A New Steganography Techniques using Huffman Coding & Hamming Code

A dissertation submitted in the partial fulfillment for the award of Degree of

Master of Technology

In

Software Technology

by

Abhishek (2K14/SWE/01)

Under the guidance of

Mr. Vinod Kumar

(Associate Professor)

Department of Computer Engineering, DTU

DEPARTMENT OF SOFTWARE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

CERTIFICATE

Delhi Technological University
(Government of Delhi NCR)
Bhawana Road, New Delhi-42

This is to certify that the thesis entitled **"A New Steganography Techniques using Huffman Coding & Hamming Code"** done by Abhishek (2K14/SWE/01) for the partial fulfillment of the requirements for the award of degree of **Master of Technology** Degree in **Software Technology** in the **Department of Computer Engineering**, Delhi Technological University, New Delhi is an authentic work carried out by him under my guidance.

Project Guide:
Mr. Vinod Kumar
Associate Professor
Department of Computer Engineering
Delhi Technological University, Delhi

ACKNOWLEDGEMENT

I take this opportunity to express my deep sense of gratitude and respect towards my guide **Mr. Vinod Kumar (Associate Professor) Department of Computer Engineering.**

I am very much indebted to her for her generosity, expertise and guidance I have received from him while working on this project. Without his support and timely guidance the completion of the project would have seemed a far –fetched dream. In this respect I find myself lucky to have my guide. He have guided not only with the subject matter, but also taught the proper style and techniques of documentation and presentation

Besides my guides, I would like to thank entire teaching and non-teaching staff in the Department of Computer Engineering, DTU for all their help during my tenure at DTU. Kudos to all my friends at DTU for thought provoking discussion and making stay very pleasant.

Abhishek
M.Tech Software Engineering
2K14/SWE/01

Abstract

In Steganography, the total message will be invisible into a cover media such as text, audio, video, and image in which attackers don't have any idea about the original message that the media contain and which algorithm use to embed or extract it. In this thesis, the proposed technique has focused on Bitmap image as it is uncompressed and convenient than any other image format to implement LSB Steganography method. For better security symmetry key cryptography technique has also been used in the proposed method. Before applying the Steganography technique, Symmetry key cryptography will change the secret message into cipher text to ensure two layer security of the message. In the proposed technique, a new Steganography technique is being developed to hide large data in Bitmap image using filtering based algorithm. This algorithm first compresses the data, so, big amount of data is embed in small space, and on that code word, Hamming Encoding is also applied. If there is any change in the pixels of image due to noise, hamming encoding will help us to recover the message. This algorithm uses MSB bits for filtering purpose. This method uses the concept of status checking for insertion and retrieval of message. This method is an improvement of Least Significant Bit (LSB) method for hiding information in images. It is being predicted that the proposed method will able to hide large data in a single image retaining the advantages and discarding the disadvantages of the traditional LSB method. Various sizes of data are stored inside the images and the PSNR are also calculated for each of the images tested. Based on the PSNR value, the Stego image has higher PSNR value as compared to other method. Hence the proposed Steganography technique is very efficient to hide the secret information inside an image.

Table of Content

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

Cyberspace makes the life of the peoples much effortless than before. They can use it to pay their debts, purchase their goods, and exchange important messages between them. Without insulate that valuable information, attackers or intruders can get them in disparate ways.

In IT sector, the security of information is very important part and it is growing rapidly. An extent amount of classified data is being vanished every year during transmission by the trespasser. Many ciphering techniques are extensively used to inscribe and interpret. But occasionally data inscribing does not seem enough and veiling of the data is needed more. The technique used for this idea is steganography.

1.1 Steganography

Steganography is the discipline or proficiency of obscuring a record, message, figure, video inwards another medium like a record, message, figure, and video. Human Visual System (HVS) cannot notice a minor change which happens in the cover media such as image, audio, video.

Steganography fuses the two Greek words, Steganos meaning "concealed or protected" and Graphein meaning "writing".

The main objective of steganography is to veiling the presence of communication and the original information does not shown to the third party.

There are three elements in steganography system:

- embedding message,
- cover object (this object hides message) ,and
- stego object (this object is cover object with message embedded in it).

The working of steganography in digital media is explained by the following diagram. Figure 1.1 shows the general steganography system at both sender and receiver side.

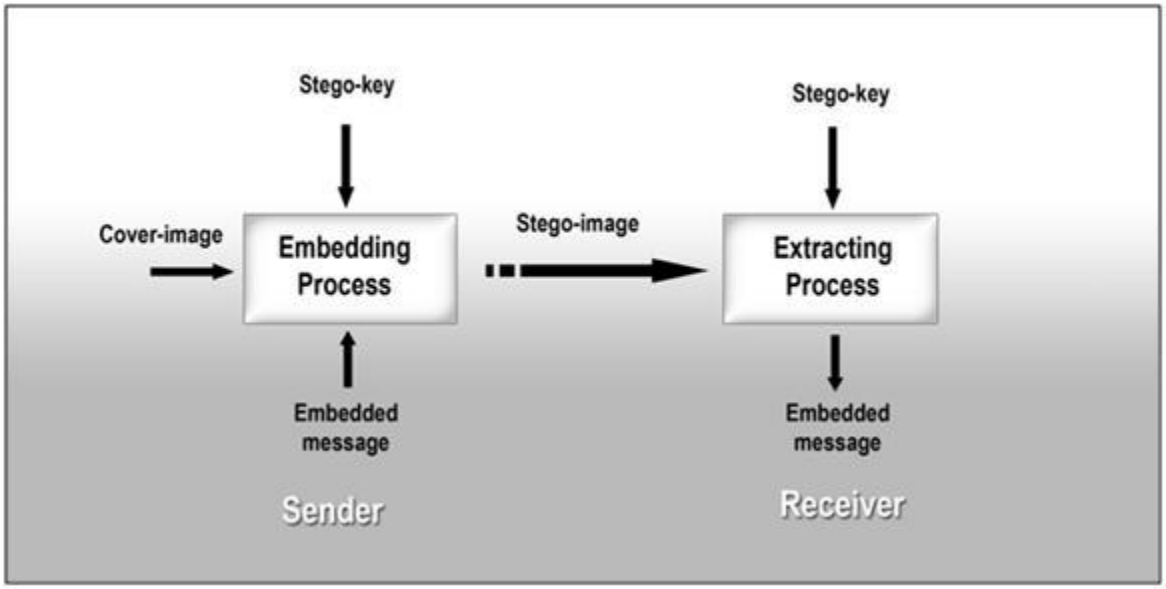


Figure 1.1: General Steganography System

1.2 Comparison between Steganography, Cryptography and Watermarking

Steganography is often considered similar to cryptography and watermarking. But they are all different methods of the spy family.

- Cryptography ensures message integrity.
- Steganography hides the message in the cover media.
- Watermarking ensures message integrity.

Cryptography and watermarking are the other forms of data hiding.

Cryptography is a method in which a message is obfuscate and sent in a deficient composition. The basic difference between steganography and cryptography is that the cryptography obfuscates the content of communication whereas steganography just obscure the data.

It has been observed that the goal of cryptography and steganography is same, but the way the goal is achieved is different.

		Cryptography	Steganography
Goal		Obfuscate the content of communication	Hide the case of communication
Characteristics	Secrecy	Ciphertext is illegible	Embedded data is 'invisible' to the third party
	Security of Communication	Relies on the confidentiality of the key	Relies on the confidentiality of the method of embedding
	Warranty of Robustness	Complexity of the ciphering method	Intuitive invisibility / statistical invisibility / compliance with protocol specification
	Attacks	Detection is easy / extraction is complex	Detection is complex / extraction is complex
Countermeasures	Technical	Reverse Engineering	Constant monitoring and analysis of exchanged data
	Legal	Cryptography export laws	Rigid device / protocol specification

Table 1.1: Cryptography vs. Steganography

Cryptography encodes or encrypts the data or information so that the intruder cannot read it. On the other hand, steganography attempts to hide the existence of information or data from the intruder.

The advantage of steganography over cryptography alone is that the intended secret message does not attract attention to itself as an object of scrutiny.

Watermarking ensures message integrity and often use for authentication and copyright protection. Watermarking is used to make the digital file recognizable. There are visible watermarking and invisible watermarking. In visible watermarking, the mark is visible on digital file and on the other hand, in invisible watermarking, mark is only visible to only creator.

The main objective of watermarking is to prohibit the illegal copying or claim of ownership on digital files.

		Watermarking	**Steganography**
Goal		Protect the carrier	Protect secret message from disclosure
Characteristics	**Secrecy**	Invisibility or perceptual visibility depending on the requirements	Embedded message is invisible to the third party
	Type of Robustness	Robustness against removal or tampering	Robustness against detection
	Effect of signal processing / compression	Must not lead to the loss of the watermark	May lead to the loss of the hidden data
	Type of Carrier	Digital files – audio, video, text, or images	Any service, protocol, file, environment employing digital representation of data

Table 1.2: Watermarking vs. Steganography

1.3 Characteristics of Steganography, Cryptography and Watermarking

The common characteristic between Steganography, Cryptography and Watermarking is that they transmit the secret information in such a way that only the receiver is able to decrypt the data. These techniques had been prevalent during the ancient times have been transported to the digital world. It has become nearly impossible to extract or detect the secret messages.

In digital domain Steganography and watermarking have a tie and it is majorly used in digital images. These have other uses as well. Both cannot exist by themselves and hence they require cover objects. Steganography requires a cover media to carry the secret information and watermarking requires a carrier object which it is intended to protect. These similarities create a link within them and hence some modifications can lead the transportation from one technique to another. Due the similarities present between these it is difficult to distinguish between the both but actually there is a remarkable difference between them.

Cryptography encrypts data in two methods secure or unbreakable (e.g. One-time pad) systems and breakable (e.g. RSA) systems. Through both the systems communication carried out is known to all. But it is time consuming and often fruitless to crack a code. The robustness of the code lies upon the difficulties faced while reversing the code in different permutations and combinations. Due to its robustness it is used for security purposes. For example Cryptography is used for online shopping, banking etc. The credit card number, expiration etc and other crucial information's are encrypted and sent so that an unintended user can't access the details.

Steganography offers high carrier capacity keeping embedded message invisible and maintaining the fidelity of the cover media. The efficiency of the Steganographic method is that one shouldn't know that a media file has been altered in order for embedding. If the malicious user knows if there is some alteration the Steganographic method is defeated and less efficient. The embedded message is very fragile and hence if any modification is done to the stego image the whole secret message is corrupted. The effectiveness lies on the ability to fool an unintended user. The layers of communication can be more than one layer. A secret message can be embedded with a digital image which in turn can be embedded within another digital media or video clippings.

Watermarking is required in order for authentication and copyright protection of digital files. The embedded watermarking is required in an object to make it impossible to remove completely. If the embedded watermarking is removed, then the marked object is either distorted or destroyed making it useless for anyone. This is the reason why watermarking is more robust when compared to the other image processing techniques, such as compression, cropping, rotation etc. hence even if a tiny bit of information is extracted by modification and tempering the rightful owner can still claim ownership. If the owners name is embedded in the digital image and the particular image then the original information is tampered and destroyed. Unlike Steganography, it is acceptable for everyone one to see the watermark embedded in it including the invisible ones.

1.4 Requirements of Steganography

Any digital media such as image (gray, color), text, audio or video etc., can be used as a cover media. In order to obscure and carry the information the cover media is required. Mainly digital images are used as a cover media. The secret information or message is hidden within the digital image. The nominees for cover image are the innocuous image which consists of scenery, people and other objects. Now days, due to the affordable price of digital cameras, the availability of natural images are not a problem.

Due to the high resolution of natural images, they are the best candidates for cover images providing flexibility and other necessary need.

The images as cover media is selected are depending upon how the Human Visual System (HVS) works. The intensity level ranges from 0 to 255 in an gray or color images. The human eye cannot detect any difference between the pixel intensity, for example, 240 and 241. To the human eye, both the pixel intensities value appear to be same. A natural image also consists of noise and hence there is lots of pixel intensities value difference, mainly in the edges. The

Human Visual System (HVS) would not be able to differentiate between the image before modulation and the resulted image after modulation.

The secret information is embedded in the cover image. The resultant image is known as Stego-image. Both the stego-image and the cover image will appear to be same. Usually, the human eye cannot differentiate between the stego-image and the original image except that there is a major change between the images. Due to the major change cases the steganography methods employed are ineffective.

So, there are certain Stego-system criterion which has to be followed while doing a steganographic implementation. Those are describes as follows:

- The major criteria that the cover image should not be significantly modified or changed.
- During transmission, there are chances of noise disturbance to occur which cannot be eliminated. The algorithm must deal with this issue.
- The embedded information should be immune to modifications or changes occur to the cover media. Means embedded information does not change due to changes in cover media.
- The embedding capacity or efficiency must be improved.
- The stego-image should be tough to steganalysis.
- The embedding capacity of the cover media should be large.

The three basic requirements of the steganography are:

- Robustness,
- Capacity, and
- Imperceptibility.

For having an effective steganography method, all these requirements should be satisfied.

1.5 Classification of Steganography Techniques

Steganography techniques are classified into various categories based on the embedding functions. They are described as follows:

1.5.1 Spatial Domain Technique

In spatial domain steganography techniques image pixels values are converted in binary values and some of the bits are changed for hiding secret data. There are many categories of spatial domain Techniques which differ mainly on the basis of manipulation of different bits in pixel values. Least significant bit (LSB)-based technique is one of the simplest and most widely used techniques that inserts or hides the secret message in the LSBs of pixel values without much visual distortion in the cover image. Another technique employs embedding of message bits at randomly chosen pixels. This technique is Pseudorandom LSB in which random pixels are chosen using algorithm where bits of secret data are embedded.

Some general spatial domain algorithms are as follows:

1.5.1.1 Least significant bit substitution Technique (LSB)

In this technique, the LSBs of the pixel values of cover image are modified according to bits of message. The simplest of LSB steganography techniques is LSB replacement for all pixels of image. Since only LSB is changed, difference between the cover (i.e. original) image and the stego-image is hardly noticeable.

1.5.1.2 Pseudo-Random LSB Encoding Technique

In this technique, a random-key is used to choose the pixels randomly where message bits will be stored. This will make the message bits more difficult to find for an intruder. Moreover the colored image has three planes (RGB). The data can be hidden in the LSB of any color plane of the randomly selected pixels. With the use of this technique it will be difficult for the attacker to identify the pattern in which message bits are hidden, as no particular pattern is followed for

embedding subsequent message bits. At transmitter side a random key is used to randomize the cover image and embeds the information bits into the LSB of the pixels. The transmitting and receiving end share the random-key. This random-key is used as a seed for pseudo-random number generator for selecting pixel locations in an image for hiding the secret message bits.

1.5.2 Transform Domain Technique

In transform domain, the information is embedded in cover image which is transformed in frequency domain. The information bits are inserted into transformed coefficients of image. Many different transformations can be used for cover image before hiding the secret data. This method of steganography gives more robustness against attacks, as the secret data is stored in image at those areas which are not directly exposed and will remain unchanged after cropping or resizing of image.

Some general Transform domain algorithms are as follows:

1.5.2.1 DCT Based Steganography

In Discrete Cosine Transform steganography technique, image is converted into frequency domain. This transformation process is divided into four distinct and independent phases.

Phase 1: In this phase, the image is divided into blocks of pixel size of 8 x 8.

Phase 2: Each block is subjected to DCT transformation to convert the information into frequency domain.

Phase 3: The information from step 1 is quantized to remove unnecessary information in frequency domain.

Phase 4: Standard compression techniques are applied to bit pattern.

This transformation is mainly used when the stego-image is prone to image modification processes like compression, cropping etc. This explains the reason for storing data in the areas of the image which are not much affected after application of these processes.

1.5.2.2 DWT Based Steganography

Discrete Wavelet Transform (DWT) steganography is another frequency domain transformation in which haar matrices are used for transformation of images is discrete domian. This technique is divided into two operations i.e. horizontal operation and vertical operation. Various step of the procedure are as follows:

Step1: The pixels in a row are scanned in from left to right and addition & subtraction operations are performed on neighboring pixels. On left hand side summation of the pixels is stored and on the right hand side difference value is stored. This process is repeated for all the rows. The addition of pixels gives the low frequency component and the difference of pixels gives the high frequency component of the original-image.

Step2: The pixels are scanned in column in vertical direction from top to bottom. The sum and difference is calculated on neighboring pixels. The summation of pixels in column is stored at top and difference value is stored at the bottom. This process is repeated for all the columns. The information is converted into four sub-bands termed as LL, HL, LH, and HH. The LL sub-band looks very similar to the original image as it is the low frequency portion.

1.5.3 Distortion Technique

In this technique, information is stored by changing value of the pixel that is termed as distortion. The signal distortion is provided by introducing deviation in the pixel value for embedding secret data. At the receiver side the same deviation is used for retrieving hidden data from the image. The original image is the fundamental requirement at the receiver side to retrieve the secret data. The deviation between original image and stego-image is used to recover data. To implement this method certain modifications are applied to the cover image. The secret message bits are inserted in pixels in cover image which are chosen pseudo-randomly. During retrieval process

the original image and stego-image are compared. If the value of pseudo-randomly chosen pixel is different the message bit is logic 1 otherwise it is logic 0.

1.5.4 Visual Cryptography (VC) Technique

Instead of using image directly to embed data, it is broken into two or more parts called shares. Message is broken into bits and inserted into shares which in turn are transmitted via different paths. An intruder can't recover complete until data all the shares are received and combined in particular order. At the intended receiver side all the shares are received and stacked to recover the original data. Thus this technique provides a simple and robust method to embed data.

1.6 Motivation

An extent amount of classified information or data is being vanished every year during transmission by the trespasser. Many cryptography techniques are extensively used to inscribe and interpret. But intermittently, data inscribing does not seem enough and veiling of the data is needed more. The technique used for this idea is steganography.

Steganography is the discipline or proficiency of obscuring a record, message, figure, video inwards another medium like a record, message, figure, and video. Human Visual System (HVS) cannot notice a minor change which happens in the cover media such as image, audio, video.
Steganography fuses the two Greek words, Steganos meaning "concealed or protected" and Graphein meaning "writing".
Digital image based steganography, in recent years, has established itself as an important discipline of the signal processing. This is happening due to the active interest from the research society.
Many existing algorithms of steganography, target on the embedding approach, but these algorithms does not give any attention to the pre-processing stages, such as encryption.

The algorithm either focuses on the complexity of the embedding method or the amount of the information to be embedded in the cover media, but not focus on both at the same time.
Many present algorithms take for granted that resilience to noise, double compression, and other image processing controls are not required in the steganography background.
From the above discussion it is evident that there is a need for the development of an hybrid algorithm which can take of both the complexity of the information hiding and the amount of the information to be embedded in the cover media.

1.7 Organization of Thesis

The remaining sections of the thesis are organized as follows:

Chapter 2 provides a description of various proposed steganography algorithms. It gives an insight to the advantages as well as disadvantages of the available techniques.

Chapter 3 provides a description of the algorithms used in the proposed work.

Chapter 4 provides the description of proposed algorithm and its architecture.

Chapter 5 provides the result of the proposed method.

Chapter 6 concludes the thesis and depicts the possible improvements in the research work in future.

CHAPTER 2: LITERATURE REVIEW

In this chapter we provide the literature review of many proposed techniques in steganpgraphy and the type of attacks done on steganography. In section 2.1 we discuss briefly some of the existing steganographic techniques. In section 2.2 we present some of the steganalytic attacks proposed till date as a counter measure to the steganographic algorithms.

2.1 Existing Steganographic Techniques

The steganographic algorithms proposed in literature can broadly be classified into two categories.

- Spatial Domain Techniques
- Transform Domain Techniques

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

2.1.1 Spatial Domain

These techniques use the pixel gray levels and their color values directly for encoding the message bits. These techniques are some of the simplest schemes in terms of embedding and extraction complexity. The major drawback of these methods is amount of additive noise that creeps in the image which directly affects the Peak Signal to Noise Ratio and the statistical properties of the image. 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.

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 gray levels is used to represent the message bit. 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 kind of embedding also leads to an asymmetry and a grouping in the pixel gray values (0,1);(2,3);. . . (254,255). This asymmetry is exploited in the attacks developed for this technique as explained further in section 2.2. 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. 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. For further reducing 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.

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. The variance value for a block of pixels is used to compute the number base to be used for embedding. A similar kind of algorithm based on human vision sensitivity has been proposed by [5] by the name of Pixel Value Differencing. 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].

According to [15], "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." 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 $\frac{q}{2^q - 1}$.

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. The maximum embedding capacity achievable is 2=3 = 0:67 bits/pixel. F5 algorithm [14] is probably the most popular implementation of Matrix Encoding. LSB replacement technique has been extended to multiple bit planes as well. Recently [3] has claimed that LSB replacement involving more than one least significant bit planes is less detectable than single bit plane LSB replacement. Hence the use of multiple bit planes for embedding has been encouraged. But the direct use of 3 or more bit planes leads to addition of considerable amount of noise in the cover image. [8] and [9] have given a detailed analysis of the noise added by the LSB embedding in 3 bit planes. Also, a new algorithm which uses a combination of Single Digit Sum Function and Matrix Encoding has been proposed. It has been shown analytically that the noise added by the proposed algorithm in a pixel of the image is 0:75p as compared to 0:875p added by 3 plane LSB embedding where p is the embedding rate. One point to be observed here is that most of the approaches proposed so far are based on minimization of the noise embedded in the cover by the algorithm. Another direction of steganographic algorithm is preserving the statistics of the image which get changed due to embedding. Chapter 2 of this thesis proposes two algorithms based on this approach itself. In the next section we cover some of the transform domain steganographic algorithms.

2.1.2 Transform Domain

These techniques try to encode 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). By being embedded in the transform domain, the hidden data resides in more robust areas, spread across the entire image, and provides better resistance against signal processing. For example, we can perform a block DCT and, depending on payload

and robustness requirements, choose one or more components in each block to form a new data group that, in turn, is pseudo randomly scrambled and undergoes a second-layer transformation. Modification is then carried out on the double transform domain coefficients using various schemes. These techniques have high embedding and extraction complexity. Because of the robustness properties of transform domain embedding, these techniques are generally more applicable to the "Watermarking" aspect of data hiding. Many steganographic techniques in these domain have been inspired from their watermarking counterparts.

F5 [14] uses the Discrete Cosine Transform coefficients of an image for embedding data bits. F5 embeds data in the DCT coefficients by rounding the quantized coefficients to the nearest data bit. It also uses Matrix Encoding for reducing the embedded noise in the signal. F5 is one the most popular embedding schemes in DCT domain steganography, though it has been successfully broken in [26]. The transform domain embedding does not necessarily mean generating the transform coefficients on a blocks of size 8 x 8 as done in JPEG compression techniques. It is possible to design techniques which take the transforms on the whole image [10]. Other block based JPEG domain and wavelet based embedding algorithms have been proposed in [11] and [16] respectively.

2.2 Existing Attacks

The steganalytic attacks developed till date can be classified into visual and statistical attacks. The statistical attacks can further be classified as

- Targeted Attacks
- Blind Attacks

Each of these classes of attack 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 keeping a particular steganographic algorithm in mind. These attacks are based on the image features which get modified by a particular kind of steganographic embedding. A particular steganographic algorithm imposes a specific kind of behaviour on the

image features. 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:**

   The histogram analysis method exploits the asymmetry introduced by LSB replacement. The main idea is to look for statistical artifacts of embedding in the histogram of a given image. 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. 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 hypothesis:

   Null Hypothesis H0: The given image contains steganography embedding

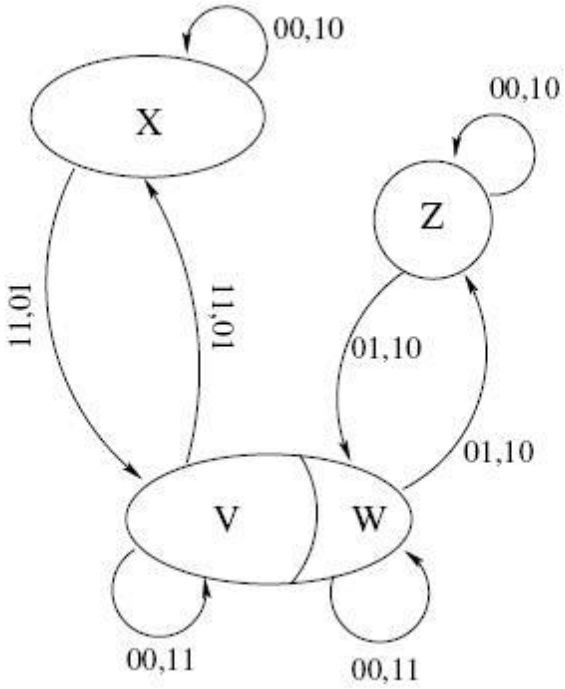


Figure 2.1: Flipping of set cardinalities during embedding

Alternative HypothesisH1: The given image does not contain steganography embedding

The decision to accept or reject the Null Hypothesis H0 is made on basis of the observed confidence value p. A more detailed discussion on Histogram Analysis can be found in [22].

2. **Sample Pair Analysis :**

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. It can precisely measure the length of the embedded message, even when the hidden message is very short relative to the image size. 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. A detailed analysis on Sample Pair technique can be found in [20]. Another attack called RS Steganalysis based on the same concept has been independently proposed by [23].

3. **HCF-COM based Attack:**

This attack first proposed by [27] is based on the Center of Mass (COM) of the Histogram Characteristic Function (HCF) of an image. This attack was further extended for LSB Matching by [24]. 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 $(C(H_{\hat{C}})/C(H_{\hat{S}}))$. It has been proved empirically that:

$$C(H_C) \approx C(H_{\hat{C}}) \tag{2.1}$$

$$C(H_C) - C(H_S) > C(H_{\hat{C}}) - C(H_{\hat{S}}) \tag{2.2}$$

From Equations 2.1 and 2.2, a dimensionless discriminator for classification can be obtained as $(C(H_c)/C(H_S))$. By estimating suitable threshold values of the discriminator from a set of training data, an image can be classified either as cover or stego. Some other targeted attacks also exist in literatures which have not been covered in this survey. A detailed survey can be found in [21].

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. 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. 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. This estimation is sometimes referred to as "Cover Image Prediction". Some of the most popular blind attacks are defined next.

1. **Wavelet Moment Analysis (WAM):**

   Wavelet Moment Analyzer (WAM) is the most popular Blind Steganalyzer for Spatial Domain Embedding. It has been proposed by [25]. 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 filtering is performed in the

wavelet domain, all the features (statistical moments) are calculated as higher order moments of the noise residual in the wavelet domain. The detailed procedure for calculating the WAM features in a gray scale image can be found in [25]. 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.

2. **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 [13] and are designed for JPEG domain steganographic schemes. They estimate the cover image statistics by a process termed as Self Calibration. The steganalysis algorithms based on this self-calibration process can detect the presence of steganography noise with almost 100% accuracy even for very low embedding rates [13, 17]. 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 recompression produce a "calibrated" image with most macroscopic features similar to the original cover image. The process of cropping by 4 pixels is an important step because the 8 x 8 grid of recompression "does not see" the previous JPEG compression and thus the obtained DCT coefficients are not influenced by previous quantization (and embedding) in the DCT domain.

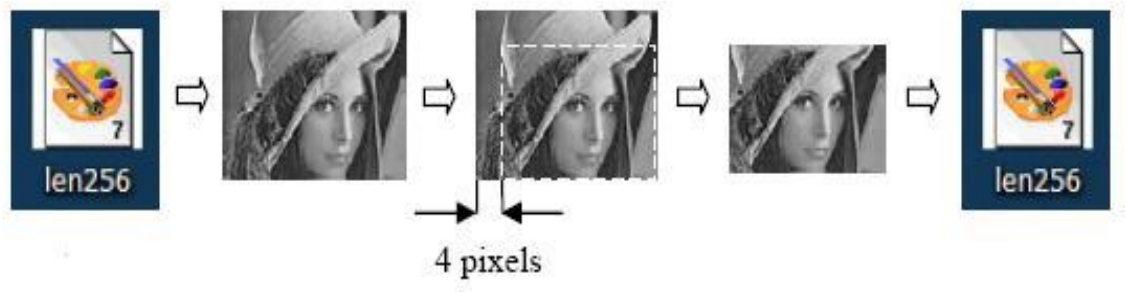


Figure 2.2: Calibration of the stego-image for cover statistics estimation

3. **Farid's Wavelet Based Attack:**

   This attack was one of the first blind attacks to be proposed in steganographic research [12] for JPEG domain steganography. It is based on the features drawn from the wavelet coefficients of an image. 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 X (n - 1) coefficients. 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 X (n - 1) dimensional feature vector. 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 at [19]. After extraction of features, a Support Vector Machine (SVM) is used for classification. We would like to mention that although in [19] a SVM has been used for classification we have used the Linear Discriminant Analysis for classification.

2.3 Evaluation of Image Quality

For comparing Stego images with cover results it requires a measure of image quality. Commonly used measures are Mean-Squared Error, Peak Signal-to-Noise Ratio and histogram.

2.3.1 Mean Square Error (MSE)

The mean squared error (MSE) of an estimator is one of many ways to quantify the difference between values implied by an estimator and the true values of the quantity being estimated. MSE is a risk function, corresponding to the expected value of the squared error

loss or quadratic loss. The mean-squared error (MSE) between two images A(x y) and B(x y) is:

$$MSE = \sum_{x=1}^{p} \sum_{y=1}^{q} \frac{\left(\left| M_{xy} - N_{xy} \right| \right)^2}{p * q}$$

where p and q are the width and height of the image.

2.3.2 Peak Signal-to-Noise Ratio (PSNR)

As a performance measurement for image distortion, the wellknown Peak-Signal-to-Noise Ratio (PSNR) which is classified under the difference distortion metrices can be applied on the stego images. It is defined as:

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

where, $C_{\max}^2$ holds the maximum value in the image.

For example:

$$C_{max}^2 \leq \begin{cases} 1 \text{ in double precision intensity images} \\ 255 \text{ in } 8-\text{bit unsigned integer intensity images} \end{cases}$$

CHAPTER 3: RESEARCH METHODOLOGIES

In this chapter, we provide the necessary background required for the proposed algorithm. The proposed algorithm uses Huffman Compression of data, private key substitution cryptography, hamming code for noise disturbance and already proposed technique for embedding of data in the image which uses least significant bit as only status bit, not for storing data.

3.1 A Novel Steganography Method for Image Based on Huffman Encoding

Hiding the secret image/message in the spatial domain can easily be extracted by unauthorized user and in the frequency domain the quality of the extracted secret image deteriorates. In this paper we proposed a spatial domain steganographic technique based on Huffman encoding for hiding a large amount of data with high security, good invisibility and no loss of secret message. The schematic block diagram of the whole process is given in Fig. 3.1 and Fig. 3.2.

In [29] this technique is proposed. The main objective in here is to develop a procedure which will provide a better security to the secret image without compromising on the quality of the stego image. Our algorithm has three main parts. First, it embeds the Huffman encoded bit stream of the secret image into the cover image. Second, it embeds the size of the encoded bit stream into the cover image. Third, it also embeds the Huffman table corresponding to the secret image into the cover image.

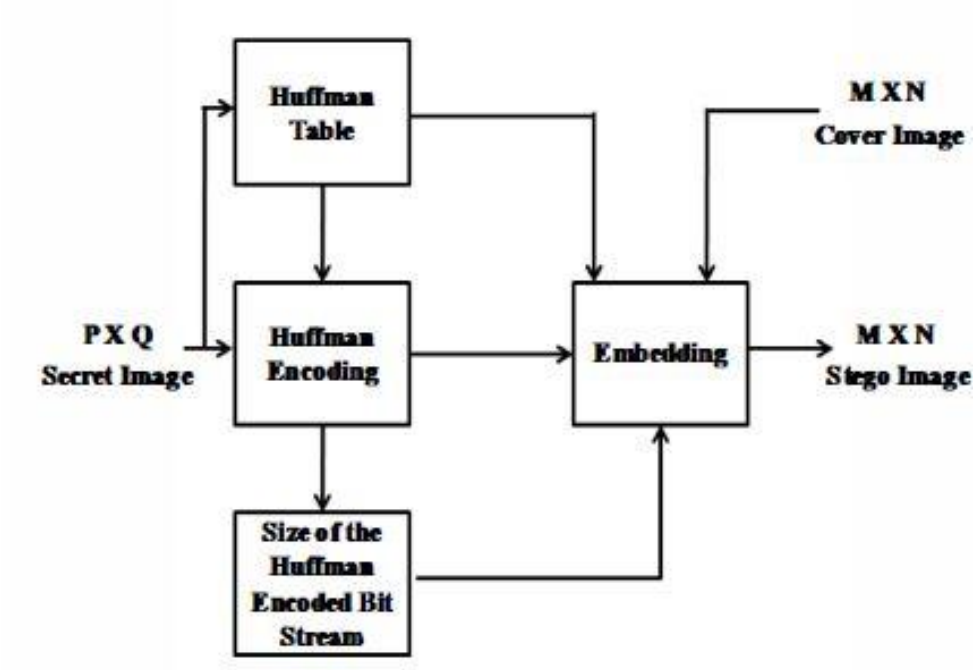


Figure 3.1: Insertion of the Secret Image/Message into a Cover Image

3.1.1 Four Tier Storage Procedure of "Size of Huffman Encoded Bit Stream"

The size of the Huffman encoded bit stream needs to be embedded inside the cover image to let the decoder know till which pixel's LSB holds the Huffman encoded bit stream. Now the question is how will we get the size of the "Size of Huffman encoded bit stream"? If we store this size of the "Size of Huffman encoded bit stream" then also the same question comes recurrently like what is the size of this size? As a permanent answer to this question we have used a four tier storage procedure to store the "size of Huffman encoded bit stream". In the first tier we find the size of Huffman encoded bit stream and store it into a variable e.g. A. Then convert A into binary format and find the size of the binary bit stream of A and store it into another variable e.g. B (second tier). Similarly in the third and fourth tier, we find the size of the bit stream of B and store it into another variable e.g. C and again find the size of the bit stream of C and store it into another variable e.g. D, respectively. Finally D will have a value that can be represented by two binary bits only. Now these two binary bits can be stored in the first or in the last pixel's two LSB positions of the first 8 X 8 block of the cover

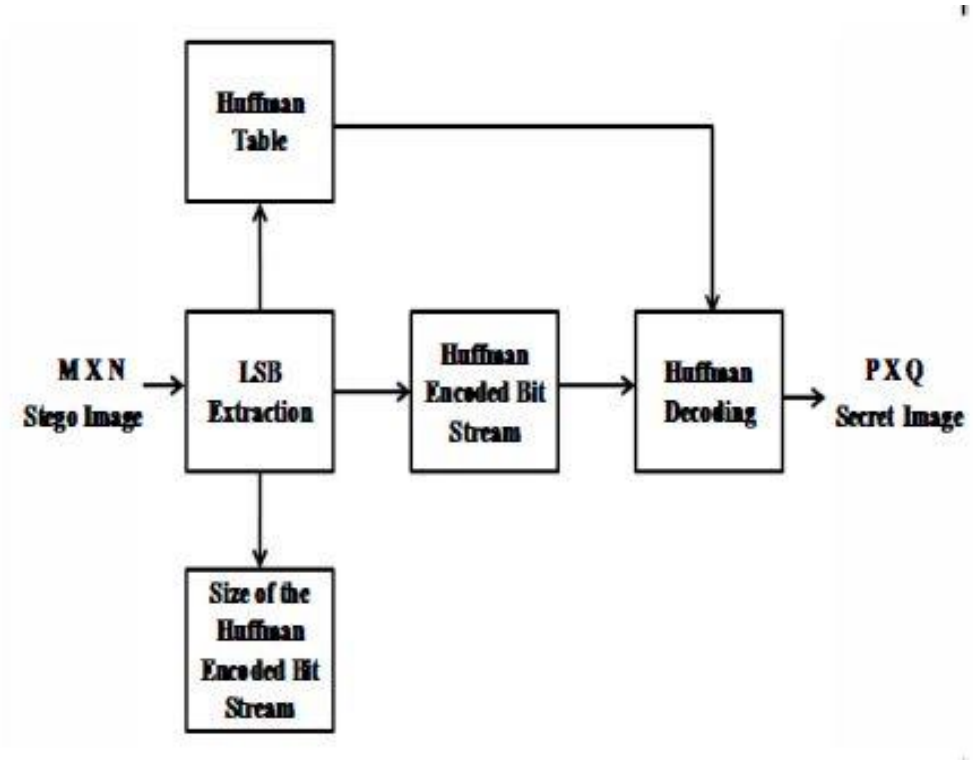


Figure 3.2: Extraction of the Secret Image from the Stego Image

image. The variables C, B, A also needs to be stored into the cover image one by one sequentially. E.g. of four tier storage of the "size of Huffman encoded bit stream", let us say the size of a Huffman encoded bit stream is 461690 so,

A = 461690 = 1110000101101111010, which is of size 19 bits.

B = 19 = 10011, which is of size 5 bits.

C = 5 = 101, which is of size 3 bits.

D = 3 = 11, which is of size 2 bits

This four tier storage procedure is very important because the size of the Huffman encoded bit stream comes down to 2 bits from 19 bits. Variables are stored from D to C, B, A sequentially. At the time of LSB extraction first variable D is extracted then C to B to A to finally find the actual size of Huffman encoded bit stream.

3.1.2 Proposed Algorithm for Embedding and Extraction of the Secret Image

3.1.2.1 Embedding Algorithm

Input: An M X N carrier image and a P X Q secret message/image.

Output: An M X N stego-image.

1. Read both the Cover Image and the Secret Image and store their intensity value of different pixels in two different arrays.
2. Calculate the size of the Secret Image. The size of the Secret Image multiplied by 8 (for 8 bit images) should be less than the size of the Cover Image. E.g. if the Secret Image size is 256 x 256 = 65536 then after multiplying it by 8 it becomes 523288. This is lesser than the size of Cover Image i.e. 1024 x 1024 = 1048576.
3. Obtain Huffman table of secret message/image.
4. Find the Huffman encoded binary bit stream of secret-image by applying Huffman encoding technique using Huffman table obtained in Step-3.
5. Calculate size of Huffman encoded bit stream.
6. Store the size found in Step-5 using the four tier storage procedure described above by modifying the LSB of pixels of fITst 8 X 8 block of the cover image.
7. Change the LSBs of the cover image excluding the first 8 X 8 block of pixels for every bit of Huffman encoded bit stream found in Step-4.

8. Change the LSBs of the cover image to embed the Huffman table found in Step-3 excluding the first 8 X 8 block of pixels and the pixels used in Step-7.

9. Write the new Stego Image into the disk.

10. End.

3.1.2.2 Extraction Algorithm

Input: An M X N stego-image**.**

Output: A P X Q secret image.

1. Read the Stego Image and extract the size of the Huffman encoded bit stream from the first 8 X 8 block by extracting the LSB of the pixels using the procedure described in the four tier storage method like variable D to C to B to A.
2. Using the size found in Step-l extract the Huffman encoded bit stream by extracting the LSB of pixels excluding first 8 X 8 block and add it into an array.
3. Construct the Huffman table by extracting the LSB of pixels excluding the first 8 X 8 block of pixels and the pixels used at Step-2.
4. Decode the array obtained in Step-2 using the Huffman table obtained in Step-3 to extract the secret image from the stego image.

3.2 Symmetry Key Cryptography

Cryptography is a method in which a message is obfuscate and sent in a deficient composition. Cryptography ensures message integrity.

In the proposed algorithm, we used transposition cryptography. In this, a private key is shared on both sender's and receiver's sides.

Let us take an example for understanding the method of transposition.

Suppose we have a matrix 'A' of a string in bits and the length of the string is 96 bits long,

A = [1 0 1 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 1 0 1 0 0 1 1 1 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 1 0 1 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 1 0 1 0 0 1 1 1 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1]

And the private key for transposition of length k is

P = [5 9 12 6 1 7 11 3 4 10 8 2] (k=12)

Now, Matrix A is divided into the sets of length k and the passed in to the function of transposition.

The positions are shuffled according to the private key, and we get the cipher matrix as result. A reverse key is generated by using the private key for deciphering the ciphered matrix.

The private key is shared by the sender and receiver through a secure channel.

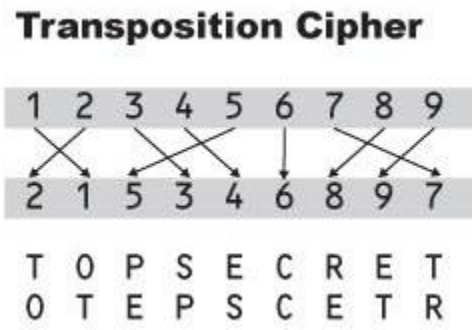


Figure 3.3: Method of transposition of positions

3.3 A Highly Secure Video Steganography using Hamming Code (7, 4)

The Hamming code is one of the most well-known block code methods that can do both error detection and correction on a block of data. In the Hamming code technique, the original information will be coded by adding some extra data with the minimum amount of redundancy, which is called the code word, of length n bits [28]. The added part consists of parity information of length (n-k) bits where k is the length of message that is expected to be coded [28]. In this paper, the (7, 4) Hamming code is used that can detect and correct a single bit error of data or parity. First, the message (m1, m2, m3, m4) of length k bits (k=4) is encoded by adding three parity bits (p1, p2, p3) to become the code word of length n (n=7), which is ready for transmission. There are different ways to mix both types of data (message and parity) together and the general combination is to put the parity bits at position 2i such as (p1, p2, m1, p3, m2, m3, m4) where I = 0, 1, ... ,(n-k-1).

The Hamming codes are linear codes so they have two matrices: parity-check matrix H and generator matrix G, which they need for both encoding and decoding. On the encoding side, each message M, which consists of 4-bits, will be multiplied by the generator matrix and then have modulo of 2 applied; the result is the code word X of 7-bits ready to be sent through a noisy channel.

$$X = M \times G \text{ Where } G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

On the decoding side, for the purpose of checking the encoded message of 7-bits R (data + parity) will be received, and then will be multiplied by the transpose of the parity-check matrix, and taking modulo of 2 again.

$$Z = R \times H^T \text{, where } H^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

The result is a syndrome vector Z (z1, z2, z3) of three bits, which has to be all zeroes (000) if it's an error-free message. Otherwise, any change in the message during transmission will lead to flipping one or more bits of the message; then it needs an error correction process.

Example: Assume we have a message M1 of 4-bits (1, 1, 1, 1) and the Hamming code (7, 4) is done by the following steps:

1. Calculate, the result is (3, 3, 3, 1, 1, 1, 1) then taking modulo of 2, the result is the code word X=1111111, which is sent through the communication channel.
2. At the destination, to get the correct message the syndrome vector Z must be zero. The first assumption R=1111111 is received without any errors. Then Z will become (0, 0, 0), where $Z = R \times H^T$.
3. In the second assumption, suppose that during transmission due to the noisy channel one of the bits has changed. The received data will be R=1111011, then calculating the syndrome we will get Z=433, and taking modulo of 2 the syndrome will become Z=011.
4. Comparing Z value with the parity-check matrix H, it appears that the Z value (0, 1, 1) is equal to the 5th row (0, 1, 1) of the H matrix, which means that the 5th bit of R has changed.
5. Correcting the 5th bit of R by flipping it to 1, R then is corrected to become (1, 1, 1, 1, 1, 1, 1).
6. The four first bits are the original message M1 (1, 1, 1, 1) and the last three other bits will be ignored.

3.3.1 Data Embedding Phase

Data embedding is a process of hiding a secret message inside host videos, and it can be done by the following steps:

1. Convert the video stream into frames.
2. Separate each frame into Y, U and V components.
3. Change the position of all pixels in three components Y, U and V by a special key.
4. Convert the message (which is a binary image) to a one dimension array, and then change the position of the whole message by a key.
5. Encode each 4 bits of the message using Hamming (7, 4) encoder.
6. The result of the encoded data, which consists of 7 bits (4 bits of message + 3 bits of parity) is XORed with the 7 bits of random value using a key.
7. Embed the result of those 7 bits in one pixel of YUV components (3-bits in Y, 2-bits in U and 2-bits in V).
8. Reposition all pixels of YUV components to the original frame pixel position.
9. Rebuild the video stream again from those embedded frames.

There are three keys that have been used in this work, which give to proposed steganography scheme an improvement in both security and robustness. Those keys are shared between sender and receiver in both data embedding and extracting processes.

The first key is used to reposition pixels in Y, U, V, and the secret message into a random position, which makes the data chaotic. In order to select the locations for embedding the secret

message into the host data, the second and third keys are used. They are used to pick the random rows and columns respectively in each chaotic Y, U and V component. The XOR function that has been used increases the quality of the system.

3.3.2 Data Extracting Phase

Data extracting is a process of retrieving the secret message from the stego videos which can be done by the following steps:

1. Convert the video stream into frames.
2. Separate each frame into Y, U and V components.
3. Change the position of all pixel values in the three Y, U, and V components by the special key that was used in the embedding process.
4. Obtain the encoded data from the YUV components and XOR with the random number using the same key that was used in the sender side.
5. Decode 4 bits of the message by the Hamming decoder.
6. Reposition the whole message again into the original order.
7. Convert the message array to 2-D.

3.4 An Efficient Filtering Based Approach Improving LSB Image Steganography using Status Bit along with AES Cryptography

A digital image consists of different pixels. In this method we used color image. As we know, a colored pixel can be represented as a mixture of red, green and blue color with appropriate proportions. In binary notation, a color level is represented by a stream of 8 bits. Therefore in total, 24 bits are required to denote a pixel. Thus an image is an array of many bytes each representing a single color information lying in a pixel. In the proposed method, a group of three sequential bytes from such an array is used to embed a bit of the entire message. In [30], this technique is proposed.

The proposed technique has two main parts:

1. Changing the secret message (plain text) to cipher text by AES Cryptography
2. Hiding the cipher into image by a proposed Steganographic technique

128 bits AES Cryptographic algorithm takes a password and encrypts the plain text to cipher text. This cipher text will be embedded into a cover image using our Steganographic technique. In the Steganographic technique, a filtering algorithm has been used to hide the information. The MSB bit specify the area where to embed the secret message.

This algorithm has the concept of randomly select an image and find if it is a darker or lighter image. Lighter image means MSB bits of Red, Green, and Blue component of a pixel contain at least 2 bit 1's and darker image means MSB bits of Red, Green, and Blue component of a pixel contain at least 2 bit 0's. If lighter pixel is greater than darker pixel, we select lighter pixel area to embed message and vice versa. Following figure shows this concept.

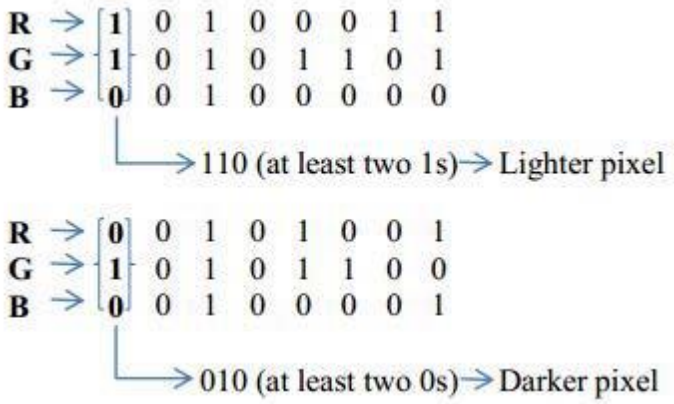


Figure 3.4: Concept of Lighter and darker pixel

It embeds the binary representation of the ASCII value of alphabets, for example, "A" will be embedded in the cover image as "01000001".

First, it embeds the size of the embedding cipher using the first 96 pixels of the cover; here each bit takes only one pixel to hide. After that it uses the resting pixels for embedding the secret cipher (binary representation of ASCII value of each character).

3.4.1 Embedding Process

In a bitmap image, color information is arranged byte by byte as R-G-B-R-G-B-R-…. etc. In the proposed method, a message bit is embedded in the LSB of such a byte. This byte is chosen from a block of spatially adjacent 3 bytes. The choice where the message bit embedded is determine by the MSB of the 3 byte.

In this case there will be two situations:

1. When the image has lighter area than darker area

2. When the image has darker area than lighter area

3.4.1.1 Embedding Message in Lighter Area

If the MSB bits contain at least two bit 1, then this pixel is selected for hiding the message bit. The decimal representations Pn of the 3 MSB bits are used for validating the LSB bit. If the message bit Mn and the bit position Pn of 3rd byte is same, that means true, then change the LSB of 3rd byte into 1 or 0, otherwise. In this approach the message bit is embedded into the cover image. Following table shows that which pixels are used to embed the data.

Pixel	Color Component	Component Value	MSB (3Bit)	Message Bit	Resulting Color Value
1st	R	00001111	$011_2 = 3_{10}$	1	00001111
	G	10000101			00000101
	B	10000000			10000000
2nd	R	00001010	$010_2 = 2_{10}$	skip	00001010
	G	10011100			10011100
	B	01001110			01001110
3rd	R	11010000	$110_2 = 6_{10}$	0	11010000
	G	10001111			10001111
	B	01001110			01001110
4th	R	01001010	$011_2=3_{10}$	0	01001010
	G	10001101			10001101
	B	10110000			10110001
5th	R	10001011	$110_2=6_{10}$	1	10001011
	G	10111101			10111101
	B	01001110			01001111
6th	R	00001111	$001_2 = 1_{10}$	skip	00001111
	G	01101011			11101011
	B	11001000			01001000

Table 3.1: EMBEDDING A MESSAGE IN LIGHTER PIXEL

3.4.1.2 Embedding Message in Darker Area

If the MSB bits contain at least two bit 0, then this pixel is selected for hiding message bit. The decimal representations Pn of the 3 MSB bits are used for validating the LSB bit. If the message bit Mn and the bit position Pn of 3rd byte is same, that means true, then change the LSB of 3rd byte into 1 or 0, otherwise. Here there is one more condition that is if all 3 bits of MSB are 0, then decimal value of MSB bits is 0. In this situation message bit will be directly inserted into the Blue color component. In this approach the message bit is embedded into the cover image. Following table shows that which pixels are used to embed the data.

Pixel	Color Component	Component Value	MSB (3Bit)	Message Bit	Resulting Color Value
1st	R	00001111	$010_2 = 2_{10}$	1	00001111
	G	10000101			10000101
	B	00000000			00000000
2nd	R	10001010	6_{10}=110	skip	10001010
	G	10011100			10011100
	B	01001110			01001110
3rd	R	11010000	$100_2 = 4_{10}$	0	11010000
	G	00001111			00001111
	B	01001110			01001111
4th	R	01001010	1_{10}=001_2	0	01001010
	G	00001101			00001101
	B	10110000			10110001
5th	R	00001011	0_{10}=000_2	1	00001011
	G	00111101			00111101
	B	01001110			01001111
6th	R	10001111	$101_2 = 1_{10}$	skip	10001111
	G	01101011			01101011
	B	11001000			11001000

Table 3.2: EMBEDDING A MESSAGE IN DARKER PIXEL

3.4.1.3 Embedding Algorithm

6. Get the Original message
7. Encrypt this original message with AES encryption technique
8. Convert the AES cipher into binary number
9. Get the cover image
10. Check the image whether it is a lighter or darker image
11. Collect the MSB bits from a pixel (Red, Green, Blue color component)
12. From the MSBs, for lighter image if it contains two bits 1 and for the darker image if it contains two bits 0, select this pixel for hiding message bit. Otherwise, skip this pixel.
13. Convert MSB into decimal number Pn.
14. If Pn=0 for the darker image only embed the message bit into the Blue color component of the pixel.
15. For lighter image or other value of Pn:
    a. Check Pn bit position of the Blue color component with message bit
    b. If it matches then change the LSB of Blue color component with 1(indicate status true)
    c. If it does not match with message bit then change the LSB of Blue color component with 0(indicate status false).

3.4.2 Extracting Process

At first collect the Stego image and arrange it into byte. Check the color whether it is a lighter or darker image.

3.4.2.1 Extracting Message from Lighter Pixel

First, collect the 3 byte of a pixel. Then collect the MSB bit of the 3 byte and check whether it contain at least 2 bits with 1. This indicates the selected bit for extracting message bit. Then the decimal representation Pn of the 3 MSB bits is taken. The LSB bits of the 3rd byte of the selected pixel indicate the cipher message bit. If it is 0 that's mean false then the bit position Pn of 3rd byte checked. If it is 0 extract message bit 1 or 0, otherwise. If it is 1 that's mean true then the bit position Pn of 3rd byte checked and extract the message bit as same as the Pn bit of the 3rd byte. This approach will be continued until all the message bit is extracted from the stego image. Following table illustrate how data is extracted from the pixels of the image.

Pixel	Color Component	Component Value	MSB (3 Bit)	Extracted Message Bit
1st	R	00001111	$011_2 = 3_{10}$	1
	G	10000101		
	B	10000000		
2nd	R	00001010	2=010	skip
	G	10011100		
	B	01001110		
3rd	R	11010000	$110_2 = 6_{10}$	0
	G	10001111		
	B	01001110		
5th	R	10001011	$6_{10}=110_2$	1
	G	10111101		
	B	01001111		

Table 3.3: EXTRACTING MESSAGE FROM LIGHTER PIXEL

3.4.2.2 Extracting Message from Darker Pixel

Collect the 3 byte of a pixel. Then collect the MSB bits of the 3 byte and check whether it contains at least 2 bit with 0.This indicates the selected bit for extracting message bit. Then the decimal representation Pn of the 3 MSB bits is taken. If Pn = 0, we simply collect the message bit from the LSB bit of the 3rd byte. Otherwise the LSB bit of the 3rd byte of the selected pixel indicates the cipher message bit. If it is 0 that's mean false, then the bit position Pn of 3rd byte is checked and if it is 0, extract message bit 1 or 0, otherwise. If it is 1 that's mean true then the bit position Pn of 3rd byte is checked and extract the message bit as same as the Pn bit of the 3rd byte. This approach will be continued until all the message bit is extracted from the stego image. The received AES cipher text is then decrypted to get the original message. Following table illustrate how data is extracted from the pixels of the image.

Pixel	Color Component	Component Value	MSB (3 Bit)	Extracted Message Bit
1st	R	00001111	$010_2 = 2_{10}$	1
	G	10000101		
	B	00000000		
2nd	R	10001010	6_{10}=110	skip
	G	10011100		
	B	01001110		
3rd	R	11010000	$100_2 = 4_{10}$	0
	G	00001111		
	B	01001111		
5th	R	01001010	1_{10}=001_2	1
	G	00001101		
	B	10110001		

Table 3.4: EXTRACTING MESSAGE FROM DARKER PIXEL

3.4.2.3 Extracting Algorithm

1. Get the Stego image.
2. Check the Stego image as lighter or darker.
3. Collect the MSB bits from a pixel (Red, Green, Blue color component).
4. From the MSBs, for lighter image if it contains two bits 1 and for the darker image or if it contains two bits 0, select this pixel for hiding message bit. Otherwise, skip this pixel.
5. Convert MSB into decimal number, Pn.
6. If Pn=0 only get the cipher binary from Blue component of the pixel.
7. If Pn>0, Check the LSB (Status bit) whether it is 0 or 1.
8. If the LSB equals to 0, then collect the cipher binary by toggling the Pn bit of the Blue component of the pixel.
9. If the LSB equals to 1, then collect the cipher binary as the Pn bit of the Blue component of the pixel.
10. Apply AES to decrypt the original message from cipher text.

This proposed algorithm changes very small number of bits when embedding a large cipher. As the algorithm can use all the pixels to hide data, a 512x512 size cover image can hide at most 4000 character which is of course a big cipher. For a larger cover image and larger cipher, one can use more than 50 pixels to hide the cipher size. Moreover, it is not necessary to work with LSBs of the color components. The proposed technique can be applied using any bit position which makes more difficult to retrieve the cipher by Stegoanalyst.

CHAPTER 4: Proposed Work

Hiding the secret image/message in the spatial domain can easily be extracted by unauthorized user and in the frequency domain the quality of the extracted secret image deteriorates.

In this thesis, I proposed a spatial domain steganographic technique based on Huffman encoding, private key encryption, and hamming code for hiding a large amount of data with high security, good invisibility and no loss of secret message.

The main objective in here is to develop a procedure which will provide a better security to the secret image without compromising on the quality of the stego image.

This proposed algorithm has four main parts:

1. **Huffman Encoding**

   First, it compresses the message using Huffman compression. From this we get a compressed code word and Huffman table which consists of the unique symbols with their probabilities.

   Then, it combines the length of the code word, code word and Huffman tables to get a new string.

   Let us take an example,

   We have a secret message say M of length L.

   For example, M = this is a secret message to be embedded

   Then we pass this secret message (M) to the Huffman encoding function which compresses the secret message and returns the resulting code word in bits. And, also

we get Huffman table from Huffman encoding which consists of unique symbols with their probabilities of occurrence in the secret message.

Now, we have a code word say C, and Huffman table say T.
C is a string of bits like [1 1 1 0 0 1 0 1 0 1].

Huffman encodes compresses the data to approximately 55%, and helps to embed large amount of data in small area.

2. Symmetry Key Encryption

In this step, the resulted code word is divided in to the block of length 26. There is a secret key for encryption which is shared between the sender and receiver of length 26. Secret key is stores the shuffled positions for a block of length 26.

For example,
K = [15,12,26,2,10,21,11,23,4,1,3,20,18,22,25,13,19,5,24,9,6,8,17,14,7,16]

The code word is divided in to the blocks of length 26 and on each block transposition is applied using key K.
After applying transposition, we get the cipher text.
The advantage of encryption is that the message is not readable to the trespasser.

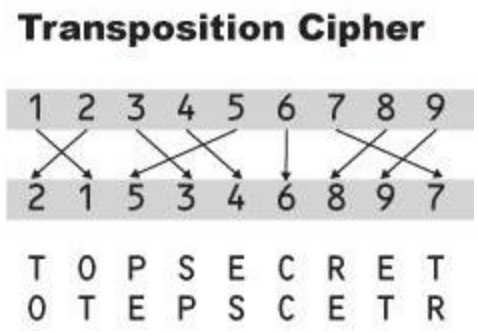


Figure 4.1: Transposition Ciphering

The resultant cipher text is of same length as of the code word. But the positions are shuffled in the blocks of length 26.

3. Hamming Code (n , k)

The Hamming code is one of the most well-known block code methods that can do both error detection and correction on a block of data. In the Hamming code technique, the original information will be coded by adding some extra data with the minimum amount of redundancy, which is called the code word, of length n bits. The added part consists of parity information of length (n-k) bits where k is the length of message that is expected to be coded.

General terms used in hamming code:

For each integer $r \geq 2$,

Code with block length $n = 2^r - 1$, and

Message length $k = 2^r - r - 1$

Rate of Hamming codes is $R = k / n = 1 - r / (2^r - 1)$

In the proposed algorithm, the hamming code used has values:

r = 5,

n = 31,

k = 26,

R = 26/31 = 0.8387

So, the hamming code (31, 26) is used in this algorithm.

The biggest advantage of hamming code (31, 26) is that it can find two bit error and correct one bit error.

The cipher text is divided into blocks of length 26 and passes in to the hamming code function. For a 26 length block, we get 31 length block of message. So, we are adding 5 bits in a block of length 26. These bits are added at the power of 2 positions with the help of generator matrix. An example of generator matrix is given in the figure 4.2, but our generator matrix has 25 rows and 31 columns.

$$X = M \times G \text{ Where } G = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 4.2: Generator Matrix

4. Filtering Based LSB Image Steganography using Status Bit

There are two types of digital images, lighter images and darker images. If the image has more number of lighter pixels than darker pixels then it is known as lighter image, otherwise it is known as darker image.

If the image is lighter image than we embed the secret message only in the lighter pixels, but if image is darker image then we embed the secret message only in the darker pixels.

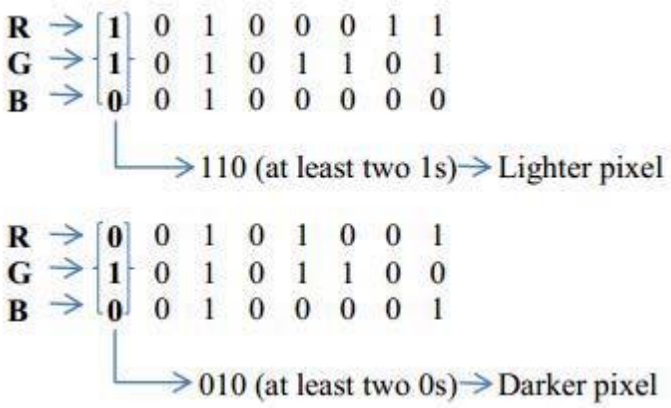


Figure 4.3: Lighter pixels and darker pixels

So, we are filtering the pixels of the image for embedding the secret information according to the lighter image or darker image.

The data is embedded into the pixels of an image at any bit position which is calculated dynamically from the most significant bits of the red, green, and blue band. The most significant bits of red, green, and blue pixels are fetched and a position is calculated as described below.

For example,

MSB of Red is r, MSB of Green if g, and MSB of Blue is b.

Then, Bit position = 4*r + 2*g + b

If the bit value at that bit position matches with the data bit then we set the least significant bit to 1, but if data does not match then set the least significant bit to 0. Here, we use least significant bit as a status bit, if data match then set to 1, otherwise set to 0.

In this method, we are embedding the data at a dynamic position but changes are made only in the least significant bit. So, maximum change in the pixel value is 1.

Following table shows the data embedded in the lighter pixels.

Pixel	Color Component	Component Value	MSB (3Bit)	Message Bit	Resulting Color Value
1st	R	00001111	$011_2 = 3_{10}$	1	00001111
	G	10000101			00000101
	B	10000000			10000000
2nd	R	00001010	$010_2 = 2_{10}$	skip	00001010
	G	10011100			10011100
	B	01001110			01001110
3rd	R	11010000	$110_2 = 6_{10}$	0	11010000
	G	10001111			10001111
	B	01001110			01001110
4th	R	01001010	$011_2=3_{10}$	0	01001010
	G	10001101			10001101
	B	10110000			10110001
5th	R	10001011	$110_2=6_{10}$	1	10001011
	G	10111101			10111101
	B	01001110			01001111
6th	R	00001111	$001_2 = 1_{10}$	skip	00001111
	G	01101011			11101011
	B	11001000			01001000

Table 4.1: Data embed in lighter pixels

Following table shows the data embedded in the darker pixels.

Pixel	Color Component	Component Value	MSB (3Bit)	Message Bit	Resulting Color Value
1st	R	00001111	$010_2 = 2_{10}$	1	00001111
	G	10000101			10000101
	B	00000000			00000000
2nd	R	10001010	6_{10}=110	skip	10001010
	G	10011100			10011100
	B	01001110			01001110
3rd	R	11010000	$100_2 = 4_{10}$	0	11010000
	G	00001111			00001111
	B	01001110			01001111
4th	R	01001010	1_{10}=001_2	0	01001010
	G	00001101			00001101
	B	10110000			10110001
5th	R	00001011	0_{10}=000_2	1	00001011
	G	00111101			00111101
	B	01001110			01001111
6th	R	10001111	$101_2 = 1_{10}$	skip	10001111
	G	01101011			01101011
	B	11001000			11001000

Table 4.2: Data embed in darker pixels

4.1 Embedding Algorithm

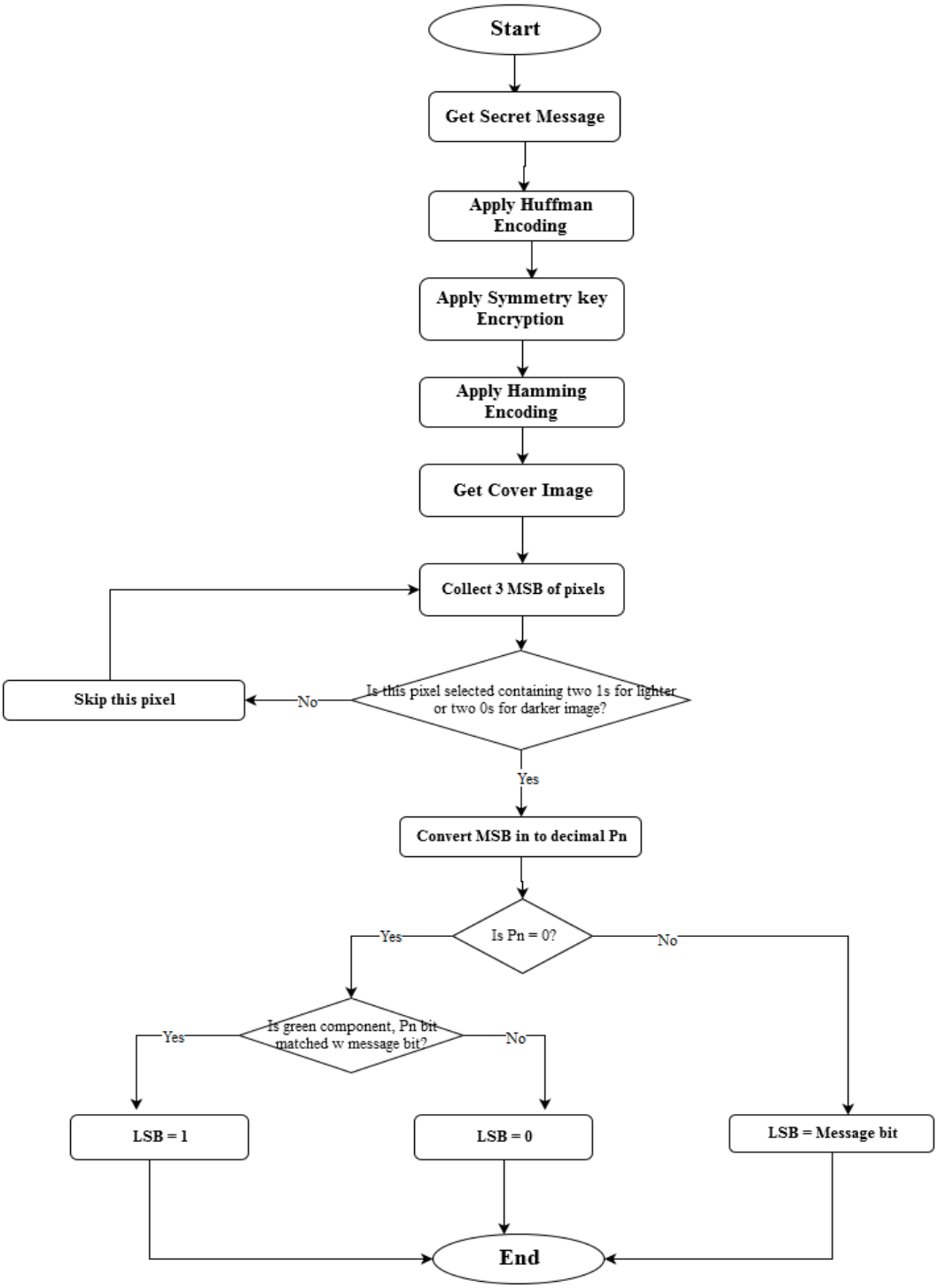
Input: An M X N carrier image and a secret message.

Output: An M X N stego-image.

1. Get the Original message.
2. Compress the message using Huffman encoding.
3. Now apply symmetry key encryption.
4. Now on the cipher text apply hamming code (31, 26).
5. Pass the resultant code word for embedding into the image.

6. Get the cover image
7. Check the image whether it is a lighter or darker image
8. Collect the MSB bits from a pixel (Red, Green, Blue color component)
9. From the MSBs, for lighter image if it contains two bits 1 and for the darker image if it contains two bits 0, select this pixel for hiding message bit. Otherwise, skip this pixel.
10. Convert MSB into decimal number Pn.
11. If Pn=0 for the darker image only embed the message bit into the Blue color component of the pixel.
12. For lighter image or other value of Pn:
    a. Check Pn bit position of the Blue color component with message bit
    b. If it matches then change the LSB of Blue color component with 1(indicate status true)
    c. If it does not match with message bit then change the LSB of Blue color

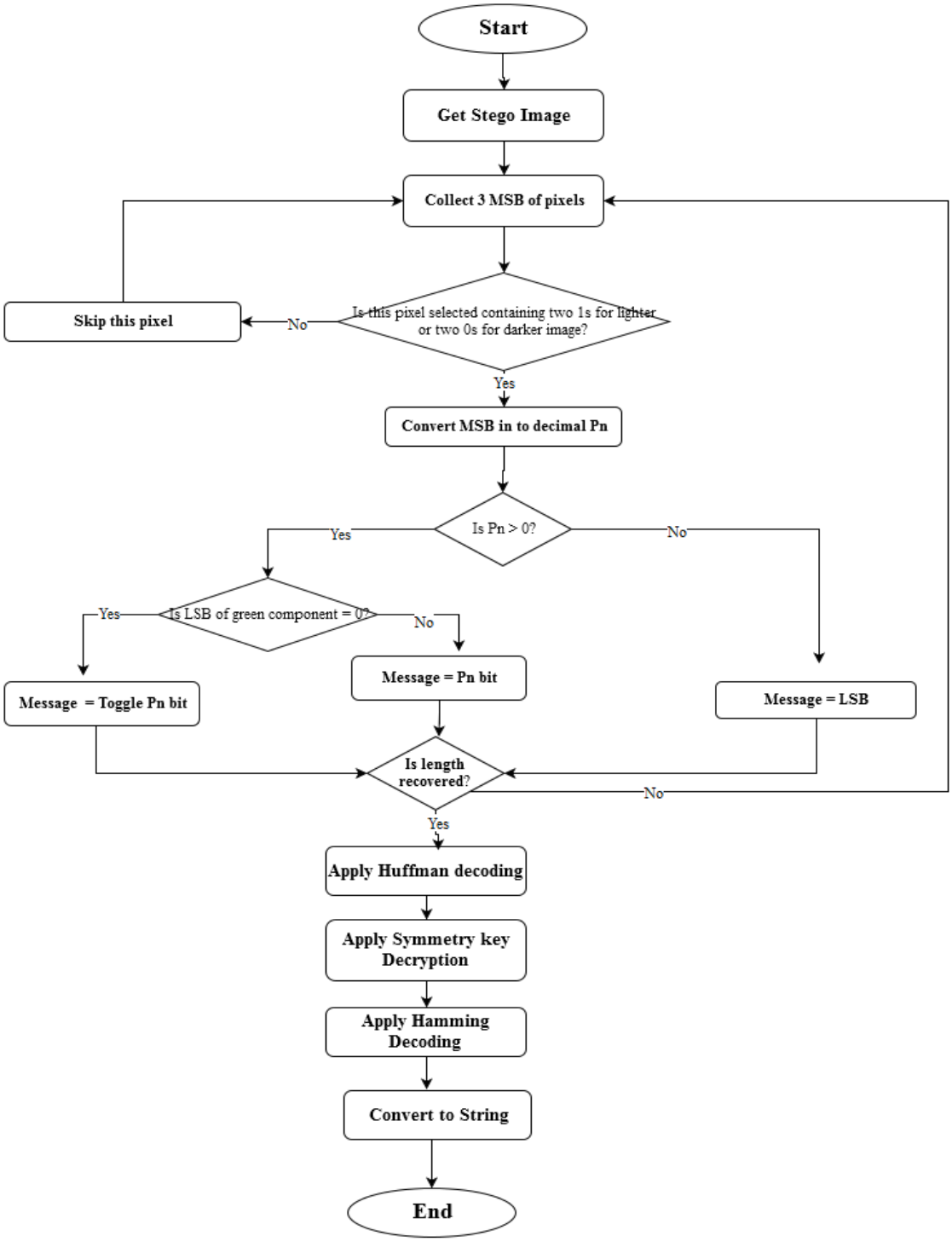
4.2 Embedding Flowchart



4.3 Extracting Algorithm

1. Get the Stego image.
2. Check the Stego image as lighter or darker.
3. Collect the MSB bits from a pixel (Red, Green, Blue color component).
4. From the MSBs, for lighter image if it contains two bits 1 and for the darker image or if it contains two bits 0, select this pixel for hiding message bit. Otherwise, skip this pixel.
5. Convert MSB into decimal number, Pn.
6. If Pn=0 only get the cipher binary from Blue component of the pixel.
7. If Pn>0, Check the LSB (Status bit) whether it is 0 or 1.
8. If the LSB equals to 0, then collect the cipher binary by toggling the Pn bit of the Blue component of the pixel.
9. If the LSB equals to 1, then collect the cipher binary as the Pn bit of the Blue component of the pixel.
10. Apply Hamming Decoding on the fetched codeword.
11. Apply symmetry key decryption.
12. Apply Huffman Decoding on plain text.
13. Print secret message.

4.4 Extracting Flowchart

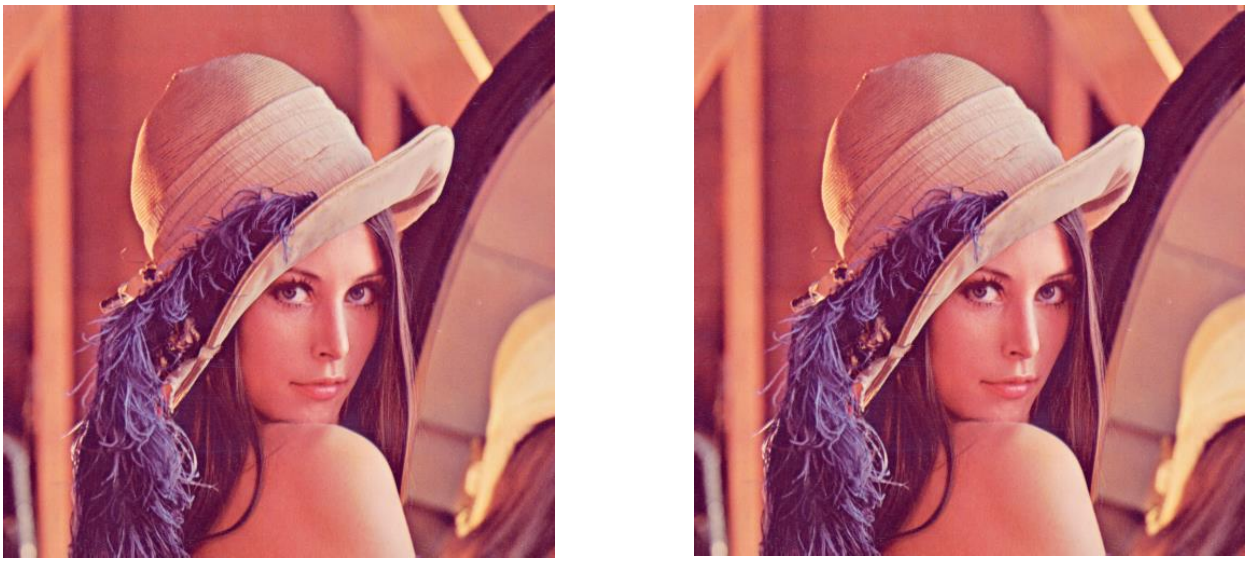


CHAPTER 5: Result

A 24-bit image namely Lenna.png as shown in Figure 5.1 was used as the cover image. The outputs of the program run were remarkably similar to the original images. The message hidden in the image is given in Table 5.1, and the histograms are shown in Figure 5.2. It is seen that the proposed sterilization technique does not detectably distort the histogram of the cover image.

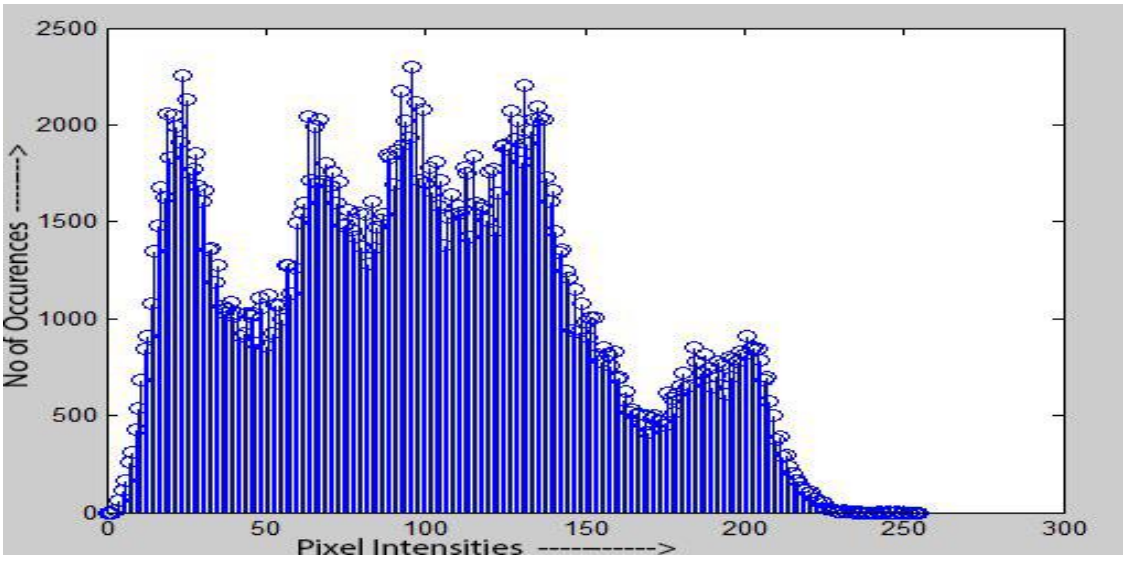
Bitmap	Image Size	Original Message	Size of Message (in bits)	Compressed Size (in bits)	Embedding Size (in bits)
Lenna.png	463KB	Facebook (stylized as facebook) is a for-profit corporation and online social networking service based in Menlo Park, California, United States. Its website was launched on February 4, 2004 by Mark Zuckerberg with his Harvard College roommates and fellow students Eduardo Saverin, Andrew McCollum, Dustin Moskovitz, and Chris Hughes.[7][8][9] The founders had initially limited the website's membership to Harvard students, but later expanded it to higher education institutions in the Boston area, the Ivy League, and Stanford University. It gradually added support for students at various other universities and later to high school students. Since 2006, anyone in general aged 13 and older has been allowed to become a registered user of the website, though variations exist in the minimum age requirement, depending on applicable local laws.[10] Its name comes from the face book directories often given to U.S. university students.	7488	4107	5133

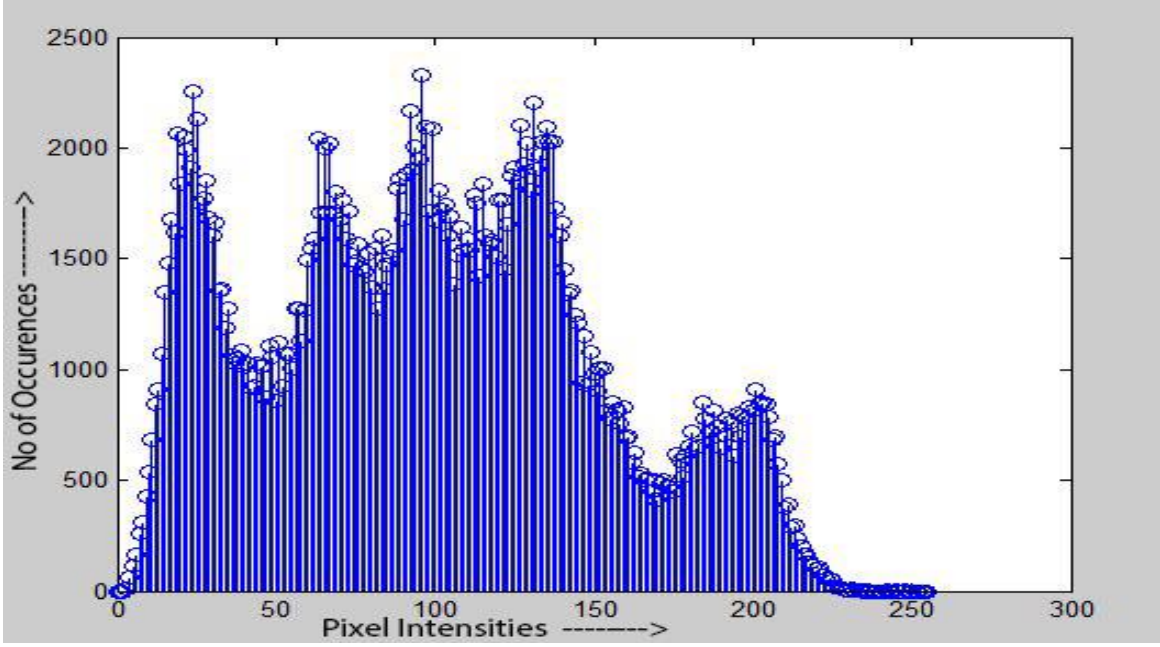
Table 5.1: CONCEALING OF DATA IN A COVER IMAGE



(a) (b)

Figure 5.1: The input (a) and the corresponding output (b) of the program using the proposed technique for hiding data.





(b)

Figure 5.2: (a) The Histogram of input and (b) the corresponding Histogram of output.

The proposed algorithm changes very small number of bits when embedding the message. As the algorithm can use all the pixels to hide data, a 512x512 size cover image can hide at most 4000 character which is of course a big cipher. For a larger cover image and larger cipher, one can use more than 50 pixels to hide the cipher size.

Moreover, it is not necessary to work with LSBs of the color components. The proposed technique can be applied using any bit position which makes more difficult to retrieve the cipher by Stego-analyst. In addition, the histogram is also showing very negligible changes.

5.1 Evaluation of Image quality

For comparing Stego images with cover results it requires a measure of image quality. Commonly used measures are Mean-Squared Error, Peak Signal-to-Noise Ratio and histogram.

I have calculated MSE and PSNR values by giving high capacity secret message in eight different images as shown in Figure 5.3 file. For each file format the value of MSE and PSNR is given in table5.2.

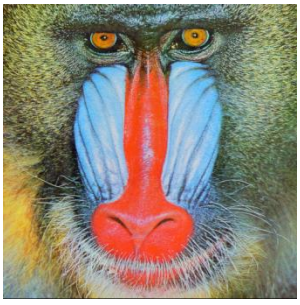
Lenna.png

Fruits.png

Landscape1.png

Landscape2.png

Garden.png



Baboon.png

Figure 5.3: Five Cover Images

Cover Image	MSE	PSNR
Leena.png	0.00032425	83.022
Fruits.png	0.00032425	83.022
Landscape1.png	0.00003689	92.4614
Landscape2.png	0.00018483	85.4631
Garden.png	0.00005404	90.8035
Baboon.png	0.00032425	83.022

Table 5.2: MSE AND PSNR OF DIFFERENT COVER IMAGES

5.2 Comparison

In this proposed technique, LSB is used as status bit when retrieving data from Stego image. We also use symmetry key encryption technique for two layer security, Huffman for decreasing the size of message and Hamming Code for noise disturbance.

From Table 5.3, we see that the PSNR value of the proposed technique is better than other methods providing an efficient way to embed a message into the image without producing clear distortion. However, when the quantity of the lighter and darker pixel in a cover image is very nearest, the size for embedding a message is reduced as compared to that image which has a far difference between the quantity of its lighter and darker pixel.

The length of the secret message in more than the length of message other algorithms used, but still our algorithm gives better results.

Cover Image	PSNR (in dB) Block DCT & Huffman Encoding Method	PSNR (in dB) Huffman Encoding Method	PSNR (in dB) in Secret Key Method	PSNR (in dB) in Status bit Method	PSNR (in dB) in our Method
Leena.png	50.48	57.43	53.7618	74.3923	83.022
Baboon.png	50.28	57.46	53.7558	75.5310	83.022

Table 5.3: COMPARISON WITH FOUR OTHER METHODS

Here, Lenna.png and Baboon.png both are 512 X 512 size cover images. Following figure 5.4 shows the graph of the PSNR values for the algorithms described in the above table.

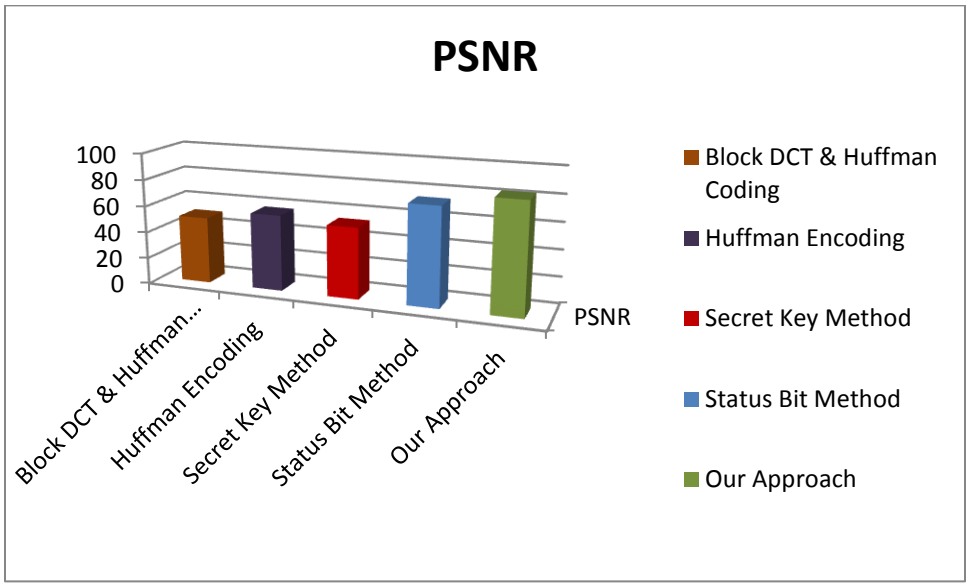


Figure 5.4: Graph of the PSNR values for the proposed algorithms

CHAPTER 5: Conclusion & Future Work

In the proposed technique, I am trying to improve the existing concept of Steganography. I would like to emphasis that the goal of the technique is to increase the capacity of the message and also try to make it difficult to the unauthorized person to determine the presence of a secret cipher.

In ordinary LSB Steganography technique only message bit will be replaced with the LSB bit of the image but out algorithm does not just replace the message bit but it would replace the status of the message bit. Moreover, we merge symmetry key Cryptography with it so that the secret message can be secured by two security layers. And, also we merge Hamming code for taking care of noise disturbance and also making the compressed code word more obfuscated. So, the proposed technique fulfills the requirement of Steganography technique.

This proposed algorithm is applied in the spatial domain, so, the message is changed if any operation is applied on the image like image compression.

So, in the future, we can apply this method in the transform domain steganography and uses DCT or DWT.

References

[1] M. Chen, R. Zhang, X. Niu and Y. Yang, "Analysis of Current Steganography Tools: Classifications & Features," 2006 International Conference on Intelligent Information Hiding and Multimedia, Pasadena, CA, USA, 2006, pp. 384-387.

[2] J. Mielikainen,"LSB Matching Revisited", IEEE Signal Processing Letters , vol. 13, no. 5,May 2006, pp. 285 - 287 .

[3] A. Ker,"Steganalysis of Embedding in Two Least-Significant Bits", IEEE Trans. on Information Forensics and Security, vol. 2, no. 1, March 2007, pp. 46-54.

[4] X. Zhang , and S. Wang, "Steganography using multiple-base notational system and human vision sensitivity", IEEE Signal Processing Letters, vol. 12, Issue 1, Jan. 2005, pp. 67-70.

[5] D.C. Wu, and W.H. Tsai, "A Steganographic method for images by pixel-value differencing", Pattern Recognition Letters, vol. 24, Jan. 2003, pp. 1613–1626.

[6] H.C. Wu, N.I. Wu, C.S. Tsai, and M.S. Hwang, "Image steganographic scheme based on pixel-value differencing and LSB replacement methods", IEE Proc. Vision, Image and Signal Processing, vol. 152, Oct. 2005, pp. 611-615.

[7] R. Crandall, "Some Notes on Steganography", Posted on Steganography Mailing List, 1998.

[8] A. Sur, P. Goel, and J. Mukhopadhyay, "A Spatial Domain Steganographic Scheme for Reducing Embedding Noise", in Proc. 3rd International Symposium on Communications, Control and Signal Processing (ISCCSP 2008), St. Julians, Malta, 12-14 March, pp. 1024 - 1028.

[9] A. Sur, P. Goel, and J. Mukhopadhyay, "A SDS based Steganographic scheme for reducing Embedding Noise", 15th International Conference on Advanced Computing and Communication, (ADCOM-2007), Guwahati, India, 18-21 Dec., pp. 771-775.

[10] I.J. Cox, J. Kilian, F.T. Leighton, T. Shamoon, "A Secure, Robust Watermark for Multimedia", in Proc. of the 1st Int. Workshop on Information Hiding, Cambridge, U.K, 30th May - 1 June 1996 ,pp. 185-206.

[11] E. Koch, and J. Zhao, "Towards Robust and Hidden Image Copyright Labeling", in Proc. IEEE Workshop on Nonlinear Signal and Image Processing, Halkidiki, Greece, June. 1995, pp. 452-455.

[12] H. Farid, and L. Siwei, "Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines", in Proc. 5th Int. Workshop on Information Hiding, Noordwijkerhout, The Netherlands, 7-9 Oct. 2002, pp. 340-354.

[13] J. Fridrich, "Feature-Based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes", in Proc. 6th Int. Workshop on Information Hiding, Toronto, Canada, 23-25 May 2004, pp. 67-81.

[14] A.Westfeld, "High capacity despite better steganalysis (F5 - a steganographic algorithm)", in Proc. 4th Int. Workshop on Information Hiding, Pittsburgh, PA, USA, pp. 289-302, 25- 27 April 2001.

[15] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on wet paper", IEEE Trans. on Signal Processing, Special Issue on Media Security, vol. 53, Oct. 2005, pp. 3923-3935.

[16] X.G. Xia, C.G. Boncelet, and G.R. Arce, "A multiresolution watermark for digital images", IEEE Int. Conf. on Image Processing, Washington, DC, USA, 26-29 Oct. 1997.

[17] T. Pevny , and J. Fridrich, "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.

[18] C. Chen, Y.Q. Shi, W. Chen, and G. Xuan, "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.

[19] H. Farid, "http://www.cs.dartmouth.edu/farid/research/steg.m" (Code for generating wavelet-based feature vectors for steganalysis.)

[20] S. Dumitrescu, X. Wu, and N. Memon, "On steganalysis of random lsb embedding in continuous-tone images",in Proc. IEEE International Conference on Image Processing, Rochester, New York., September 2002.

[21] J. Fridrich, T. Pevny, and J. Kodovsky, "Statistically Undetectable JPEG Steganography: Dead Ends, Challenges, and Opportunities", in Proc. ACM Multimedia and SecurityWorkshop, Dallas, TX, 20-21 Sept. 2007, pp. 3-14.

[22] R Chandramouli , M Kharrazi and N Memon, "Image Steganography and Steganalysis: Concepts and Practices", in Proc. 2nd Int. Workshop on Digital Watermarking, Seoul, Korea, 20-22 Oct. 2003, pp. 35-49.

[23] J. Fridrich, M. Goljan and R. Dui, "Reliable Detection of LSB steganography in Color and Grayscale Images",in Proc. ACM Workshop on Multimedia and Security, Ottawa, CA, 5th Oct. 2001, pp. 27-30.

[24] A.D. Ker, "Steganalysis of LSB matching in grayscale images", IEEE Signal Processing Letters, vol. 12,pp. 441–444, June 2005.

[25] J. Fridrich, M. Goljan, and T. Holotyak, "New Blind Steganalysis and its Implications", in Proc. SPIE Security, Steganography, and Watermarking of Multimedia Contents VIII, vol. 6072, pp. 607201, Jan. 2006.

[26] J. Fridrich, M. Goljan, and D. Hogea,"Steganalysis of JPEG Images: Breaking the F5 Algorithm", in Proc. 5th International Workshop on Information Hiding, Noordwijkerhout, The Netherlands, 79 Oct. 2002, pp. 310 - 323.

[27] J. Harmsen, and W. Pearlman, "Steganalysis of additive noise modelable information hiding", in Proc. Security andWatermarking of Multimedia Contents V, vol. 5020, June 2003, pp. 131-142.

[28] R. J. Mstafa and K. M. Elleithy, "A highly secure video steganography using Hamming code (7, 4)," Systems, Applications and Technology Conference (LISAT), 2014 IEEE Long Island, Farmingdale, NY, 2014, pp. 1-6.

[29] R. Das and T. Tuithung, "A novel steganography method for image based on Huffman Encoding," Emerging Trends and Applications in Computer Science (NCETACS), 2012 3rd National Conference on, Shillong, 2012, pp. 14-18.

[30] M. R. Islam, A. Siddiqa, Md. Palash Uddin, A. K. Mandal and M. D. Hossain, "An efficient filtering based approach improving LSB image steganography using status bit along with AES cryptography," Informatics, Electronics & Vision (ICIEV), 2014 International Conference on, Dhaka, 2014, pp. 1-6.