

# HIGH CAPACITY STEGANOGRAPHY USING DISCRETE COSINE TRANSFORM



**Researcher:**

**Laeq Aslam**

**Reg. No.424-FET-MSEE/F15**

**Supervisor:**

**Dr. Muhammad Amir**

**Professor, DEE, FET**

**Co-Supervisor:**

**Dr. Ijaz Mansoor Qureshi**

**Professor, DEE, Air University**

**Department of Electrical Engineering  
Faculty of Engineering and Technology  
International Islamic University, Islamabad**

**2017**



Accession No TH:18147 *V44*

MS  
005.82  
LAH

Data encryption (Computer Science)

Data protection

# HIGH CAPACITY STEGANOGRAPHY USING DISCRETE COSINE TRANSFORM



**Laeq Aslam**

**(Reg No.: 424-FET/MSEE/F15)**

This dissertation is submitted in partial fulfillment of the requirements for the Master of Science (MS) Electronic Engineering at Department of Electrical Engineering, Faculty of Engineering & Technology, Islamabad.

**Supervised By:**  
**Prof. Dr. Muhammad Amir**

**August 2017**

DEDICATED TO

My Teachers,  
Family and Friends.

CERTIFICATE OF APPROVAL

**Title of Thesis: High Capacity Steganography Using Discrete Cosine Transform**

**Name of Student: Laeeq Aslam**

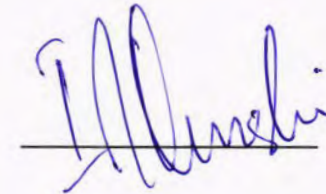
**Registration No: 424-FET/MSEE/F15**

Accepted by the Department of Electrical Engineering, Faculty of Engineering and Technology, INTERNATIONAL ISLAMIC UNIVERSITY, ISLAMABAD, in partial fulfillment of the requirements for the Master of Science (MS) degree in Electronic Engineering.

**Viva Voce Committee**

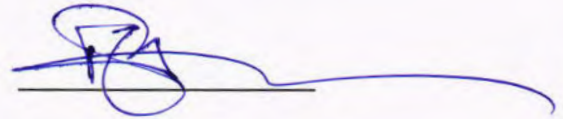
**Co-Supervisor**

Prof. Dr. Ijaz Mansoor Qureshi  
Department of Electrical Engineering  
Air University, Islamabad.



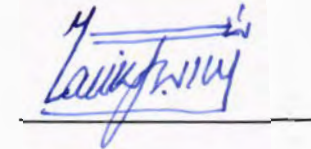
**Supervisor**

Prof. Dr. Muhammad Amir  
Department of Electrical Engineering  
IIU, Islamabad.



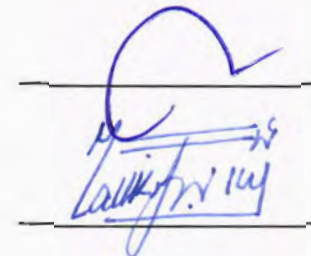
**Internal Examiner**

Dr. Suheel Abdullah Malik  
Associate Professor  
DEE, FET, IIUI



**External Examiner**

Dr. Muhammad Usman  
Director, NECOM, Islamabad



**Chairman**

Dr. Suheel Abdullah Malik  
Associate Professor  
DEE, FET, IIUI.



**Dean**

Professor Dr. Muhammad Amir  
FET, IIUI.

## LIST OF PUBLICATIONS AND SUBMISSIONS

[1]. Muhammad Zaheer, I.M.Qureshi, Zeeshan Muzaffar, and **Laeq Aslam** "Compressed Sensing Based Image Steganography System for Secure Transmission of Audio Message with Enhanced Security" International journal of computer science and Network Security,2017.

## SUBMITTED PAPERS

[1]. **Laeq Aslam** , Muhammad Amir , I. M.Qureshi *and* Wasim Khan " Novel Image Steganography using Adaptive Number of Dominant Discrete Cosine Transform Coefficients", Journal of Information Security and Application, (submitted) 2017.

[2]. **Laeq Aslam**, Muhammad Amir, I. M.Qureshi and Wasim Khan "Novel Image Steganography Based on Pre-processing of Secrete Messages for Enhanced Data Security and Improved Payload Capacity", Multimedia Tools and Applications,(submitted) 2017.

[3]. Muhammad Zaheer, I.M.Qureshi, Zeeshan Muzaffar, and **Laeq Aslam** "High capacity Image Stegnography based on prime series representation and payload capacity reduction" KSII transaction on internet and information security system, (submitted) 2017.

The research work presented in this dissertation is based on the submitted publications 1 and 2 .

## Contents

1	Introduction and Literature Review .....	1
1.1	History .....	1
1.2	Modern Age Data Security .....	2
1.3	Steganography .....	3
1.3.1	Terminology.....	3
1.3.2	Features.....	3
1.4	Steganography VS Cryptography .....	5
1.5	Comparison of Performance Parameters .....	6
1.6	CARRIER BASED CATEGORIZATION OF STEGANOGRAPHY.....	7
1.6.1	Steganography in Text/Documents.....	8
1.6.2	Image Steganography.....	8
1.6.3	Audio Steganography.....	10
1.6.4	Video Steganography.....	11
1.6.5	Steganography in TCP/IP Packets .....	11
1.7	CLASSIFICATION OF IMAGE STEGANOGRAPHY.....	12
1.8	Spatial Domain Image Steganography .....	12
1.8.1	Transform Domain Image Steganography.....	12
1.8.2	Compression Based Image Steganography.....	14
2	Image Steganography using Adaptive Number of Dominant Discrete Cosine Transform Coefficients .....	16
2.1	Introduction.....	16
2.2	Mathematical Foundations .....	21
2.2.1	Discrete Cosine Transform .....	21
2.2.2	Stego-image Quality Measurement & PSNR .....	23
2.3	Proposed scheme .....	24
2.3.1	Embedding process .....	24
2.3.2	Constraint <i>k<sub>min</sub></i> and <i>adaptive k</i> .....	26

2.3.3	Retrieving Process .....	26
2.4	Results comparison .....	27
2.4.1	Results in Proposed scheme.....	28
2.4.2	Results comparison with existing schemes .....	29
2.4.3	Example scenario .....	31
2.5	Conclusion .....	31
Chapter 3	.....	33
3	Preprocessing of Messages for Improving the Payload Capacity .....	33
3.1	Introduction.....	33
3.2	Mathematical Formulation .....	33
3.2.1	Secret Image Estimation .....	33
3.3	Proposed Algorithm .....	35
3.3.1	Embedding Algorithm .....	35
3.3.2	Retrieving Algorithm .....	36
3.4	Results and Comparisons .....	37
3.4.1	Results of the Proposed Algorithm.....	37
3.4.2	Comparison .....	41
3.5	Conclusion .....	42
4	Conculsion and Future Work.....	43
5	Refrences .....	45
6	Appendixes .....	49
6.1	Appendix A .....	49
6.1.1	First Contribution Embedding Algorithm.....	49
6.1.2	First Contribution Retriving Algorithm .....	55



# CHAPTER 1

## 1 Introduction and Literature Review

Exchange of information nowadays is hardly face to face. Every little communication is done using modern day communication technologies. With this, questions of privacy are also raised. Any data stream on internet channels, important or unimportant, can be interrupted and changed by unknown entities. This concern formulated the need of covert and robust techniques to hide sensitive information being transmitted over the web. It started off with military and intelligence purposes and ancient methods like ciphers were modified according to modern needs but technological race, spying governments, copyrights violation and powerful code breaking techniques caused development of better, more sophisticated and specialized techniques in the field of information hiding.

### 1.1 History

The history of hidden communication dates back to the starting of long distance communication by man. The need of privacy and secrecy of messages was ever present and it elevated with time as newer methods of coding and decoding were developed. Data hiding has been achieved by various methods throughout the history including subtle messages like writings on the wax tables by Greeks, straightforward methods like disguising the messenger and imaginative methods of hiding data in plain sight using slaves' shaven heads [1]. Credit of the development of this field largely goes to military and political communications. Most famous data hiding of history is the use of poking holes in the books over intended letters in World War-I by German spies which was replaced by Microdots (invisible ink) in world war-II[2], [3].

Another traditional approach towards hiding data is Cryptography. Using specific information or keys, data is shuffled so that only authorized personnel can decipher it. This method has also been developed since ancient times but lack of technology greatly limited its progress over time. Most successful applications of cryptography were developed using electro-mechanical machines like Enigma. This device used in World War-II by German intelligence used all types of notations and letters in messages and changed decipher keys every 24 hours making messages almost impossible to intercept and decode within time. It wasn't until the computer age that these encryption schemes could be decrypted easily.

## **1.2 Modern Age Data Security**

Information Technology revolution and exponential growth in digital communication has made it challenging to hide information. It is becoming more and more difficult to secure data over internet. Concerns regarding protection of intellectual property have sparked a debate for new data hiding schemes which can be communicated over wireless networks without being eavesdropped or intercepted by malicious attackers. Traditional encryption schemes like RSA[4] and DES[5] are still among most widely used solutions. These schemes scramble data such that it becomes unreadable. However, this approach is becoming ineffective as unreadable data is suspicious so it can be intercepted all the same. With the resources that internet provides us nowadays, there is always a chance that someone is tracking these messages and decrypting them as they come. . Resultantly, the message that were encrypted become ataxia of encrypted data that cannot pass the checkpoint on the network [6]. The modern day digital steganography provides the solution to this problem.

## **1.3 Steganography**

Steganography is the method of hidden conversation using public channel. Modern research in information hiding is greatly focused on steganography domain but it is not a new field. It has been used for centuries to exchange information but the boost in internet technology has shed light on its digital applications which are widespread.

### **1.3.1 Terminology**

Steganography came from two ancient Greek words 'steganos' meaning covered and 'graphein' meaning writing [7]. It means for hiding message within plain sight using unsuspecting cover. Steganography provides added layer of tutelage on the secret message, which will be injected in another information medium such that the transmitted data is expressive and innocuous to everyone. Layer of protection or cover media can be variable depending on the hidden message. Cover medium should be chosen so as to blend with actual message perfectly. These media are least suspicious because of their redundant nature and thus are rendered best to act as a camouflage . A good stegano-system also takes into consideration the compression of the medium. It is designed with a specific compression method which keeps the embedded message from damage in re-compression.

### **1.3.2 Features**

It is necessary for further research in the field that some distinguishing features be defined for a technique to belong to Steganography domain. Following are the basic requirements and features for a hidden communication to be steganography:

- 1) The intended message is intact. There is no scrambling of data. The carrier used to hide the data within, is also unsuspecting and unscrambled.
- 2) The security of data is ensured by the hiding capabilities of algorithm used and the commonality of the carrier on which it is applied and by the capacity to withstand detection algorithms or steganalysis. It is the statistical analysis of captured traffic of legitimate data which might contain a secret message.

It is apparent that most important part of steganography is the choice of carrier. A good carrier must have two qualities. First, it should be popular. There should be massive amount of similar data being transmitted over network or whichever channel is used to communicate so that it doesn't fall into the category of discrepancy if data stream is being captured. Secondly, carrier must not change after the insertion of steganogram. Videos are a popular carrier because they hide the embedded information better than Image steganography. Even if the carrier is degraded by the modifications, it should be limited to a level that it is not suspicious.

The level of secrecy provided by steganography is attracting massive attention due to modern concern for privacy and anonymity. Even general public demands techniques to communicate securely as there is no such thing as secure channel anymore. Everyone and anyone can tap into communication channels. In such a situation, steganography provides the craved result as it makes sender and receiver invisible. But for the very same reason, it also poses a threat to societies and states. Steganography is not being monitored by government and law like cryptography in most countries. Therefore, Steganography is a tradeoff between benefits and threats involving many parameters like legal, ethical and technological issues. We have discussed only technological facets of steganography in this thesis.

## 1.4 Steganography VS Cryptography

Steganography is often mixed up with the cryptography and watermarking. These all fields are related to data hiding. It is difficult to define a boundary between the methods due to the lack of coherent classification of the invented clandestine communication methods. Cryptography is ancient method of communication and its techniques have been developed over time. Watermarking is a very similar field to steganography but it embeds copyrights and intellectual property in cover objects and focuses on ensuring that an intruder cannot change the message hence the main requirement is robustness. On the other hand, steganography is relatively newer mass media interest. Fig 1.1 shows a general categorization of these security systems.

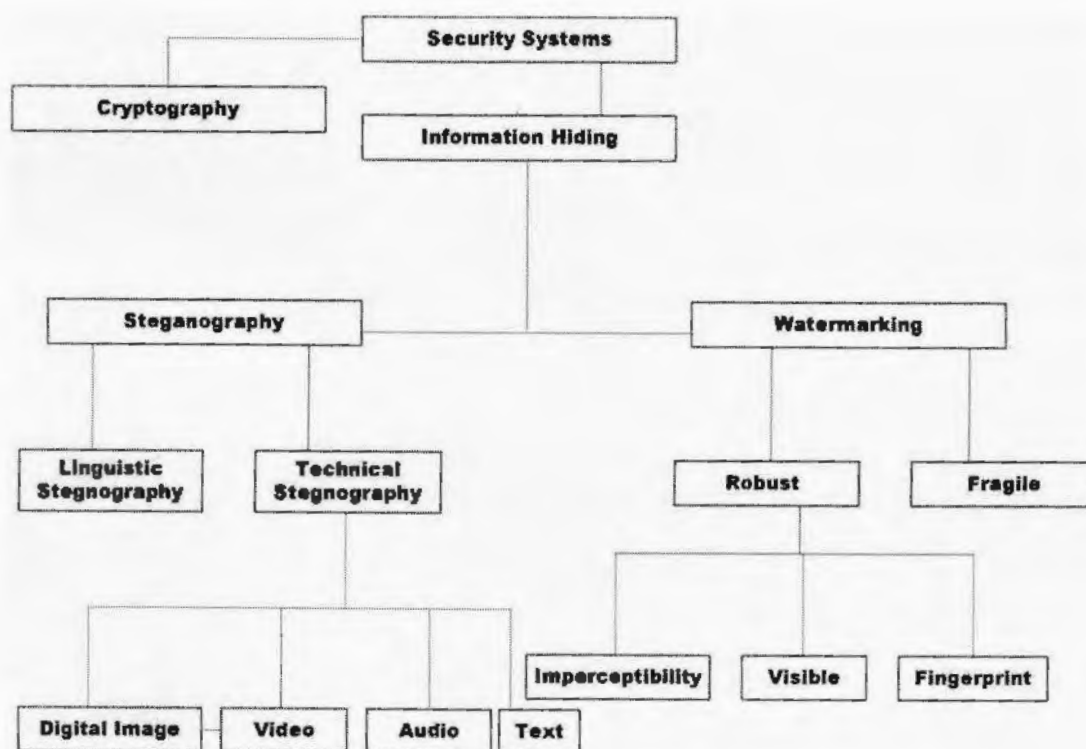


Fig. 1.1 General categorization of security systems

Though the goal of Steganography, Cryptography and watermarking is the same i.e. Information Security; yet all three have very different approach to achieve this objective.

### **1.5 Comparison of Performance Parameters**

Having clearly defined the basic requirements of steganography, we can now compare the technical facets or features required in a good steganography system and compare them with the cryptography and watermarking. Main parameters are:

- 1) Invisibility is also a mandatory property in Steganographic system. Changes in the carrier are made so that human visual or auditory system cannot detect it making it most sophisticated of all techniques. Cryptography meddles with the content of the message which draws attention that something is hidden therefore, does not provide this feature. Watermarking hides a watermark along with the actual data which is relatively easy to detect.
- 2) Payload Capacity is a major factor in determining the efficiency of any information hiding technique. It is the measure of how much data can be embedded without making it perceptible. Capacity of Stegano-systems varies with the type of carrier and algorithm used. Most of the research is focused on increasing payload capacity in image steganography without the usual trade-off of imperceptibility. Cryptography has a relatively higher capacity given longer the message, difficult it is to decrypt it. Watermarking uses the hidden data and has no size of its own therefore, its capacity is low to medium.
- 3) Detection of the hidden information is basic requirement of all techniques. Steganography is preferable if detection is relatively more crucial because it is very difficult of even spot the stego-image let alone extract the hidden information. The systems that use statistical analysis to detect secret stego-messages in random traffic

are called steganalysis. Cryptography needs the data to be large in order to avoid decryption but that makes information more detectable.

- 4) Imperceptibility of the hidden message is also essential. As mentioned above, imperceptibility decreases with the increase in capacity rate. Therefore, cryptography is most perceivable of the data hiding domains. It has high capacity and its robustness increases with the increase in the size of the data but large scrambled data is more prone to cryptanalysis as well. Steganography expects the message to leave the stego-medium statistically similar to the original camouflage in every way possible providing highest level of imperceptibility insurance among all the techniques. Watermarking also provides high imperceptibility by hiding watermark.
- 5) Robustness is the measure of alteration or damage caused to the hidden information in case of attack. On the other hand, for encryption schemes and watermarking, robustness plays a key role in efficiency of a system.

Requirements of these major applications in data hiding vary slightly and therefore, trade-offs are made in above features. In watermarking applications, imperceptibility is sacrificed for robustness and steganography values imperceptibility over robustness. It only needs robustness so that format change in carrier file would not affect the hidden message. Due to this variation in requirements, design problems differ from one to the other. Steganographic techniques are discussed in detail in the next chapter.

## **1.6 CARRIER BASED CATEGORIZATION OF STEGANOGRAPHY**

It is the primary way to categorize steganography. It is also the first choice one makes while working in the field as it determines the performance parameters like capacity and imperceptibility and the trade-off between them.

### **1.6.1 Steganography in Text/Documents**

Text steganography hides the existence of secret message with in text documents as proposed in [8]–[10]. Here the cover media or carrier is a text document. It can be format based, linguistic method or random and statistical generation. Format type is most subtle in text steganography. It could be Line Shift Coding [11] which shifts lines vertically and can be decoded at the receiver end without any original document since it is known to have uniform spacing between the lines.

In [12], authors suggests a technique to embed information by replacing words with synonyms which are not usually the first choice in uniform language like inserting American words in British English. Substitution is done if hidden bit does not correspond to the written word bit. This substitution will represent the binary pattern corresponding to hidden data bits. Word Shift coding changes location of some words within the text lines horizontally reducing largest space and equally extending smallest space keeping the length of the line the same. Variable spaces hold the bits of information. Feature coding is also a text altering technique in which certain text features like length of some alphabets is varied keeping others uniform.

### **1.6.2 Image Steganography**

Image steganography hides existence of secret message in images as discussed in [13]–[15]. As Digital images are made of pixels which are manipulated to deliver the message. Images are very redundant which makes them a popular carrier choice. Human perception of colors is also relatively weak. Due to these reasons, large variety of



techniques and algorithms for hiding information within images have been proposed in the recent years.

Information is hidden in pixel values of the image. Common images are 8-bit grayscale image and 24-bit colored image. Suppose an image is 8-bit. Each pixel uses one byte and one bit of each of those pixels is altered, it would not make any significant changes in the image itself therefore, it will be unsuspecting and will be transmitted without interruption. The larger the image, the more is the capacity to store information. However, large images are not usually transmitted over the web. Therefore, compression is required. Digital images are most widely used carriers due to rapid growth of photographic social media, computer graphics power and limited visual perception of colors by human beings. Large images are capable of hiding another image in themselves and thus compression methods are applied to make large images useable.

Compression of stegano-medium can be lossy or lossless. Lossy compression gives high compression rate but damages original image whereas lossless compression has low compression rate but it preserves the original image serving the basic idea of stegano-system making lossless compression the preferred choice[16].

Image Steganography employs some of the most efficient data hiding algorithms. In [17], authors employ tri-way method of pixel value differencing (TPVD) for enhanced data capacity. Pixel value Differencing usually uses sets of two pixels each to embed secret bit. A range table computes how many bits one set can contain. Secret bit is then added to lower bound range in decimal form. Authors in [17], implements this

approach but with three directional edges instead of one to design the scheme and results in higher embedding capacity.

Image pixels can carry a triplet message by adding or subtracting them from a gray scale value. In [18], authors implement this approach using Hamming code. The method proposed is based on hamming covering function  $C(R, n, k)$ , where  $R$  is the maximum possible changes whenever  $k$  bits of messages  $(m_1, \dots, m_k)$  are injected in the bit that has least significant weight of Bit  $n$  pixel gray scale values  $(b(x_1), \dots, b(x_n))$ . They proposed “Hamming+1” method, in which the block of pixels of the hamming covering function is expanded by one. This algorithm minimizes the rate of embedding changes.

### 1.6.3 Audio Steganography

Audio steganography techniques [19]–[23] have been proposed in the recent years hides the existence of information within audio signals. This is a challenging form of data hiding as human listening has a small differential range as compared to visibility. Even the slightest change in an audio can be detected and analyzed for secret information. Audio steganography involves the choice of audio storage and transmission medium of audio signal to ensure its efficiency. Sampling rate is usually 8KHz and quantization method uses 16-bit linear quantization. Transmission medium used can be digital end to end or analog-resampling.

Audio steganography can be done by low-bit encoding which is similar to LSB in image steganography. Phase coding uses DFT on original sound sequence and new phase frames are created for segmented data. It is a relatively secure method because human

auditory system cannot detect phase noise. Spread spectrum and echo data hiding are also commonly used methods. Audio steganography manipulates redundant bit of the carrier audio in order to hide information as noise or echos.

An implementation of LSB based audio steganography in [23], uses two step algorithms which exploits higher LSB layers to hide information. First step embeds hidden bits of information in  $i$ th layer and second step impulse noise of the embedded data is shaped. Embedding error is reduced by flipping other bits.

#### **1.6.4 Video Steganography**

Video steganography hides the existence of secret messages with in video that allows high data capacity as discusses in [24], [25] Videos are made up of image frames and sounds. Two previously mentioned categories are combined in this type of communication providing a wide spectrum for steganography. Anything from text to image to sound can be hidden in a video with low chance of detection. Video steganography employs different combinations of the techniques of audio and image steganography e.g. hiding half the information in image frame and the other half in audio component using wavelet transform for image hiding and LSB to hide length in audio. Another technique is to split frames in non-uniform rectangular partitions.

#### **1.6.5 Steganography in TCP/IP Packets**

Network or protocol steganography hides messages in TCP /IP packets as discussed in [26]. Though it is not very popular, open system interconnection or OSI

network models can provide a good host for covert communication. Reserved bits in TCP/IP packets are used for this purpose.

## **1.7 CLASSIFICATION OF IMAGE STEGANOGRAPHY**

By far, we have discussed basic features and methodology of Steganography. Now we narrow down problem areas in the field and explore the literature on the subject. There are two main ways to categorize i.e. based on carrier type and based on algorithms. For every type of carrier, there has to be a specific data embedding algorithm. The different approaches used to perform Image steganography on the basis of embedding algorithms fall into three categories:

### **1.8 Spatial Domain Image Steganography**

In this approach, the camouflaged image is directly encoded in to the pixels of the stegano-medium image. Some of the methods used include LSB encoding, RGB encoding, pixel value differencing encoding, palette based steganography, mapping based steganography. In [27] , authors proposed a lossless generalization of the LSB Modification technique that compresses and embeds the parts of the signal that a susceptible to distortion as description as part of the payload.

Whereas, in [28], authors used LSB substitution and by replacing Wang et al's genetic algorithm with dynamic programming achieved optimal solution and reduced computation time. Wang et al proposed exhaustive least-bit substitution scheme and then improved it by proposing a genetic algorithm to find proximate ideal solution.

#### **1.8.1 Transform Domain Image Steganography**

In this approach, the stegano-medium image is first transformed into different domains (Frequency domain, wavelet Domain etc) by using methods including Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Wavelet Transform and

other methods. Then the secret Image or Message is encoded into the Quantized coefficients obtained from the transform. The Quantization scheme used influences the strength of the Steganography. Varying the Quantization scheme changes the overall capacity, imperceptibility and robustness of the steganography.

Authors in [15], proposed a lossless and reversible method of data hiding using DCT and JPEG images. For each block the two successive zeros between medium frequencies were used to encode covert data. They also mutated the JPEG quantization matrix to achieve improved imperceptibility. Whereas, in [29], authors proposed a solution based on the theory that major chunk of image data is in the phase spectrum of Fourier transform, if the phase information is preserved the hidden image can be extracted even if magnitude information is distorted achieving higher payload capacity (up to half the size of stegno-medium image) and better robustness. Moreover in [30], same authors present improvements to previous method to achieve better capacity without compromising on any other factors. In[31], authors addresses the issue of increasing payload capacity with increasing imperceptibility by proposing a method using Discrete Cosine Transform and Global Adaptive Regioning (using variable region size to embed hidden data within a block) to attain excessive payload space and high imperceptibility. The size of the region is adapted based on the correlation of the image values in that block. The regions that have less correlation are used to large amount of data while regions with more correlation are used to hide less data. Thus, when a database of images is available a suitable image can be chosen which provides the capacity to embed the hidden image as well as used least amount of bandwidth (smaller size). To achieve this, the proposed method is capable of providing suitable estimate of

payload capacity for candidate images and hence the ability to choose optimal image for steganography. However, they have used square blocks for data embedding whereas, work presented in this thesis embeds data in non square L shaped blocks for high data capacity. First part of the thesis is an extension of this technique in which we improve the payload capacity as well as capacity of the stego image.

### **1.8.2 Compression Based Image Steganography**

In this technique, the hidden image or message is first compressed using different methods (JPEG, VQ, SMVQ etc ). The compressed message is encoded into the uncompressed stegano-medium image. The decoding process results in the hidden image or message.

Authors in [32], proposed a method to hide secret images using compression steganography. For embedding data into compression codes, they used genetic algorithm, block truncation techniques and modification direction techniques. They maintained the compression ratio of the original bitmap generation procedure so that it is difficult for attackers to distinguish between stegno and normal images. They also achieved high payload capacity while maintaining reasonable imperceptibility. Authors in [33], proposed a method for compression steganography based on Side Matching Vector Matching Quantization to achieve high payload and acceptable stegno image quality. They proposed encrypting hidden message and embedded it in uncompressed SMVQ image. Hence upon decompressing the hidden message and actual SMVQ code are obtained without error.

## Chapter 2

### 2 Image Steganography using Adaptive Number of Dominant Discrete Cosine Transform Coefficients

This chapter presents our first research paper of an image steganography scheme that preserves an adaptively chosen block of dominant coefficients from each Discrete Cosine Transform coefficients, whereas the rest of the coefficients are replaced with normalized secret image pixel values. The pixel values of secret image are normalized in an adaptively chosen range. Embedding such kind of normalized data in adaptively chosen non-square L- shaped blocks utilize maximum embedding space available in each block that consequently results in maximizing payload capacity, while maintaining the image quality.

#### 2.1 Introduction

Throughout the history humanity has been striving to achieve secrecy in information communication. Steganography was one of the most frequently used solutions. Steganography is a Greek word that refers to concealed writing. The word “Steganos” means “covered” and “Graphial” means “writing”[34], [35], so steganography is nothing but concealing information in some other sort of information. Greeks used to tattoo secret message on the salves shaved heads and messages were sent when their hair grew back. On the receiving end they shave the head again and retrieve the secret message. They were also using wax tables for the same purpose. In Second World War, the use of invisible ink was very common[36]. Germans were the first to use microdots for secret communication[36], [37].

In the modern era of digital communication, achieving secrecy is still an open challenge. Numerous methods like encryption and digital steganography have been developed to meet the challenge. Encryption schemes like RSA (Rivest, Shamir, and Adelman) algorithm [4] and DES (Data Encryption Standard) algorithm [5] are frequently used commercially. Encryption scrambles data in such a way that data become unreadable. As a matter of fact the unreadable data is suspicious and attracts more attention of unintended users. Thus a third party may keep on trying and consequently succeed in decryption of data. However, Steganography camouflages the presence of secret message in a host medium. That cover medium could be any ordinary sort of information in any format. This secret data embedding is meaningful only if the existence of secret message is not obvious. Thus the life of steganography lies in imperceptibility.

Many digital steganography techniques have been suggested in the last two decades. All of them share a fundamental concept of injecting secret message in a host medium to generate a stego-output as shown in Fig.2.1. Among different types of steganography techniques i.e. text, image audio, video and network or protocol steganography are famous. The types are classified on the basis of their cover medium i.e. image steganography takes image as a cover medium. Secret messages could be of the same or of different type. Furthermore, the parameters for comparison among different steganography techniques are imperceptibility, robustness (against channel impairments) and payload capacity measured in bit per pixel (bpp) [38].



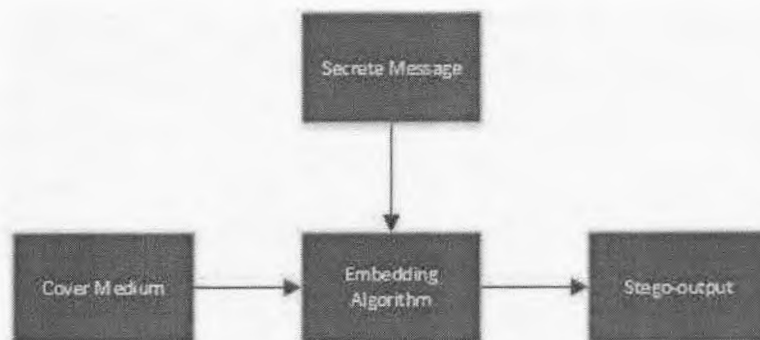


Fig. 2.1 General Steganography diagram

The image steganography techniques can be branched into spatial domain steganography techniques and frequency domain techniques. Whereas, in spatial domain techniques the secret message is injected right in to the pixel intensity value. The most widely used method for spatial domain steganography is injecting information with in those bits that have least significant value in with in a pixel intensity value. In [14] author proposed a spatial domain high capacity steganography scheme by selecting variable size of L.S.B substitution in each pixel to maintain resemblance between cover and stego-images. Another scheme[39] used shared colour palette achieved embedding capacity up to 6 bpp while keeping Peak Signal to Noise ratio (PSNR) up to 40 dB. Another scheme proposed in [40], used pixel value differencing (PVD) along with adaptive LSB injection for data hiding in edge area of cover images .

In frequency domain techniques, first step is to transformed data that is an image from spatial to its frequency domain equivalent using Fast Fourier Transform (FFT) [41], Discrete Cosine Transform (DCT)[15], [31], [42]–[44] and Discrete Wavelet Transform (DWT)[45]. Later the high frequency coefficients of the cover image are discarded. As in [31], the author used the standard JPEG quantization matrix with the quality factor 50. Finally the covert message is injected in place of those coefficients that has the highest frequency and inverse transformation is applied to generate stego-image.

Among many one challenge is to achieve maximum payload capacity while maintaining PSNR above a certain level. The maximum payload capacity will help us to save the bandwidth that is a precious commodity. Improving PSNR will make it almost impossible to perceive the existence of secret message in an image. Recently the scheme proposed in [31] achieved much better results than the existing schemes. The author introduced an idea of using a Global Adaptive region approach for data embedding that achieved a much higher payload capacity.

The work presented in this thesis introduces a new Discrete Cosine Transformed (DCT) based image steganography algorithm for extremely high capacity embedding with an improved PSNR. Instead of choosing a square block for secret data embedding as in [31], [46], the proposed algorithm embeds data in an adaptive non-square L shaped block. Such an adaptive non-square blocks embedding results in improving payload capacity. Furthermore, existing scheme like [31], [46] normalise the secret message in a fixed range before embedding, whereas this algorithm adaptively choose the normalization range for each  $8 \times 8$  window. This results in improving quality of the stego-images. Moreover, the scheme is computationally less complex then the schemes discussed in [31], [46].

The fundamental requirements for any steganography algorithms are high capacity data embedding, less distortion in the covered image and high data security. PSNR and the high capacity data embedding are inversely related to each other i.e. increasing one results in decreasing other. As imperceptibility is the lifeline of every steganography algorithm, therefore, one cannot improve capacity at the cost of decreasing quality of stego-images below a certain level. Hence, capacities of various image steganography algorithms[14], [30], [31], [39], [43], [47]–[49] were very less during the past few years. Recently proposed algorithms[31], [47], [49] attained significant improvement in term of capacity as well as image quality, however, it

requires a key to be sent along with the stego-image for recovering the secret image. Once this key is disclosed the communication will no more be secure.

The image steganography algorithms can be further subdivided into two main categories that are spatial domain techniques and frequency domain techniques. Spatial domain techniques hide secret information directly into the pixel intensity value as in [14], the authors proposed a scheme that substitute adaptively chosen number of bits. The said adaptive number of substituting bits was relatively high in boundary pixel values. Method presented in [40], use LSB substitution and pixel value differencing (PVD) to hide in edges areas of an image. In [28], author used dynamic programming strategy to find out number of optimal substitution bits. Beside such substitution techniques authors in [50] proposed a steganography method based on singular value decomposition. Another scheme [51] propose an algorithm for reversible data hiding scheme for the images compressed with block truncation codes.

Frequency domain techniques transforms images to its frequency domain equivalent using Discrete Fourier Transform (FFT) [30], [41], Discrete Cosine Transform (DCT) [15], [31], [47], [52]–[54] or Discrete Wavelet Transform (DWT) [13], [55]. Later on they substitute bits of the frequency domain coefficients. As already discussed human visual system is not sensitive against changes in high frequencies or low correlating areas therefore, more bits can be replaced in place of coefficients representing higher frequency. In [41], author proposed that the changes in the Fourier magnitude are not obvious if the Fourier Phase are maintained. Hence, they substitute the Fourier magnitudes with secret image while maintaining the phase of those Fourier magnitudes to achieve higher imperceptibility. In [55] authors proposed an algorithm that hides data after taking 4th level DWT using Mallet algorithm with biorthogonal 9/7 basis. They used  $HL_{4,1}, LH_{4,2}, HH_{4,3}, HL_{3,1}, LH_{3,2}, HH_{3,3}, HL_{2,1}, LH_{2,2}, HH_{2,3}$  for hiding message. These sub-bands have

high energy of all high frequency sub-bands. The algorithm proposed in [47], first converts image into  $N \times N$  block size and later on take DCT of each block. Later on they quantize each block with the standard Jpeg Quantization matrix with quality factor 50 to calculate an average block size for data embedding. The results achieved in [47] were improved in [31] by replacing adaptive number of DCT coefficients in each block of  $N \times N$  size. The size of data embedded in each block is sent as an adaptive key along with stego-image for retrieving secret image. This scheme achieved high capacity as well as acceptable PSNR than all existing schemes. However, sending key along with stego-image reduces security while communicating i.e. in case the key is decoded then the covert data can easily be recovered from each window.

## **2.2 Mathematical Foundations**

This subsection explains important mathematical concept used for injecting, extraction and quality measurement of the stego-image.

### **2.2.1 Discrete Cosine Transform**

The discrete cosine transform is used for converting signals to their frequency domain representation. In image processing 2D DCT (i.e. the extension of 1D DCT) is used for getting frequency domain representation of an image. The jpeg (joint photographic expert group) standard uses 2D DCT for image compression is discussed in [43], [56]. Jpeg image compression first converts an RGB image to Ycber and then transforms non overlapping windows of  $8 \times 8$  size from spatial to frequency domain using DCT: given by

$$F(u, v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \times \left[ \cos\left(\frac{\pi u(2x+1)}{2N}\right) \right] \left[ \cos\left(\frac{\pi v(2y+1)}{2N}\right) \right]$$

$$\text{where } C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{otherwise,} \end{cases} \quad (2.1)$$

and  $f(x, y)$  is the intensity value of the image.

The DCT transformation of a  $N \times N$  window will give us the same size window of DCT coefficients. The coefficients in the top left corner represent lower frequencies whereas the coefficients in the bottom right corner of the window represent higher frequencies. Furthermore, the coefficient  $F(0,0)$  is the dc coefficient i.e. the coefficient representing frequency equal to zero and the coefficient  $F(N, N)$  is the highest frequency component.

Later jpeg compression standard quantizes the non-overlapping windows using  $8 \times 8$  quantization matrix. Various quantization matrices have been proposed for achieving different compression ratios. All of them are designed in such a way that high frequency coefficients are heavily quantized as compared to the low frequency coefficients of  $8 \times 8$  window. This is just because sensitivity of human visual system is high for low frequency coefficients as compared to the high frequency coefficients. The proposed scheme embeds secret message in place of high frequency coefficients and then take inverse discrete cosine transform using to generate stego-image. The DCT inverse is given by

$$f(x, y) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v) F(u, v) \times \left[ \cos\left(\frac{\pi u(2x+1)}{2N}\right) \right] \left[ \cos\left(\frac{\pi v(2y+1)}{2N}\right) \right] \quad (2.2)$$

$$\text{where } C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{otherwise} \end{cases}$$

### 2.2.2 Stego-image Quality Measurement & PSNR

As already discussed, imperceptibility is the lifeline of steganography. Once the presence of secret message in a cover medium is perceived, the communication will no more be secure. In image steganography one of the subjective test discussed in [31] asks individuals to find out the original images from a stack of images containing cover and stego-images. If rate of success is under 50% it is stated that the secret message is invisible.

The secret message added in a cover image acts as noise within an image. Instead of the subjective test already discussed the Peak signal to Noise Ratio (PSNR) is widely used in signal processing community. Author in [14], [15], [43], [56], [57] used PSNR for calculating stego-image quality: given by

$$PSNR = 20 \log_{10} \left( \frac{Max_X}{\sqrt{MSE}} \right) \quad (2.3)$$

$$\text{Where, } MSE = \frac{1}{M \times N} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [X(i, j) - \hat{X}(i, j)]^2$$

Where  $M$  and  $N$  represents rows and columns of host image  $X$  whereas,  $\hat{X}$  is the stego-image. Mean square error is less if the stego and the cover resemble each other resulting in a higher PSNR.

## 2.3 Proposed scheme

The proposed scheme for achieving high capacity embedding with improved PSNR is discussed in this section. All steganography techniques propose an embedding and a retrieving algorithm. In the coming subsection we will discuss both embedding and retrieving algorithm.

### 2.3.1 Embedding process

**Step 1:** First step is to sectioning the host image into  $8 \times 8$  non overlapping windows.

**Step 2:** Take 2D-DCT of each window using (2.1). Fig. 2.2a shows the example of 2D-DCT coefficient magnitudes.

**Step 3:** Quantize the coefficient of each  $8 \times 8$  window using the standard JPEG quantization matrix "Q" with the quality factor 50 shown in Fig. 2.2b: using

$$D(u, v) = \text{round} \left\{ \frac{F(u, v)}{Q(u, v)} \right\} \quad (2.4)$$

**Step 4** Detect a non-zero coefficients block of size  $K \times K$  in the top left corner of matrix **D**. This is the minimum size of cover image dct coefficients that much be retained in top left corner of stego-image. The size of this block is adaptively chosen for each window and sent as a secret key along with the stego-image. We can put some constraint on minimum size of this block to improve PSNR. This constraint is discussed at the end of this section.

851.00	90.77	43.38	10.01	3.75	11.16	9.36	3.81
99.03	5.05	25.84	23.17	14.71	12.01	1.20	4.47
41.64	25.15	25.30	25.85	11.82	7.19	1.48	0.62
10.24	14.34	13.41	11.91	6.95	4.09	4.10	1.82
6.75	15.26	9.91	5.94	4.50	2.19	2.38	2.24
9.84	12.56	5.18	2.48	2.35	3.26	0.98	1.98
7.10	1.39	0.98	2.26	2.40	1.21	0.19	0.64
3.13	3.91	1.07	3.13	2.06	0.79	0.50	0.22

(a)

16.00	11.00	10.00	16.00	24.00	40.00	51.00	61.00
12.00	12.00	14.00	19.00	26.00	58.00	60.00	55.00
14.00	13.00	16.00	24.00	40.00	57.00	69.00	56.00
14.00	17.00	22.00	29.00	51.00	87.00	80.00	62.00
18.00	22.00	37.00	56.00	68.00	109.00	103.00	77.00
24.00	35.00	55.00	64.00	81.00	104.00	113.00	92.00
49.00	64.00	78.00	87.00	103.00	121.00	120.00	101.00
72.00	92.00	95.00	98.00	112.00	100.00	103.00	99.00

(b)

53.00	8.00	4.00	1.00	0.00	0.00	0.00	0.00
8.00	4.00	2.00	1.00	1.00	0.00	0.00	0.00
3.00	2.00	2.00	1.00	0.00	0.00	0.00	0.00
1.00	1.00	1.00	0.00	0.00	0.00	0.00	0.00
0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

(c)

851.00	90.77	43.38	Embedded Secret Image Pixel values				
99.03	50.05	25.84					
41.64	25.15	25.30					

(d)

Fig. 2.2 (a) DCT coefficient of 8x8 window, (b) Standard jpeg quantization matrix Q, (c) Quantized DCT coefficient and (d) DCT coefficient of cover image updated with secret pixel values

**Step 5:** In this step the non-zero  $K \times K$  block ( $3 \times 3$  red boarded area Fig. 2.2c) of matrix **D** is replaced with original cover image dct coefficients i.e. the coefficients before quantization. Whereas, the L shaped block (blue shaded area Fig. 2.2c) of matrix **D** is replaced with the normalised pixel values of the secret message. The values are normalised using (2.5).

$$NSI = \left[ \frac{SI}{255} \right] \times NF \quad (2.5)$$

Where  $SI$  is the secret message and  $NSI$  is the normalized secret message the normalization factor  $NF$  is the upper limit value of the replaced original coefficients i.e. the coefficients before quantization. The value of ( $NF$  i.e. 26 in the given example) is



saved along with the size of block already saved in step 4. Final  $8 \times 8$  DCT transformed window of a updated host image is shown in Fig. 2d.

**Step 6:** Finally take Inverse discrete transform using (2) and then join all  $8 \times 8$  windows to generate the stego-image and is sent along with the key .

### 2.3.2 Constraint $k_{min}$ and adaptive $k$

The scheme proposed in [31], [46] divide image in  $N \times N$  window size where  $N$  varies from 8 to 256. Increasing window size results in achieving higher payload capacity, consequently the stego-image quality decreases. Every steganography scheme requires such flexibility for practical applications. One such application scenario is discussed in [31], [46].

However, the method proposed here sections an image in a fix window size i.e.  $8 \times 8$ . Hence, for achieving the flexibility of trading-off between capacity and stego-image quality the concept of  $k_{min}$  is introduced. In step 4 of proposed algorithm the size  $K \times K$  is adaptively chosen. Preserving  $K \times K$  block of cover image dct coefficients in each  $8 \times 8$  stego-image window results in maximum payload capacity consequently the PSNR is low. The size of  $K \times K$  block can be restricted such that if  $K < K_{min}$  then  $K = K_{min}$  this results in improving PSNR at the cost of decreasing payload capacity.

### 2.3.3 Retrieving Process

**First step:** The stego-image received with secret key is first divided into  $8 \times 8$  window and then each window is again converted back to its frequency domain equivalent using (2.1).

**Second step:** In a  $8 \times 8$  DCT coefficient window we have total 64 coefficients. Beside that we know the size of cover image coefficient's square block in top left corner of each window. The remaining coefficients belong to the secret message and are extracted.

**Third step:** The secret key also contains the N.F for each window the data extracted from each window in the last step is rescaled using (2.6) and finally rearranged to generate secret image  $SI$ .

$$SI = \left[ \frac{NSI}{NF} \right] \times 255 \quad (2.6)$$

## 2.4 Results comparison

This section compares the result of proposed scheme with the existing schemes. The data set used to test proposed scheme is shown in fig 2.3. This data set is similar to the one used in [31]. In [31], the author used different window size varying from  $8 \times 8$  to  $256 \times 256$ . The proposed scheme has a fix window size of  $8 \times 8$  but the constraint  $K_{min}$  already discussed in previous section can limit the embedding capacity for improving PSNR. Finally we have shown that this method is computationally less complex for the application scenario proposed in [31].

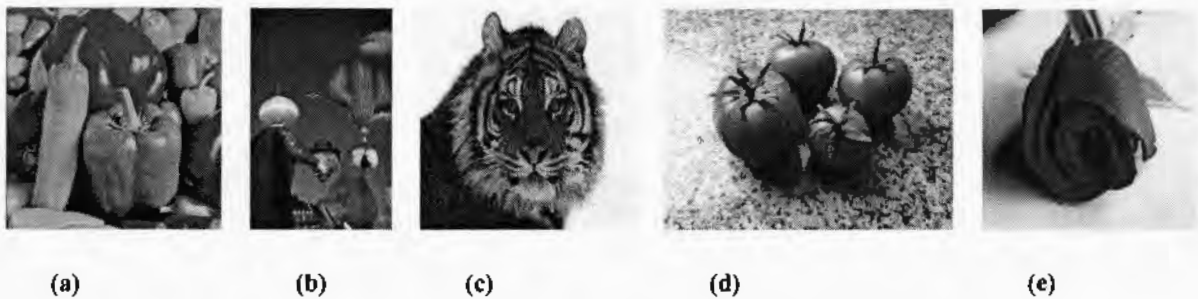


Fig. 2.3 From (a) to (d) cover images,(e) secret image

## 2.4.1 Results in Proposed scheme

Table 2.2. Comparison between GAR-DCT scheme proposed in [31] and the scheme proposed in this research

Cover Image	Window Size	GAR DCT		Proposed Scheme window size $8 \times 8$		
		Capacity(bpp)	PSNR(db)	constraint	Capacity(bpp)	PSNR(db)
Pepper	$8 \times 8$	10.15	32.8	$K_{min} = 4$	16.45	32.88
Balloons	$8 \times 8$	11.09	36.4	$K_{min} = 4$	16.45	42.9
Snow Tiger	$8 \times 8$	9.03	27.9	$K_{min} = 4$	16.45	37.94
Tomatoes	$8 \times 8$	8.1	37.6	$K_{min} = 4$	16.45	43.05
Zebra	$8 \times 8$	7.81	32.5	$K_{min} = 4$	16.44	37.06
Pepper	$64 \times 64$	17.64	28.0	$K_{min} = 3$	18.84	32.29
Balloons	$64 \times 64$	17.25	32.2	$K_{min} = 3$	18.84	39.34
Snow Tiger	$64 \times 64$	17.25	24.9	$K_{min} = 3$	18.76	37.39
Tomatoes	$64 \times 64$	16.62	29.4	$K_{min} = 3$	18.85	41.87
Zebra	$64 \times 64$	17.17	23.0	$K_{min} = 3$	16.69	34.87
Pepper	$128 \times 128$	18.21	27.4	$K_{min} = 2$	20.43	29.53
Balloons	$128 \times 128$	18.21	31.0	$K_{min} = 2$	20.48	33.69
Snow Tiger	$128 \times 128$	18.79	24.3	$K_{min} = 2$	20.26	34.62
Tomatoes	$128 \times 128$	18.29	27.8	$K_{min} = 2$	20.55	40.20
Zebra	$128 \times 128$	19.88	18.6	$K_{min} = 2$	20.07	34.10
Pepper	$256 \times 256$	18.13	27.9	adaptive K	21.14	28.25
Balloons	$256 \times 256$	19.2	28.8	adaptive K	21.34	31.59
Snow Tiger	$256 \times 256$	20.48	22.8	adaptive K	20.07	32.05
Tomatoes	$256 \times 256$	20.31	26.0	adaptive K	21.5	38.24
Zebra	$256 \times 256$	20.05	18.9	adaptive K	20.81	31.48

In this section we will discuss the result achieved by applying the proposed scheme on the cover images shown in Fig 2.3. The flower image shown in Fig 2.3 is taken as a secret message. The Fig 2.4 shows different stego-images after embedding the flower image. Instead of taking different window size as in [31], [46] the proposed scheme takes a fixed window size of  $8 \times 8$  but by putting constraint " $K_{min}$ " over the maximum embedding limits improves the PSNR. Results in table 2 clearly depicts the significant improvements of the proposed algorithm in term of payload capacity and PSNR.



Fig. 2.4 stego-images, from (a) to (e) with  $k_{min} = 2$  and (f) to (j) adaptive  $K$

## 2.4.2 Results comparison with existing schemes

Secret image (flower image) was embedded in four different cover images (peppers, snow tiger, balloons and tomatoes) shown in Fig 2.3. The scheme proposed in [31] achieved the maximum payload capacity with the “tomatoes “ image that was 20.31 bpp for a coloured image and 26dB PSNR. Another scheme proposed in [13], achieved the capacity of 15.1 bpp with the 18.4 dB PSNR. Author of [46] proposed a method using discrete cosine transform for attaining embedding capacity of 20.22 bpp with 25.1 dB PSNR. The method proposed by Lee & chen in [14] embeds a secret message with the payload capacity of 12.18 bpp with 34.03 dB PSNR. The scheme proposed in [39]

TH: 18147

achieved the PSNR up to 40db that was relatively high than other proposed methods but the payload capacity was only 6bpp.

However, the method proposed in this research paper embeds the secret message at 21.5 bit per pixel with a 38.24 dB PSNR in “tomatoes” cover image. These results are achieved without any constraint “ $K_{min}$ ” on the minimum size of retained DCT coefficients in each window of cover image. If we put the constraint “ $K_{min} = 4$ ” i.e. at least  $4 \times 4$  size block of cover image DCT coefficients must be retained in each  $8 \times 8$  window of stego-image, this results in payload capacity of 13.37 bpp with 44.7 dB PSNR in same cover image. Table2.1 compares different algorithms discussed in this section with the already existing schemes

**Table 2.1. Results comparison in term of payload capacity and PSNR**

Scheme	Payload Capacity (bpp)	PSNR (dB)
Lee & Chen [14]	12.18	34.03
Brisbane et al. [39]	6	40
Saeed & Shahrokh [13]	15.1	18.4
Rabie & Kamel[46]	20.22	25.1
GAR-DCT(256x256) [31]	20.31	26
<b>Proposed Scheme (k=4)</b>	<b>13.37</b>	<b>45.22</b>
<b>Proposed scheme (k=0)</b>	<b>21.5</b>	<b>38.24</b>

### 2.4.3 Example scenario

This scheme can also be used in the same scenario proposed in [31], [46] where we have to choose an image that will give us better PSNR on a craved payload capacity. As an example if the embedding capacity of 20 bpp is required from the selected dataset, best option will be a tomatoes image that will give us PSNR up to 40 db with the constraint " $K_{min} = 2$ ".

The schemes proposed in [31], [46] takes an image and apply DCT on different window sizes and then embeds and calculate the embedding capacity and PSNR. Hence, we have to apply DCT on an image six times for six different window sizes. Whereas, the proposed scheme takes DCT of fix window size i.e.  $8 \times 8$  only once for a cover image. Later it embeds the secret data by varying constraint " $K_{min}$ " for achieving different PSNR against the payload capacities. Hence, this method requires less computation than the scheme proposed in [31][46].

## 2.5 Conclusion

The work presented in this paper proposed a DCT based steganography scheme that achieved better results in term of payload capacity and PSNR than the existing schemes. The idea is to retain a square block of cover image DCT coefficients in the top left corner of the host image. The remaning space of  $8 \times 8$  DCT block is replaced with the secret message normalized intensity values to generate a stego-image. Embedding data in a non-square block give us improved payload capacity. Furthermore, the normalization factor for each window is chosen according to the statistics of that window this gives us improvement in PSNR. Moreover, the application of proposed scheme is demonstrated by the same scenario proposed in [31], [46]. The

proposed scheme only used window size of  $8 \times 8$  , this results in less computational cost than the schemes proposed in[31], [46].

## Chapter 3

### 3 Preprocessing of Messages for Improving the Payload Capacity

#### 3.1 Introduction

The work presented in this chapter presents a new steganography algorithm with enhanced two layer data security with improved payload data embedding ability and an acceptable level image quality measured in term of PSNR. The main idea is to convert secret message in to small blocks and then find those small block with in a large but finite indexed dataset of images. The index of the the block that is approximately same to the secret image block is stored. This decreases the numer of bits required for representing secret image or subsequently increases payload capacity. Later on, these indexes are embedded in to the host image DCT coefficients .This makes the communication imperceptible. Moreover, embedding indexes into the host image make this problem unsolvable for an unintended receiver to regenerate secret image from indexes until and unless they get the same dataset indexed in same order. Thus this scheme achieved better payload capacity and enhanced data security as compared to the recently published steganography schemes [14], [30], [31], [43], [47], [49], [52].

#### 3.2 Mathematical Forumatation

This section explain the mathematical basis related to the work presented in this chapter.

##### 3.2.1 Secret Image Estimation

Consider a data set (D) of  $n$  images such that images  $I = \{I_1, I_2, \dots, I_n\}$  has unique image number. Each image of size  $m \times m$  is further sub divided into nonoverlapping window size of  $1 \times k$ . so the data set is given as



$$D = \begin{pmatrix} I_{11} & I_{21} & I_{31} & \dots & I_{n1} \\ I_{12} & I_{22} & I_{32} & \dots & I_{n2} \\ I_{13} & I_{23} & I_{33} & \dots & I_{n3} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ I_{1p} & I_{2p} & I_{3p} & \dots & I_{np} \end{pmatrix}$$

In  $I_{ij}$   $i$  represent the image number and  $j$  represents the window number and  $p$  is the total number of non over lapping windows in each image given by

$$p = \frac{m \times m}{1 \times k} \quad (3.1)$$

Now if we divide and rearrange secret image (SI) of size  $m \times m$  in to nonoverlapping window size of  $1 \times k$  we have

$$SI = (SI_1, SI_2, SI_3, \dots, SI_p)^T$$

Where  $SI_1$  represents first window of size  $1 \times k$  and  $p$  the total number of windows is give by eq. 3.1. The main objective is to find out the small blocks of secret image with in a indexed data set of images such that we get the least possible error tween the covert image block and the dataset images blocks. Generate matrix R having  $p$  rows and 2 columns where each row represents one window in secret image. Error for  $I_j$  is defined as

$$E_{I_1} = \frac{1}{p} \begin{pmatrix} \sum |SI_1 - I_{11}| & \sum |SI_2 - I_{11}| & \dots & \sum |SI_p - I_{11}| \\ \sum |SI_1 - I_{12}| & \sum |SI_2 - I_{12}| & \dots & \sum |SI_p - I_{12}| \\ \sum |SI_1 - I_{13}| & \sum |SI_2 - I_{13}| & \dots & \sum |SI_p - I_{13}| \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \sum |SI_1 - I_{1p}| & \sum |SI_2 - I_{1p}| & \dots & \sum |SI_p - I_{1p}| \end{pmatrix}$$

If any element of  $E_{I_1}$  is below a specified threshold then the index of corresponding data set image is stored at corresponding location of matrix R. Calculate  $E_{I_2}$  for the remaining

windows of Secret image which had error above acceptable level with first data set image. This iterative process continues till all Windows find their equivalent window in the data set.  $R$  is a matrix of size  $p \times 2$  where column 1 represents the image number and column 2 shows the window number within that specific image and  $p$  is the total number of windows given by eq.3.1. The secret image can be regenerated from these indexes if we have data set  $D$ . The regenerated secret image  $SI$  is an estimate of actual secret image.

### 3.3 Proposed Algorithm

Proposed algorithm further more consists of two main algorithms i.e. embedding algorithm and retrieving algorithm. Both algorithms are discussed as following.

#### 3.3.1 Embedding Algorithm

Step 1: Rearrange the given data set of images as matrix  $D$  and the secret image as matrix  $SI$ .

Step 2: Starting from the first column of matrix  $D$  i.e. image  $I_1$  calculate error matrix for first data set image i.e.  $E_{I_1}$ . If any element of  $E_{I_1}$  is less than a specific threshold (i.e. 2 per pixel on average in each window) store the corresponding index of data set image in matrix  $R$ . Calculate  $E_{I_2}$  for the remaining windows of secret image and so on until all windows find their equivalent. In case after calculating  $x$  (initially  $x=5$ ) consecutive error matrix we can not find equivalent of even single secret image window we increase error threshold level and at the same time also increase  $x$  by one i.e. if we were increasing threshold after five unsuccessful searches next time we will increase on  $x+1$  unsuccessful searches. There might be few windows that could not find their equivalent in the whole

data set. These windows are left blank and are estimated by the average of the neighbouring pixels on the receiver side. we have taken

Step 3: Third step is to divide the host image in to non-overlapping window size of  $8 \times 8$ . Then take DCT of each window using (2) and quantize with the standard Jpeg quantization matrix to calculate a non-zero coefficients block with in lower right corner of each window.

Step 4: Rescale all elements of matrix R in the range [0,10] using

$$R' = \frac{R}{255} \times 10 \quad (3.2)$$

Later embed all these values of matrix R in place of zero DCT coefficients. Size of zero coefficients block is already calculated in the previous step and is sent as a secret key with the stego-image.

Step 5: Finally take Inverse discrete transform to generate stego-image.

### 3.3.2 Retrieving Algorithm

step 1: Partition stego-image into non-overlapping window of size  $8 \times 8$  and take 2D-DCT using(2).

step 2: Retrieve data out of each window using the key received along with the stego-image and take rescale in range [0,255] using

$$R = \frac{R'}{10} \times 255 \quad (3.3)$$

step 3: Each Row in matrix R represents corresponding window of embedded secret image. Now we start regenerating secret image by the copying windows from the indexes specified by matrix R.

step 4: Rearrange data to generate secret image estimate.

### 3.4 Results and Comparisons

This section compares results of the proposed algorithm with the recently published algorithms. The proposed algorithm is tested on five different host images with three different secret images. Data set of 143 images is used to estimate secret images. Peak signal to noise ratio is used to measure quality of the stego-image and the recovered image.

#### 3.4.1 Results of the Proposed Algorithm

Five gray scale host images of size  $1024 \times 1024$  are shown in figure 3.2 (a) to (e) Baboon, Barbara, Boat, Jet and Peppers. Whereas, grayscale secret images each of size  $512 \times 512$  are shown in figure 3.3 (a) to (c) Tiger face, Tomatoes and Balloons. These three secret images were first estimated using data set of 143 images of same size. Total number of bits in each secret image before estimation is given by  $512 \times 512 \times 8 = 2097152$ . Each covert image is partitioned into non overlapping windows of size  $1 \times 4$  and thus we have total number of windows given as  $\frac{512 \times 512}{1 \times 4} = 65536$ .

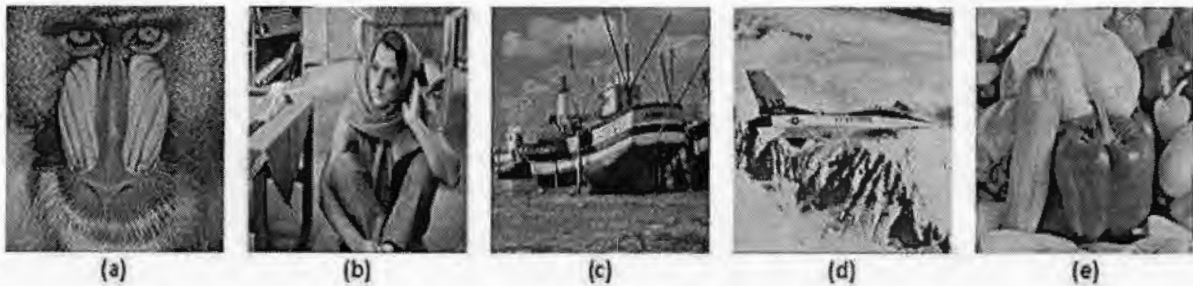


Fig 3.2 Host images

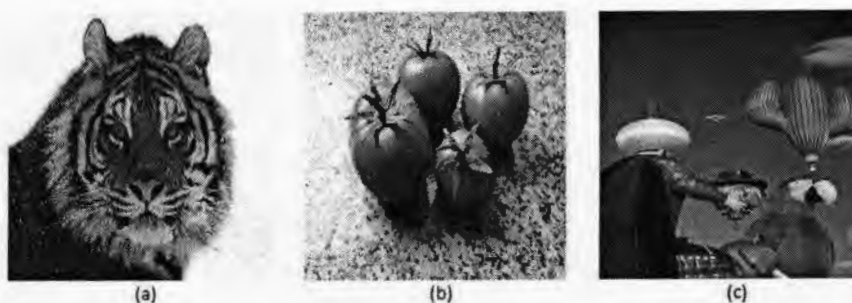


Fig 3.3 Secret Images

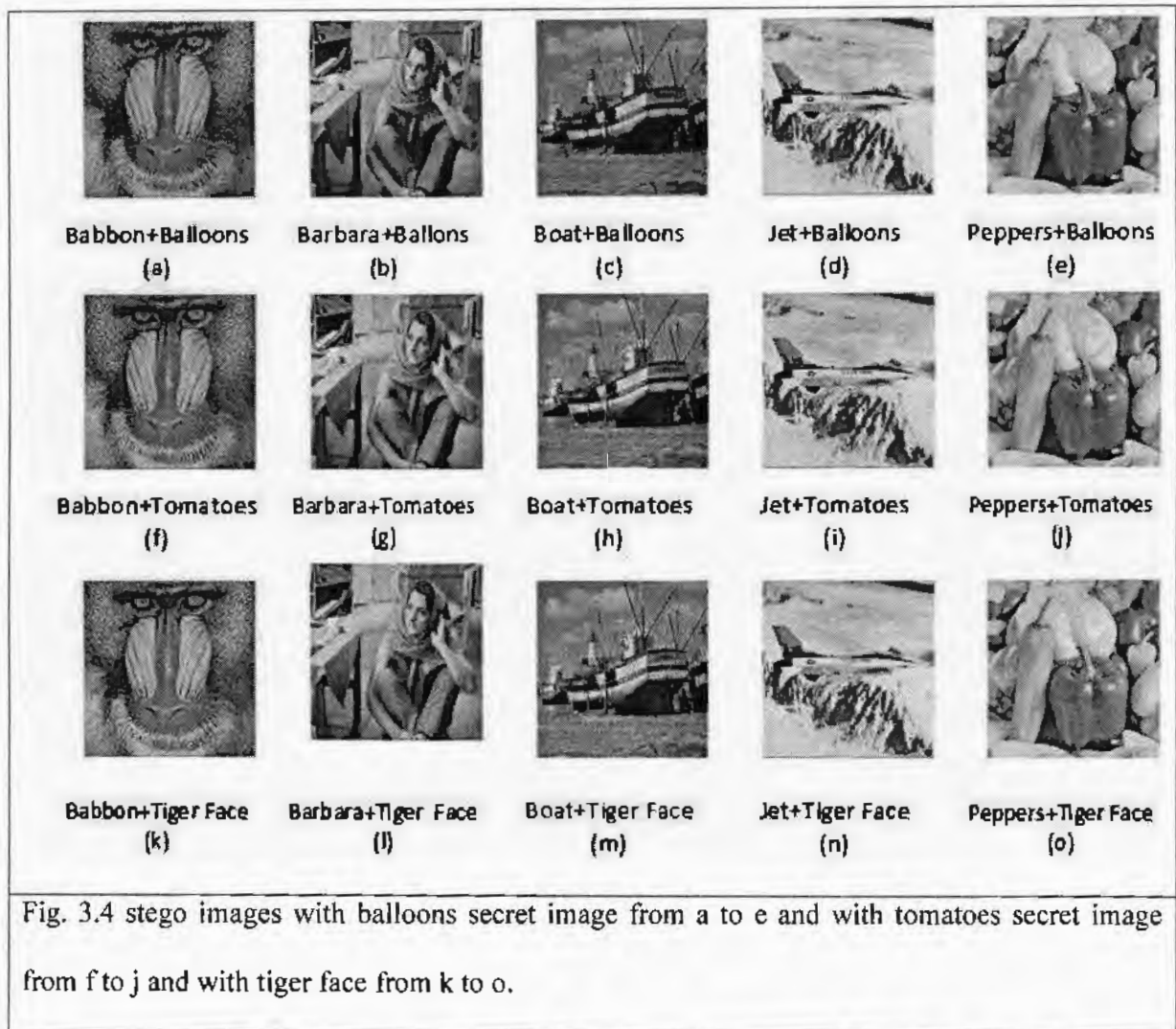
These windows are matched with non overlapping windows of data set images and a matrix R is generated. where matrix R contains 65536 rows and 2 columns, where each row represents corresponding window of secret image . First column of matrix R represent image number of dataset and as we have used 143 images so it requires only 8 bits for representation whereas, the second column represents the window with in each image .

Table 3.1 Results of the proposed Algorithm in term of bit per pixel and PSNR.

Host Image	Secret Image	$PSNR_{HS}$	$PSNR_{SS'}$	Secret Image Bits	Capacity(BPP)
Barbara	Balloons	36.37	46.27	6022110	5.74
Boat	Balloons	37.06	46.27	5442260	5.19
Jet	Balloons	36.30	46.27	6202710	5.92
Peppers	Balloons	37.50	46.27	5127280	4.89
Baboon	Balloons	36.00	46.27	5602340	5.34
Barbara	Tomatoes	36.60	46.27	6022110	5.74
Boat	Tomatoes	37.00	48.25	5442260	5.19
Jet	Tomatoes	36.20	48.25	6202710	5.92
Peppers	Tomatoes	37.40	48.25	5127280	4.89

Baboon	Tomatoes	36.40	48.25	5602340	5.34
Barbara	Tiger Face	36.40	46.12	6022110	5.74
Boat	Tiger Face	37.10	46.12	5442260	5.19
Jet	Tiger Face	36.30	46.12	6202710	5.92
Peppers	Tiger Face	37.60	46.12	5127280	4.89
Baboon	Tiger Face	36.50	46.12	5602340	5.34
Average		36.71	47.21	5679340	5.42

Each Image in data set could have 65536 windows at maximum and if we start indexing from 0 we require 16 bit to represent window with in image. Therefore, total number of bits after estimation can be given by  $24 \times 65536 = 15728462$ . Therefore, 524288 less bits are required to represent a secret image of size  $512 \times 512$ . Moreover, this gives us high data security where one can never recover the secret message until and unless he has the same dataset arranged in same order as with the sender and receiver.



These matrices  $R$  for three secret images are then embedded into five different host images to generate fifteen stego-images.  $PSNR_{HS}$  is calculated between the host and the stego-images. This gives us imperceptibility and additional layer of data security with high payload capacity. Table 1 shows result of the proposed algorithm using five different host images and three different secret images to generate 15 stego-images are shown in Fig 3.4. These stego-images are sent to the receiver along with the key where receiver first retrieves the matrix  $R$  and then regenerates the secret image.  $PSNR_{SS}$  shows

the quality of the recovered secret image over the error matrix between original secret and the recovered secret image.

### 3.4.2 Comparison

This section matches Proposed algorithm results with the schemes recently published in the literature. In [19], Lee & chen proposed a high capacity model for high capacity data embedding and achieved capacity of 4.025 bits per pixel that mean in a host image of size  $1024 \times 1024$  they can hide 4220519 bits with stego image quality of 32.57 db. Another scheme proposed in [21], achieved capacity of 2 bits per pixel with 19.41 db PSNR which mean for a host image of size  $1024 \times 1024$  they can hide 2097152 bits. Whereas, the schemes proposed in citeyang2004integer achieved capacity of 3.69 bits per pixel or 3869425 bits for selected host image size and 36.4 db PSNR. The scheme proposed in [22] achieved capacity upto 2.72 bits per pixel i.e. 2852126 bits in  $1024 \times 1024$  host image with 28.5 dn PSNR using window size of  $8 \times 8$ . This scheme was improved in [23] for achieving capacity of 3.07 bits per pixel on average i.e. 3219129 bits for host image size of  $1024 \times 1024$  with 36.4 db PSNR. However, the proposed algorithm we have achieved average injecting capacity of 5.42 bpp with 36.71 db PSNR between Host and stego images with window size of  $8 \times 8$ . Therefore, proposed algorithm has significant improvement in term of payload capacity as well as stego image quality. However, existing schemes the recovered secret image was exactly similar to the embedded secret image whereas, the proposed algorithm recovers an estimate of the secret images. However, the quality of recovered image is very high and is above 45 db in term of PSNR. Table 2 shows the evaluation of the proposed algorithm as compared to the recently published algorithms where as Fig.5 shows the recovered secret images .



Table 2 Results comparison with other schemes in term of Capacity and PSNR for grayscale

Algorithm	Capacity(bpp)	$PSNR_{HS}(db)$
Lee & Chen[14]	4.02	32.57
Rabie [30]	2.00	19.51
Yang et al.[52]	0.65	28.16
Rabie & kamel [47] ( $8 \times 8$ )	2.72	28.50
Rabie & kamel [31] ( $8 \times 8$ )	3.07	16.40
Rabie & kamel[49] ( $8 \times 8$ )	4.76	32.20
Proposed Algorithm	5.42	36.71

### 3.5 Conclusion

The proposed scheme presents a novel image steganography scheme that is based on preprocessing of secret data for reducing its payload and increasing data security . The two layer data security is achieved by first converting image into the index matrix of data set and then by embedding this matrix into the host image. Representing an image into a index matrix form reduces the size of image by 25 percent that means we can send 25 percent more data in the same host. Embedding indexes data into a host image makes indexes imperceivable. However, even if some third party manages to decode the indexes using the key sent along with the stego-image yet they cannot retrieve secret messages until and unless they have similar dataset of images ordered in similar order.

## 4 Conclusion and Future Work

The challenges in image steganography are to hide more and more data with in a least possible host. In image steganography it is measured in terms of bits per pixel. Whereas, another aspect is that we must retain the stego image quality above a certain level so that we must not lose the imperceptibility. However, improving capacity decreases quality therefore, we required algorithm that intelligently hides the secret information to attain more and more capacity while maintaining image quality measured in term of PSNR above a certain level. Moreover, there is another challenge of data security i.e. the embedded data could not easily be retrieved by an unintended users.

The work presented in this thesis proposed two different algorithms to mix secret images with in host images to generate stego-output for achieving high capacity and stego-image quality. This results in saving bandwidth and increasing imperceptibility of the host image. In the first algorithm we have used Jpeg standard quantization matrix with quality factor 50 to remove high frequency component from a  $8 \times 8$  window of dct coefficients and later secret data is embedded in non-square L shaped block. Such embedding results in achieving high capacity. Moreover, we normalize data in a specific range before embedding and this normalization range is adaptively chosen for each  $8 \times 8$  window. This results in improving Stego-image quality.

However, the second algorithm works on preprocessing of secret data to reduce its size before embedding that in other words to increase the data embedding capacity. This scheme regenerates image from a finite number of data set images. The index of those images are then embedded in to the host images. This two layer data transformation and embedding gives us extra data security. Hence, the work presented in this paper improves payload capacity, stego image quality i.e. imperceptibility and data security of the secret message with in host image

In the second paper we have used window size of  $1 \times 4$  for secret image estimation. This results in better approximation of secret image at the receiver end. However, the payload of the secret image for host images can further be reduced by taking bigger window size as an example  $1 \times 8$ . However, using such a large window size will result in decreasing quality of the recovered secret image. Therefore, Future work includes the work to find out an optimized window size for estimation of a given image or a set of secret images .

The scheme proposed in chapter 3 embeds indexes using an algorithm similar to the algorithm proposed in [31]. However, the scheme proposed in chapter 2 have achieved much better results in term of payload capacity and Stego-image quality. Therefore, the overall algorithm proposed in chapter 3 can further be improved by embedding the indexes using the scheme proposed in chapter2.

## 5 References

- [1] M. Conway, "Code Wars : Steganography , Signals Intelligence , and Terrorism," vol. 16, no. 2, pp. 45–62, 2003.
- [2] A. W. Naji, A. A. Zaidan, B. B. Zaidan, S. A. Hameed, and O. O. Khalifa, "Novel Approach for Secure Cover File of Hidden Data in the Unused Area within EXE File Using Computation between Cryptography and Steganography," vol. 9, no. 5, pp. 294–300, 2009.
- [3] L. Y. Por, B. Delina, and K. Lumpur, "Information Hiding : A New Approach in Text Steganography," 2008.
- [4] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.
- [5] M. E. Smid and K. Dennis, "The Data Encryption Standard Past and Future \*," vol. 76, no. 5, pp. 550–559, 1988.
- [6] A. Cryptography and T. Ciphers, *Foreword by Whitfield Diffie Preface About the Author Chapter 1 — Foundations Part I — Cryptographic Protocols Chapter 2 — Protocol Building Blocks Chapter 3 — Basic Protocols Chapter 4 — Intermediate Protocols Chapter 5 — Advanced Protocols. .*
- [7] M. Ghonge, A. Dhawale, and A. Tonge, "Review of steganography techniques," *Int. J. Advert Res. Comput. Electron.*, vol. 1, no. 1, pp. 35–43, 2014.
- [8] M. H. Shirali-Shahreza and M. Shirali-Shahreza, "A new approach to Persian/Arabic text steganography," in *Computer and Information Science, 2006 and 2006 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse. ICIS-COM SAR 2006. 5th IEEE/ACIS International Conference on*, 2006, pp. 310–315.
- [9] M. Shirali-Shahreza and M. H. Shirali-Shahreza, "Text steganography in SMS," in *Convergence Information Technology, 2007. International Conference on*, 2007, pp. 2260–2265.
- [10] M. Shirali-Shahreza, "Text steganography by changing words spelling," in *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, 2008, vol. 3, pp. 1912–1913.
- [11] S. Engineering, "Steganography An Analysis of Steganographic Techniques," 1998.
- [12] T. P. Nagarhalli, "A New Approach to SMS Text Steganography using Emoticons," vol. 2014, pp. 1–3, 2014.
- [13] S. Sarreshtedari and S. Ghaemmaghami, "High Capacity Image Steganography in Wavelet Domain," *Consum. Commun. Netw. Conf. (CCNC), 2010 7th IEEE*, vol. 2, pp. 1–5, 2010.
- [14] Y. K. Y.-K. Y. K. Lee and L. H. L.-H. Chen, "High capacity image steganographic

- model," *IEE Proc. - Vision, Image, Signal Process.*, vol. 147, no. 3, p. 288, 2000.
- [15] C. Chang and C. Lin, "Reversible hiding in DCT-based compressed images," vol. 177, pp. 2768–2786, 2007.
- [16] W. Luo, F. Huang, and J. Huang, "Edge Adaptive Image Steganography Based on LSB Matching Revisited," vol. 5, no. 2, pp. 201–214, 2010.
- [17] T. P. Differencing, K. Chang, C. Chang, P. S. Huang, and T. Tu, "A Novel Image Steganographic Method Using A Novel Image Steganographic Method Using Tri-way Pixel-Value Differencing," no. June, 2008.
- [18] W. Zhang, S. Wang, and X. Zhang, "Improving embedding efficiency of covering codes for applications in steganography," *IEEE Commun. Lett.*, vol. 11, no. 8, 2007.
- [19] A. Delforouzi and M. Pooyan, "Adaptive Digital Audio Steganography Based on Integer Wavelet Transform," no. 1, pp. 1–4.
- [20] H. I. Shahadi, "High Capacity and Inaudibility Audio Steganography Scheme," pp. 104–109, 2011.
- [21] Q. Qi, D. Peng, and H. Sharif, "DST approach to enhance audio quality on lost audio packet steganography," *EURASIP J. Inf. Secur.*, 2016.
- [22] S. ENCRYPTION, "Efficient method of audio steganography by modified LSB algorithm and strong encryption key with enhanced security," 2005.
- [23] N. Cvejic and T. Seppanen, "Reduced distortion bit-modification for LSB audio steganography," in *Signal Processing, 2004. Proceedings. ICSP'04. 2004 7th International Conference on*, 2004, vol. 3, pp. 2318–2321.
- [24] Y. Cao, X. Zhao, D. Feng, and R. Sheng, "Video steganography with perturbed motion estimation," in *International Workshop on Information Hiding*, 2011, pp. 193–207.
- [25] S. D. Hu and others, "A novel video steganography based on non-uniform rectangular partition," in *Computational Science and Engineering (CSE), 2011 IEEE 14th International Conference on*, 2011, pp. 57–61.
- [26] N. B. Lucena, J. Pease, P. Yadollahpour, and S. J. Chapin, "Syntax and semantics-preserving application-layer protocol steganography," in *International Workshop on Information Hiding*, 2004, pp. 164–179.
- [27] M. U. U. Celik, S. S. S. S. Member, G. Sharma, S. S. S. S. Member, A. M. Tekalp, E. Saber, S. S. S. S. Member, G. Sharma, and S. S. S. S. Member, "Lossless generalized-LSB data embedding," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 253–266, 2005.
- [28] C.-C. Chang, J.-Y. Hsiao, and C.-S. Chan, "Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy," *Pattern Recognit.*, vol. 36, no. 7, pp. 1583–1595, 2003.
- [29] T. Rabie, "Frequency-domain data hiding based on the Matryoshka principle," *Int. J. Adv.*

*media Commun.*, vol. 1, no. 3, pp. 298–312, 2007.

- [30] T. Rabie and I. Transportation, "High-Capacity Steganography," *Image Signal Process. (CISP), 2013 6th Int. Congr.*, vol. 2, no. Cisp, pp. 858–863, 2013.
- [31] T. Rabie and I. Kamel, "High-capacity steganography: a global-adaptive-region discrete cosine transform approach," *Multimed. Tools Appl.*, pp. 1–21, 2016.
- [32] C. C. Y. C. C. Lin, "A data embedding scheme for color images based on genetic algorithm and absolute moment block truncation coding," pp. 321–331, 2009.
- [33] C.-C. Chang, W.-L. Tai, and C.-C. Lin, "A reversible data hiding scheme based on side match vector quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 10, pp. 1301–1308, 2006.
- [34] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proc. IEEE*, vol. 86, no. 6, pp. 1064–1087, 1998.
- [35] I. Banerjee, S. Bhattacharyya, and G. Sanyal, "Novel text steganography through special code generation," *Int. Conf. Syst. Cybern. Informatics*, pp. 298–303, 2011.
- [36] N. F. Johnson and S. Jajodia, "Exploring steganography: Seeing the unseen," *IEEE Comput.*, vol. 31, no. 2, pp. 26–34, 1998.
- [37] R. Cowan, A. Mili, H. H. Ammar, and A. McKendall, "Software Engineering Technology Watch," *IEEE Softw.*, vol. 19, no. 4, pp. 123–129, 2002.
- [38] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, "Digital image steganography: Survey and analysis of current methods," *Signal Processing*, vol. 90, no. 3, pp. 727–752, 2010.
- [39] G. Brisbane, R. Safavi-Naini, and P. Ogunbona, "High-capacity steganography using a shared colour palette," *IEE Proc. - Vision, Image, Signal Process.*, vol. 152, no. 6, p. 787, 2005.
- [40] C. H. Yang, C. Y. Weng, S. J. Wang, and H. M. Sun, "Adaptive data hiding in edge areas of images with spatial LSB domain systems," *IEEE Trans. Inf. Forensics Secur.*, vol. 3, no. 3, pp. 488–497, 2008.
- [41] T. Rabie, "Digital Image Steganography: An FFT Approach," *Commun. Comput. Inf. Sci.*, vol. 294 PART 2, pp. 217–230, 2012.
- [42] B. Yang, M. Schmucker, W. Funk, C. Busch, and S. Sun, "Integer DCT-based reversible watermarking for images using companding technique," *Proc. SPIE, Secur. Steganography, Watermarking Multimed. Contents*, vol. 5306, no. July 2016, pp. 405–415, 2004.
- [43] C.-C. Lin and P.-F. Shiu, "High Capacity Data Hiding Scheme for DCT-based Images," *J. Inf. Hiding Multimed. Signal Process.*, vol. 1, no. 3, pp. 220–240, 2010.
- [44] M. Saidi, H. Hermassi, R. Rhouma, and S. Belghith, "A new adaptive image



- steganography scheme based on DCT and chaotic map," *Multimed. Tools Appl.*, 2016.
- [45] P.-Y. Chen and H.-J. Lin, "A DWT Based Approach for Image Steganography," *Int. J. Appl. Sci. Eng. Int. J. Appl. Sci. Eng.*, vol. 4, no. 4, pp. 275–290, 2006.
- [46] T. Rabie and I. Kamel, "On the embedding limits of the discrete cosine transform," *Multimed. Tools Appl.*, vol. 75, no. April, pp. 5939–5957, 2015.
- [47] T. Rabie and I. Kamel, "On the embedding limits of the discrete cosine transform," *Multimed. Tools Appl.*, vol. 75, no. 10, pp. 5939–5957, 2016.
- [48] T. Rabie and I. Kamel, "On the embedding limits of the discrete cosine transform," *Multimed. Tools Appl.*, no. January 2015, pp. 5939–5957, 2016.
- [49] T. Rabie and I. Kamel, "Toward optimal embedding capacity for transform domain steganography: a quad-tree adaptive-region approach," *Multimed. Tools Appl.*
- [50] K.-L. Chung, C.-H. Shen, and L.-C. Chang, "A novel SVD-and VQ-based image hiding scheme," *Pattern Recognit. Lett.*, vol. 22, no. 9, pp. 1051–1058, 2001.
- [51] W. Sun, Z. M. Lu, Y. C. Wen, F. X. Yu, and R. J. Shen, "High performance reversible data hiding for block truncation coding compressed images," *Signal, Image Video Process.*, vol. 7, no. 2, pp. 297–306, 2013.
- [52] C. Busch, "Integer DCT-based reversible watermarking for images using companding technique Integer DCT-based Reversible Watermarking for Images Using," no. July 2016, 2004.
- [53] M. Iwata, K. Miyake, and A. Shiozaki, "Digital steganography utilizing features of JPEG images," *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, vol. 87, no. 4, pp. 929–936, 2004.
- [54] C.-C. Lin and P.-F. Shiu, "DCT-based reversible data hiding scheme," in *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication*, 2009, pp. 327–335.
- [55] T. Liu and Z. Qiu, "A DWT-based color image steganography scheme," in *Signal Processing, 2002 6th International Conference on*, 2002, vol. 2, pp. 1568–1571.
- [56] C. C. Lin and P. F. Shiu, "DCT-based reversible data hiding scheme," *J. Softw.*, vol. 5, no. 2, pp. 214–224, 2009.
- [57] "A PALETTE-BASED IMAGE STEGANOGRAPHIC METHOD USING COLOUR QUANTISATION Xuefeng Wang Chang-Tsun Li Coventry , CV4 7AL , United Kingdom Email : { csubao , yao , ctli } @ dcs . warwick . ac . uk," 2005.

## 6 Appendixes

### 6.1 Appendix A

#### 6.1.1 First Contribution Embedding Algorithm

Following code takes cover image and secret image to generate stego-image Rec

```
%%  
clear all  
clc  
tic  
k=0;  
Q=[1/16 1/11 1/10 1/16 1/24 1/40 1/51 1/61;  
    1/12 1/12 1/14 1/19 1/26 1/58 1/60 1/55;  
    1/14 1/13 1/16 1/24 1/40 1/57 1/69 1/56;  
    1/14 1/17 1/22 1/29 1/51 1/87 1/80 1/62;  
    1/18 1/22 1/37 1/56 1/68 1/109 1/103 1/77;  
    1/24 1/35 1/55 1/64 1/81 1/104 1/113 1/92;  
    1/49 1/64 1/78 1/87 1/103 1/121 1/120 1/101;  
    1/72 1/92 1/95 1/98 1/112 1/100 1/103 1/99];  
Actual=imread('balloons1.jpg');  
Original = rgb2ycbcr(Actual);  
factor1=double(max(max(Original(:,:,1))));  
factor2=double(max(max(Original(:,:,2))));  
factor3=double(max(max(Original(:,:,3))));  
Original=im2double(Original);  
factor11=double(max(max(Original(:,:,1))));
```



```

factor22=double(max(max(Orignal(:,:,2))));
factor33=double(max(max(Orignal(:,:,3))));

for u=1:3
Orignal1=Orignal(:,:,u);
[Row,Col]=size(Orignal1);
ctm=rem(Row,8);
if ctm~=0
    yb=zeros(8-ctm,Col);
    Orignal1=[Orignal1;yb];
end
[Row,Col]=size(Orignal1);

ctm=rem(Col,8);
if ctm~=0
    zb=zeros(Row,8-ctm);
    Orignal1=[Orignal1,zb];
end
[Row,Col]=size(Orignal1);
for i=1:Row
    for j=1:Col
        if Orignal1(i,j)==0
            Orignal1(i,j)=1;
        end
    end
end

```

```

end
for i=1:8:Row
    for j=1:8:Col
        if u==1
            Transformed(i:i+7,j:j+7)=dct2(factor1*Original1(i:i+7,j:j+7));
            CT1(i:i+7,j:j+7)=Transformed(i:i+7,j:j+7);
            one(i:i+7,j:j+7)=round(Transformed(i:i+7,j:j+7).*Q);
        end
        if u==2
            Transformed(i:i+7,j:j+7)=dct2(factor2*Original1(i:i+7,j:j+7));
            CT2(i:i+7,j:j+7)=Transformed(i:i+7,j:j+7);
            two(i:i+7,j:j+7)=round(Transformed(i:i+7,j:j+7).*Q);
        end
        if u==3
            Transformed(i:i+7,j:j+7)=dct2(factor3*Original1(i:i+7,j:j+7));
            CT3(i:i+7,j:j+7)=Transformed(i:i+7,j:j+7);
            three(i:i+7,j:j+7)=round(Transformed(i:i+7,j:j+7).*Q);
        end

        Transformed(i:i+7,j:j+7)=round(Transformed(i:i+7,j:j+7).*Q);
    end
end

buffer1=ones(1,1);
Pointer=1;
for i=1:8:Row

```

```

    for j=1:8:Col
p=Transformed(i:i+7,j:j+7);
var=1;exit=0;
sum1=0;
while var <= 8
if abs(p(var,var))>0
    var=var+1;
else if abs(p(var,var))==0
    buffer1=[buffer1;var];
    var=9;
    end
end
end
end
end
end
buffer1=buffer1-1;
s=length(buffer1);
% for adaptive k comment this part
for i=1:s
    if buffer1(i) <= k
        buffer1(i)= k+1;
    end
end
end
if u==1
    buffer2=buffer1(2:s,:);

```

```

else if u==2
    buffer3=buffer1(2:s,:);
else if u==3
    buffer4=buffer1(2:s,:);
end
end
end
end
Simage=double(imread('flower.bmp'));
Simage=crop(Simage,buffer2,buffer3,buffer3);
[m,n,v]=size(Simage(:, :, 1))
Simv=[m,n]
for j=1:3
    himage1=Simage(:, :, j);
    for i=1:m
        Simv=[Simv,himage1(i,:)];
    end
end
[P1,P2,P3]=phase(CT1,CT2,CT3);
CT1=abs(CT1);CT2=abs(CT2);CT3=abs(CT3);
[uch1,ak1,next,pc1]=gare2(CT1,Simv,buffer2,1);
next1=next
[uch2,ak2,next,pc2]=gare2(CT2,Simv,buffer3,next);
next2=next
[uch3,ak3,next,pc3]=gare2(CT3,Simv,buffer4,next);

```

```

next3=next;
ak1=ak1(2:length(ak1),1);ak2=ak2(2:length(ak2),1);ak3=ak3(2:length(ak3),1);
buffer21=buffer2.*buffer2;
buffer21=8*(64-buffer21);
t1=sum(buffer21);
buffer31=buffer3.*buffer3
buffer31=8*(64-buffer31);
t2=sum(buffer31);
buffer41=buffer4.*buffer4
buffer41=8*(64-buffer41);
t3=sum(buffer41);
totalsm=t1+t2+t3;
[m,n,v]=size(Actual)
[s1,s2,s3]=RLE(buffer2,buffer3,buffer4);
total=m*n+length(buffer2)+length(buffer3)+length(buffer4)+length(ak1)+length(ak2)+length(ak
3);% this may change if i use m and n in some other called program
Aratio=totalsm/total;
uch1=uch1.*P1;uch2=uch2.*P2;uch3=uch3.*P3;
for i=1:8:m
    for j=1:8:n
        Rec1(i:i+7,j:j+7)=idct2(uch1(i:i+7,j:j+7));
        Rec2(i:i+7,j:j+7)=idct2(uch2(i:i+7,j:j+7));
        Rec3(i:i+7,j:j+7)=idct2(uch3(i:i+7,j:j+7));
    end
end
end

```

```

Rec11=Rec1/factor1;Rec22=Rec2/factor2;Rec33=Rec3/factor3;
Rec=cat(3,Rec11,Rec22,Rec33);
Recc=ybcr2rgb(Rec);
figure(2);
imshow(Recc);
[M,N] = size(Rec1);
error =Original(:,:,1)-Rec11;
MSE1 = sum(sum(error .* error)) / (M * N);
PSNR1 = 20*log10((factor11)/sqrt(MSE1));
[M,N] = size(Rec2)
error =Original(:,:,2)-Rec22;
MSE2 = sum(sum(error .* error)) / (M * N);
PSNR2 = 20*log10((factor22)/sqrt(MSE2))
[M,N] = size(Rec3)
error =Original(:,:,3)-Rec33;
MSE3 = sum(sum(error .* error)) / (M * N);
PSNR3 = 20*log10((factor33)/sqrt(MSE3));
APSNR=(PSNR1+PSNR2+PSNR3)/3;
imwrite(Rec,'laeeq.jpg','Quality',100);
toc
%%

```

### 6.1.2 First Contribution Retriving Algorithm

This algorithm takes stego-image Rec and regenerates the secret image.

```

Rec1=Rec(:,:,1)*factor1;Rec2=Rec(:,:,2)*factor2;Rec3=Rec(:,:,3)*factor3;
[sm1]=rec1(buffer2,ak1,Rec1);

```

```

row=sm1(1);col=sm1(2);
sm=sm1(3:length(sm1));
size1=row*col
if length(sm1)<length(Simv);
[sm2]=rec1(buffer3,ak2,Rec2);
sm=[sm;sm2];
end
if (length(sm1)+length(sm2))<length(Simv)
    [sm3]=rec1(buffer4,ak3,Rec3);
    sm=[sm;sm3];
end
c=1;
image1=zeros(row,col);image2=zeros(row,col);image3=zeros(row,col);
for i=1:row
    if c+col<length(sm)
        image1(i,:)=sm(c:c+col-1);
        c=c+col;
    end
end
for i=1:row
    if c+col<length(sm)
        image2(i,:)=sm(c:c+col-1);
        c=c+col;
    end
end

```

```
end
for i=1:row
    if c+col<length(sm)
        image3(i,:)=sm(c:c+col-1);
        c=c+col;
    end
end
image1=uint8(image1);image2=uint8(image2);image3=uint8(image3);
image=cat(3,image1,image2,image3);
figure(1);
imshow(image);
```