
- [6] Hadhoud, M. M. Ismail, N. A. Shawkey, W. & Mohammed, A.Z., 2004. Secure perceptual data hiding technique using information theory, *International Conference on Electronic and Computer Engineering (ICEEC), Egypt, 2004, pp, 249-253*
- [7] Ran-Zan Wang, Chi-Fang Lin, Ja-Chen Lin, 2000, Hiding data in images by optimal moderately significant-bit replacement, *IEE Electron. Lett.* 36 (25) (2000) 20692070.

- [8] Chi-Kwong Chan, L.M.Cheng, 2002. Hiding data in images by simple LSB substitution, *Department of Computer Engineering and Information Technology, CityUniversity of Hong Kong, Hong Kong Received 17 May 2002.*
- [9] Samir K Bandyopadhyay, DebnathBhattacharyya¹, Debashis Ganguly¹, Swarnendu Mukherjee¹ and Poulami Das, A Tutorial Review on Steganography, *Heritage Institute of Technology.*
- [10] Pratap Chandra Mandal, Modern Steganographic technique: A Survey *International Journal of Computer Science Engineering Technology (IJC- SET)*
- [11] A. Cheddad, J. Condell, K. Curran and P.M. Kevitt, 2010. Digital image steganography: survey and analysis of current methods. *Signal Processing Journal.*
- [12] P. Kruus, C. Scace, M. Heyman, and M. Mundy, 2003, A survey of steganography techniques for image. *Advanced Security Research Journal.*
- [13] E Lin, E Delp, A Review of Data Hiding in Digital Images, *Center for Education and Research Information Assurance and Security Purdue University, West Lafayette, IN 47907-2086.*
- [14] W Bender, D. Gruhl, N. Morimoto, and A. Lu, Techniques for data hiding, *IBM Systems Journal, Vol. 35, No. 3 and 4.*
- [15] M.M. Amin, M. Salleh, S. Ibrahim, et al., 2003, Information Hiding Using Steganography, *4th National Conference on Telecommunication Technology Proceedings(NCTT2003), Shah Alam, Malaysia, 2003.*
- [16] Steganography and Steganalysis by J.R. Krenn January 2004.
- [17] Mamta Juneja, Data hiding Algorithm for Bitmap Images using Steganography.

- [18] Vijay Kumar sharma et.al, 2012. A steganography algorithm for hiding image in Image byimproved lsb substitution by minimize Detection. *Journal of Theoretical and Applied Information Technology* 15th February 2012. Vol. 36 No.1.
- [19] Mrs. Kavitha, KavitaKadam, AshwiniKoshti, PriyaDunghav, 2012. Steganography Using Least Significant Bit Algorithm. *Interna-tional Journal of Engineering Research and Applications (IJERA)* ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012.
- [20] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, 2005. Writing on wet paper, *IEEE Trans.on Signal Processing, Special Issue on Media Security*, vol. 53, Oct. 2005, pp. 3923-3935.
- [21] K. Solanki , K. Sullivan, U. Madhow, and B.S. Manjunath, and S. Chandrasekaran, 2005. Statistical restoration for robust and secure steganography, in *Proc. IEEE Int. Conf. on ImageProcessing, Genova, Italy*, vol. 2, 11-14 Sep. 2005, pp. 1118-1121.
- [22] K. Solanki , K. Sullivan, U. Madhow, B.S. Manjunath, and S. Chandrasekaran, 2006. Probably secure steganography: Achieving zero K-L divergence using statistical restoration, in *Proc. IEEE Int. Conf. on Image Processing, Atlanta, GA, USA*, 8-11 Oct. 2006, pp. 125-128.
- [23] K. Solanki, N. Jacobsen, U. Madhow, B.S. Manjunath, and S. Chandrasekaran, 2004. Robust image-adaptive data hiding based on erasure and error correction, *IEEE Trans. on ImageProcessing*, vol. 13, no. 12, Dec. 2004, pp. 1627-1639.
- [24] M. Kharrazi, H.T. Sencar, and N. Memon, 2006. Cover selection for steganographic embedding, in *Proc. Int. Conf. Image Processing. Atlanta, GA, USA*, pp. 117-120, 8-11 Oct.,2006.
- [25] X.G. Xia, C.G. Boncelet, and G.R. Arce, 1997. A multi resolution watermark for digital images, *IEEE Int. Conf. on Image Processing, Washington, DC, USA*, 26-29 Oct. 1997.
- [26] A. Sarkar, K. Solanki, and B.S. Manjunath, 2008. Further Study on YASS: Steganography Based on Randomized Embedding to Resist Blind Steganalysis, in *Proc.*

SPIE - Security, Steganography, and Watermarking of Multimedia Contents X, San Jose, California, vol. 6819, pp. 681917-681917-11, Jan. 2008.

[27] S. Hetzl, and P. Mutzel, 2005. A graph theoretic approach to steganography, in *Proc. 9th IFIP Int. Conf. on Communications and Multimedia Security, Salzburg, Austria, pp. 119-128, 19-21 Sep. 2005.*

[28] T. Pevny, and J. Fridrich, 2007. Merging Markov and DCT features for multi-class JPEG steganalysis, in *Proc. SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX, San Jose, CA, vol. 6505, Jan 2007, pp. 03-04.*

[29] R.A. Johnson, 2003. Miller & Freund's Probability and Statistics for Engineers, *Prentice Hall of India Pvt. Ltd., New Delhi, 2003.*

[30] C. Chen, Y.Q. Shi, W. Chen, and G. Xuan, 2006. Statistical moments based universal steganalysis using JPEG-2D array and 2-D characteristic function, in *Proc. Int. Conf. on Image Processing, Atlanta, GA, USA, 8-11 Oct., 2006, pp. 105-108.*

Curriculum Vitae

Abhay Deep Singh

E-mail: abhaydeep88@gmail.com

Contact: 8090190020

Professional Qualification:

- Pursuing M.Tech-CS(SE) Final Year, from B.B.D.U., Lucknow.
- MCA from IGNOU (MSIT New Delhi) in Dec 2008 with 60.39% marks.
- ADCA from IGNOU (MSIT New Delhi) in Dec 2008 with 59.33% marks.
- BCA from IGNOU (M.M.M. Engineering College Gorakhpur) in Dec 2006 with 62.92% marks.

Academic Qualification:

- 10th from U.P. Board Allahabad in 1998 with 60.17% marks.
- 10+2 from U.P. Board Allahabad in 2000 with 51.8% marks.

Personal Details:

Name : Abhay Deep Singh
Father's Name : Sri Ram Kamal Singh
Mother's Name : Smt. Sharda Singh
Marital Status : Married
Date of Birth : 01th July of 1984

Address for Communication

463-A, Andhiyari Bagh(North)
Gorakhnath Mandir
Gorakhpur (UP)-273015
E-mail.: abhaydeep88@gmail.com
Contact Number: 8090190020

Date: -

(Abhay Deep Singh)

BABU BANARASI DAS UNIVERSITY, LUCKNOW

CERTIFICATE OF THESIS SUBMISSION FOR EVALUATION

(Submit in Duplicate)

1. Name:

2. Enrollment No. :

3. Thesis title:

.....

.....

4. Degree for which the thesis is submitted:

5. Faculty of the University to which the thesis is submitted

.....

6. Thesis Preparation Guide was referred to for preparing the thesis. ☐ YES ☐ NO

7. Specifications regarding thesis format have been closely followed. ☐ YES ☐ NO

8. The contents of the thesis have been organized based on the ☐ YES ☐ NO
guidelines.

9. The thesis has been prepared without resorting to plagiarism. ☐ YES ☐ NO

10. All sources used have been cited appropriately. ☐ YES ☐ NO

11. The thesis has not been submitted elsewhere for a degree. ☐ YES ☐ NO

12. Submitted 2 spiral bound copies plus one CD. ☐ YES ☐ NO

(Signature of the Candidate)

Name:

Roll No

Enrollment No:

BABU BANARASI DAS UNIVERSITY, LUCKNOW

CERTIFICATE OF FINAL THESIS SUBMISSION

(To be Submit in Duplicate)

1. Name:

2. Enrollment No. :

3. Thesis title:

.....

.....

4. Degree for which the thesis is submitted:

5. School (of the University to which the thesis is submitted)

.....

6. Thesis Preparation Guide was referred to for preparing the thesis. ☐ YES ☐ NO

7. Specifications regarding thesis format have been closely followed. ☐ YES ☐ NO

8. The contents of the thesis have been organized based on the ☐ YES ☐ NO
guidelines.

9. The thesis has been prepared without resorting to plagiarism. ☐ YES ☐ NO

10. All sources used have been cited appropriately. ☐ YES ☐ NO

11. The thesis has not been submitted elsewhere for a degree. ☐ YES ☐ NO

12. All the corrections have been incorporated ☐ YES ☐ NO

13. Submitted 4 hard bound copies plus one CD. ☐ YES ☐ NO

(Signature of the Supervisor)

Name:

(Signature of the Candidate)

Name:

Roll No

Enrollment No:

Curriculum Vitae

Abhay Deep Singh

E-mail: abhaydeep88@gmail.com

Contact: 8090190020

Professional Qualification:

- Pursuing M.Tech-CS(SE) Final Year, from B.B.D.U., Lucknow.
- MCA from IGNOU (MSIT New Delhi) in Dec 2008 with 60.39% marks.
- ADCA from IGNOU (MSIT New Delhi) in Dec 2008 with 59.33% marks.
- BCA from IGNOU (M.M.M. Engineering College Gorakhpur) in Dec 2006 with 62.92% marks.

Academic Qualification:

- 10th from U.P. Board Allahabad in 1998 with 60.17% marks.
- 10+2 from U.P. Board Allahabad in 2000 with 51.8% marks.

Personal Details:

Name : Abhay Deep Singh
Father's Name : Sri Ram Kamal Singh
Mother's Name : Smt. Sharda Singh
Marital Status : Married
Date of Birth : 01th July of 1984

Address for Communication

463-A, Andhiyari Bagh(North)

Gorakhnath Mandir

Gorakhpur (UP)-273015

E-mail.: abhaydeep88@gmail.com

Contact Number: 8090190020

Date: -

(Abhay Deep Singh)