Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be "used for any purpose other than private study, scholarship, or research." If a, user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of "fair use" that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select "Pages from: first page # to: last page #" on the print dialog screen

ABSTRACT

A STUDY OF STEGANOGRAPHY AND STEGANALYSIS

by
Nisant Nalla

Steganography is the art of hiding the fact that communication is taking place, by hiding information in other information. Steganalysis is the study of methods to detect the presence of hidden messages. The Steganography and Steganalysis are very much interdependent. The people in the Steganography domain try to develop stronger methods to hide data in different media. This spurs the discovery of new steganalysis methods which can break these Steganography methods. This is vital for the development of these fields. The internet has been growing by leaps and bounds. There is an increasing need for information security. It has been rumoured that some of the anti social organizations have been using Internet to send and receive messages covertly. This is a serious problem before all the law enforcement agencies. This is one reason why steganographic and steganalytic techniques are being studied. The applications for Steganography are not just limited to military purposes. It can be used by businesses to tackle with copyright infringements.

This work is a study of the some of the Steganography and Steganalysis techniques which are considered to be the best. In this work experiments were done with Steganography methods F5, Outguess and the Model based methods which are considered the most secure steganographic schemes at present. The Steganalysis methods used here are the NJIT's 39D and 78D methods which are based on statistical moments of characteristic functions of wavelet subbands for the test images and prediction error images.

A STUDY OF
STEGANOGRAPHY AND STEGANALYSIS

by
Nisant Nalla

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Department of Electrical and Computer Engineering

January 2006

APPROVAL PAGE

A STUDY OF STEGANOGRAPHY AND STEGANALYSIS

Nisant Nalla

Dr. Yun-Qing Shi, Thesis Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Frank Y. Shih, Committee Member — Date
Professor of Computer Science, NJIT

Dr. Mengchu Zhou, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Nisant Nalla

Degree: Master of Science

Date: January 2006

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2006

- Bachelor of Technology in Electronics and Communication Engineering,
  Jawaharlal Nehru Technological University, Hyderabad, India, 2002

Major: Electrical Engineering

To my parents

ACKNOWLEDGMENT

I would like to express my deepest appreciation to Dr. Yun-Qing Shi, who not only served as my research supervisor, providing valuable and countless resources, insight, and intuition, but also constantly gave me support, encouragement, and reassurance. Special thanks are given to Dr. Mengchu Zhou and Dr. Frank Y. Shih for actively participating in my committee.

I am also grateful to Chunhua Chen and Wen Chen for their cooperation in the research.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

Chapter **Page**

CHAPTER 1

INTRODUCTION

Steganography is derived from the greek words 'steganos' and 'graphein', 'steganos' meaning to cover or to hide and 'graphein' meaning to write and so steganography literally means covered writing.

1.1 Overview of Steganography

The origins of covert communication can be traced back to 500BC in Greece. It is mentioned in the writings of the Greek historian Heredotus. One of the stories stated that King of Susa shaved the head of one of his slaves and wrote a secret message on his scalp. The slave was sent to the King Aristogoras in Miletus and the message was passed on undetected when the person's hair grew back. In another story which also came from Herodotus, which claims that a soldier named Demeratus needed to send a message to Sparta that Xerxes intended to invade Greece. Now wax-covered tablets were used for sending the message. Wax was removed from the tablet, the secret message was written on the underlying wood, recovered the tablet with wax to make it appear as a blank tablet and finally sent the message without being detected. It is also learnt that the Romans used invisible inks, which were based on natural substances such as fruit juices. The steganographic techniques were being used for covert communication not just by the Romans and the Greeks but by many different people in different parts of the world. In the recent past a lot of advancement has taken place in Steganography. Many new steganographic methods have been developed and are being used all over the world.

Let us compare Steganography and Cryptography. Steganography is the art and science of hiding the very existence of the message in a communication. In contrast the focus of Cryptography is encrypting the message so that it cannot be accessed or read by users who it is not intended to. Here the enemy is allowed to detect, intercept and modify messages without violating certain rules. The objective of cryptography is to make it difficult to read the message. The goal of Steganography is to hide messages inside other harmless messages in a way that does not allow any enemy to even detect that there is a second message present. The communication media for steganography can be any audio files, video files, image files.

1.2 Prisoner Problem

The Steganography concepts can be explained in terms of a hypothetical prisoner problem proposed by Simmons [2]. There are two prison inmates, Alice and Bob who wish to communicate with each other. Alice wishes to send a secret message to Bob so that nobody can understand the message. Wendy is the prison warden. Alice and Bob try to communicate with each other and Wendy tries to intercept them. There can be 2 cases here. First case in which Wendy does not know the presence of secret message and also the method used; and only Alice and Bob share this fact. Second case is wherein both the inmates Alice and Bob communicate using a method which is unknown even to Wendy but they share a key. Here Wendy cannot know the message if she doesn't know the key, but she may suppress the entire message is she is able to detect the presence of secret message.

The growth of information hiding or Steganography is being driven by many factors. One factor can be the need to avoid copyright infringement and protect intellectual property rights. As the Internet is growing by leaps and bounds it is becoming more difficult to protect intellectual property and enforce copyright laws. The use of digital watermarks provides a way to insert a copyright notice into a document or image. The watermark is a small image or text that is repeated through out the cover media. The watermark is a unique digital trademark which can be used to trace back the original copy. So the copyright violators can be prosecuted. This Steganography can be used for covertly sending messages within a cover media.

1.3 Steganalysis

Steganalysis is the art of finding the presence of any covert message in a transmission media. Though the first goal of steganalysis is detection, there can be additional goals such as disabling, extraction, and confusion. While detection, disabling, and extraction are self-explanatory, confusion involves replacing the intended embedded file [3]. Steganography and Steganalysis are complementary to each other. The growth of more difficult steganographic methods leads to development on steganalytic methods and vice versa is also true. The steganalysis attacks can be:

- Visual Attacks
- Statistical Attacks

1.3.1 Visual Attacks

It is almost impossible to detect steganographic content when simply viewing the whole image on the display, things change if we consider only the LSBs of the particular image. This can be done through human observers detecting minute changes between a cover file and a stego file or it can be done by a machine. For palette-based images if the embedded file was inserted without first ordering the cover file palette according to color, then dramatic color shifts can be found in the stego file. Additionally, since many steganography tools take advantage of close colors or create their own close color groups, many similar colors in an image palette may make the image become suspect [5]. By filtering images as described by Westfield and Pfitzmann in [4], the presence of an embedded file can become obvious to the human observer.

1.3.2 Statistical Attacks

It can found if LSB substitution was used by analyzing changes in an image's close color pairs. Close color pairs consist of two colors whose binary values differ only in the LSB. The sum of occurrences of each color in a close color pair does not change between the cover file and the stego file. These statistical techniques take advantage of the fact that the embedding process alters the original statistics of the cover file and in many cases these first order statistics will show trends that can raise suspicion of Steganography.

1.4 Media

The media used in Steganography can be almost all digital file formats. The most common files in which data is embedded are

- Text
- Audio
- Video
- Images
- TCP/IP Protocol

Most of the Steganographic techniques try to make use of the redundancies in the cover data. Text has been used as an important media for information hiding. And hiding data in text is the simplest of all the techniques. The amount of redundant data contained in text files is very less when compared to other file formats. And also larger the size of the cover media, larger is amount of information that can be hidden in the cover media. So, the focus of media used for Steganography has moved from text to other media files. After the internet revolution, due to the vast number of images available on the internet and the large amount of redundant bits in images, the most popular data used for Steganography are image files.

Embedding data into an image can be accomplished by either of two techniques:

- Image Domain-based Methods
- Transform Domain-based Methods

1.4.1 Images and Compression

There are many applications to hide information into images. Since images are present in vast numbers on the internet, they can be used without arousing suspicion. A picture on a website can be used to convey information secretly. This picture could even contain

different information at different times, without any visible differences at all. Due to this, many Steganography and steganalysis techniques for images are being developed and are discussed here in detail. Due to storage constraints and bandwidth constraints for transmitting images most of the the image file formats are compressed in size. Different file formats use different compression algorithms to accomplish this. Based on the different compression schemes used the digital image file types can be classified as: lossless and lossy images.

Lossless Compression

A lossless compression algorithm discards no information. It looks for more efficient ways to represent an image, while making no compromises in accuracy. The compressed image will contain all the information contained in the original uncompressed one.

TIFF:

TIFF is a lossless format which uses a lossless compression algorithm called LZW. The TIFF files support 32-bit color depth.

PNG:

PNG is also a lossless storage format. PNG stands for Portable Network Graphics.

GIF:

GIF creates a table of up to 256 colors. If the image has fewer than 256 colors, GIF can render the image exactly. When the image contains more than 256 colors the software which creates GIF images tries to approximate the image colors with the 256 color palette. GIF is used extensively on the web.

RAW:

RAW is an another lossless image format.

Lossy Compression

JPEG:

The JPEG (Joint Photographic Experts Group) image files are a lossy format. The DOS filename extension is JPG, although other operating systems may use JPEG.

1.4.2 Image Domain-based methods

Image Domain-based methods manipulate the Least Significant Bit (LSB) of the cover image. These are also known as Bit Wise Methods.

- LSB Insertion

In this method the leftmost bit of each pixel in the cover image is replaced with one bit from the secret message. Because the LSB can only contain zeros and ones, approximately half the time the bit does not need to be altered in order to embed the data from the secret message. The least significant bit or the the 8th bit of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue colour components can be used, since each color is represented by a byte. This implies we can store 3 bits of secret message per pixel since we can only modify LSB's.

- LSB Insertion in Palette-based Images

Palette-based images, for example GIF images, are another popular image file format commonly used on the internet. A GIF image cannot have a bit depth greater than 8, thus the maximum number of colours that a GIF can store is 256. GIF images can also be used for LSB Steganography.

Some of the Image Domain tools are :

- Hide and Seek
- Mandelsteg
- Steganos
- S-TOOLS
- White Noise Storm
- StegoDos

1.4.3 Transform Domain-based methods

In Transform Domain methods, the image is converted into transform domain using transforms such as the Discrete Cosine Transform or wavelet transform and then information is hidden in significant areas of the image. Steganographic methods which try to make changes in transform domain are more robust and are difficult to break.

Some of the Transform Domain tools used for Image Steganography are :

- Jpeg-Jsteg
- JPHide
- Outguess
- F5
- Model-based Method 1(MB1)

- Model-based Method 2(MB2)
- PictureMarc
- SysCop

This thesis is organized as follows. Some of the modern steganographic techniques have been discussed in the chapter 2. Section 2.1 deals with the model-based methodology for Steganography and model-based Steganography methods MB1 and MB2. The different steganalysis techniques have been covered in chapter 3.

CHAPTER 2

MODERN STEGANOGRAPHIC TECHNIQUES

This chapter deals with the steganographic techniques which are studied in this work. The steganographic techniques discussed here are the Outguess, F5 and the Model-based methods.

2.1 Model-based Methods

This method is based on statistical modeling and information theory. It utilizes a statistical model of the cover media.

Any steganographic method is perfectly secure if there is no statistical difference between the class of cover images and the class of stego images. Most of the Steganography methods which are currently being used concentrate on ways to flip least significant bits, but this method tries to best model the cover data.

The method [14] is described here. If x is an instance of a class of potential cover media. x can be considered as instance of a random variable X . x is divided into x_α and x_β where x_α is unchanged and x_β will be replaced with x_β' which has the encoded message. Using the model distribution P_X the conditional distribution $P_{X\beta/X\alpha}(X_\beta/X_\alpha = x_\alpha)$ is evaluated.

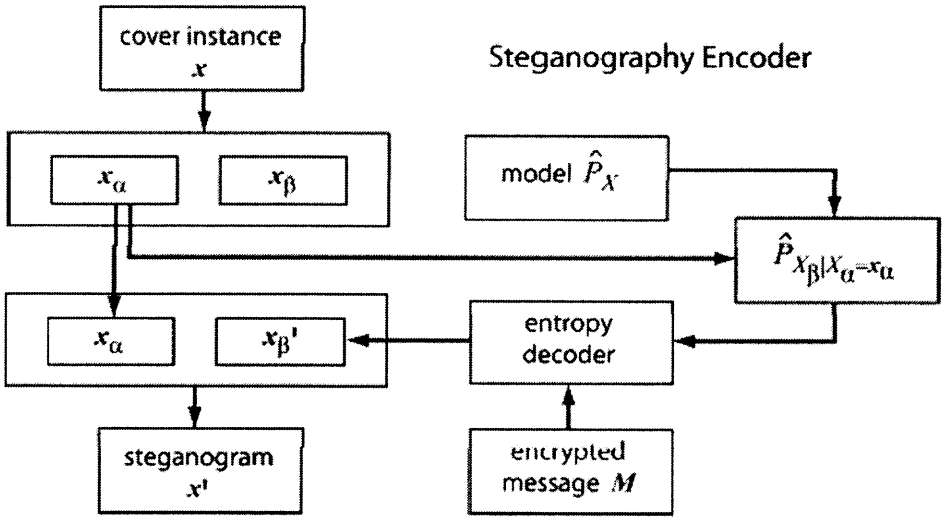

Fig 2.1 Encoder for Model-based Steganography

The above figure shows the Encoder for a Model-based Steganography method. A encrypted and compressed message M is given. The encrypted message M and the model distribution calculated are fed into an entropy decoder. M is decompressed according to the model distribution using the entropy decoder. This gives rise to x_β' which is then combined with x_α to form x' . In a similar fashion, the decoding takes place. At the decoder, the x_α part is fed into the model which is used to compute the condition distribution. Thus, the same model is given to the entropy encoder that was fed into the entropy decoder. The entropy decoder gives the encrypted message. The original message can be retrieved from the encrypted message if we have the key. The decoder for the Model-based Steganography method is shown in figure 2.

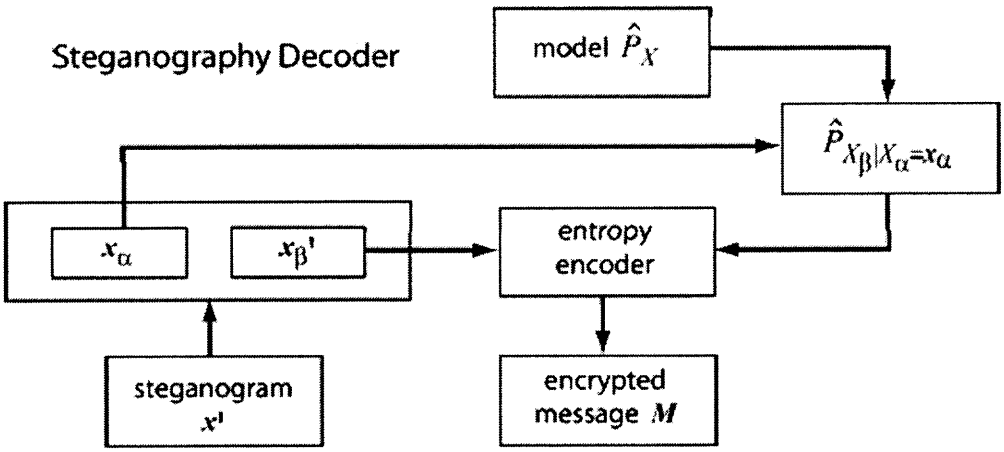

Fig 2.2 Decoder for Model-based Steganography

2.1.1 Model-based Steganography applied to JPEG: Model-based method 1 (MB1)

In JPEG compression of images, the image is divided into 8X8 blocks and Discrete Cosine Transform is applied on each block. The DCT coefficients are then quantized and encoded using the different encoding techniques. The DC coefficients are encoded using differential encoding and AC coefficients are encoded using run-length encoding for each block. Now all the coefficients are encoded using Huffman encoding. During the run length encoding, the zero valued coefficients are skipped for the encoding.

In the MB1 algorithm, first the low precision histograms each type of AC coefficient for a cover image x are calculated. Each coefficient value is represented by a histogram bin index and a symbol which indicates its offset within the bin. The bin indices for all the coefficients comprise x_α, which will remain unchanged, and the bin offsets will comprise x_β which will be changed to encode our message. These offset symbols, and their respective probabilities are passed to an arithmetic entropy decoder along with the

message we wish to embed in the cover image. The offset symbols returned by the entropy decoder are x_β' which are combined with x_α to calculate x'. The order in which coefficients are used for encoding the message is determined by computing a pseudo-random permutation seeded by a key. A similar process is used to decode the message from the steganogram, except that the bin offset symbols x_β' in the steganogram are passed along with the symbol probabilities to an arithmetic encoder.

2.1.2 Model-based method 2: MB2

It has been shown by Friedrich et al. showed in their paper "Attacking Outguess" that JPEG steganography methods can be foiled easily by using blockiness attacks using a simple measure. Blockiness is the measure of the discontinuities between adjacent 8X8 JPEG blocks. This problem is specific to methods which use JPEG data.

$$B = \sum_{i=1}^{[(M-1)/8]} \sum_{j=1}^{N} | g_{8i,j} - g_{8i+1,j} | + \sum_{j=1}^{[(N-1)/8]} \sum_{i=1}^{M} | g_{i,8j} - g_{i,8j+1} |$$

The above formula given in Friedrich's paper is used to calculate the blockiness measure. And an estimate of the message length can be obtained from this blockiness measure. In order to reduce this blockiness measure the model-based method, MB2 has been developed.

For method MB2, a message is embedded in the same manner as MB1, but at least half of the coefficients are reserved for the purpose of reducing blockiness artifacts. These coefficients which are not used to encode the message are then adjusted within the limits

imposed by the embedding step size to reduce the blockiness to the amount present in the original image. First the adjustment that has to be made to each pixel to reduce the blockiness measure is calculated. Then DCT is applied on the matrix which contains these changes. The resulting blockiness is calculated assuming all the non-message coefficients are changed using the adjustments calculated. Based on this the number of coefficients that have to be changed to achieve the desired blockiness i.e., blockiness of the original image and accordingly coefficients are changed. The last two steps are repeated until the desired blockiness measure is obtained.

2.2 F5

Any Steganography technique tries to embed messages in a media so that the attacker does not have any clue of the hidden message. In this process many changes are made to the original cover media. These changes may be immune to visual attacks, but they are easily discovered by a statistical attack. The F5 technique which I'm going to discuss here now tries to reduce the number of the necessary changes to be made to the original media.

This F5 algorithm was proposed by Westfeld [10]. This is based on a series of algorithms JSteg, F3, F4. The heart and soul of this algorithm lies in Matrix encoding. As I've mentioned in the previous sections the JPEG compression. In JPEG compression, the image is read and 8X8 block DCT is applied on it. The DCT coefficients undergo the quantization process. The quantized DCT coefficients are then encoded (within each block) according to the different encoding techniques. And finally the entire matrix is

encoded using Huffman coding. The F5 algorithm is applied between the quantization and the Huffman coding stages. This algorithm can be explained using the figure below.

In F5 algorithm, the block DCT is calculated for the given input image. Then the DCT coefficients undergo permutative straddling. The resulting coefficients are passed through Huffman encoding process.

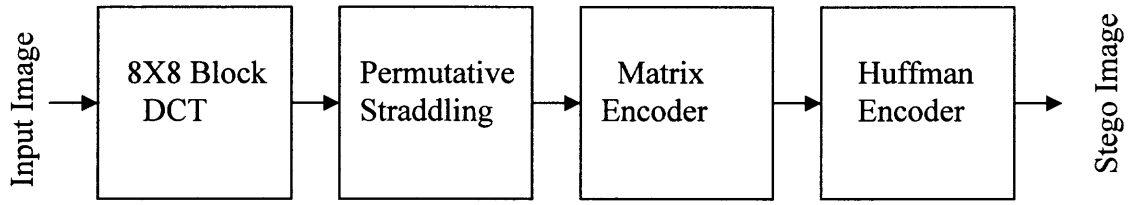

Fig.2.3 F5 Steganography method

2.2.1 Permutative Straddling

The changes which are made to an image by a Steganography technique are mostly concentrated in the beginning of the file and the unused part of the media is at the end. To avoid these changes being visible, an element of randomness is added to these changes. Steganographic algorithms try to scatter the hidden messages over the entire carrier medium to make the changes introduced in the medium uniformly distributed. This technique is called Permutative straddling. The straddling can be done perfectly if the capacity of the carrier medium is exactly known.

In this method all the coefficients are shuffled according to a random permutation. This permutation depends on a key which can be a password. The resulting coefficient matrix is sent to the matrix encoder.

2.2.2 Matrix Encoding

This technique has been proposed by Ron Crandall [11]. This technique significantly increases the embedding efficiency. This can be explained using an example. A cover message is divided into smaller blocks called cells. In our example let us consider the cover image to be dividsed into blocks of 3 cells each and two bits of information is stored in each block. Let a, b, c be the bits which can be modified. And the message bits to be embedded be x, y. The minimum no. of changes that can be made to the given three bits can be calculated.

x = a xor c, y = b xor c => change nothing

x ≠ a xor c, y = b xor c => change a

x = a xor c, y ≠ b xor c => change b

x ≠ a xor c, y ≠ b xor c => change c

In the first case if the message bit x is equal to (a xor c) and message bit y is equal to (b xor c) ,none of the bits a, b, c need to be changed . In the second case, if the message bit x is not equal to (a xor c) and message bit y is equal to (b xor c) , the bit a is changed. Similarly in the third case, if the message bit x is equal to (a xor c) and message bit y is not equal to (b xor c) , the bit b is changed . If both the message bits are wrong, bit c is changed. Here in this method we can see that not more than one bit is changed in any case. The method described here is called as a (1, 3, 2) design.

The generalized form of this arithmetic coding can be shown as (d_{max}, n, k) ,an ordered triple. Here a code word with n places will be changed in not more than d_{max} places to embed k bits. F5 implements matrix encoding only for d_{max} =1 i.e., the code word is (1, n, k) . The code words will have length $k= 2^k -1$.

The change density and the embedding rate can be calculated as

$$D(k) = 1/(n+1) = 1/2^k$$

$$R(k) = k/n = k/(2^k -1)$$

The average number of bits we can embed per change or the Embedding Efficiency can calculated as:

$$W(k) = R(k) / D(k)$$

$$= [2^k / (2^k -1)] . k$$

It has been shown that for one pixel cells,(using the above mentioned (1, 3, 2) code) this method has an embedding rate of 67% and a change density of 25% . And for (1, 7, 3) it is seen that embedding rate is 42% and change density is 12.5% . From the above formula it is seen that the embedding efficiency of the (1,n,k) code is always greater than k . After this Matrix encoding, the coefficients are Huffman coded to complete the JPEG compression.

Huffman Coding is an entropy encoding algorithm used for lossless data compression. In Huffman coding, long strings of binary digits are replaced by shorter codewords. Most frequently occuring characters are converted to shortest bit strings and the least frequent

ones are converted to longest. This algorithm involves two steps. In the first block of data is analyzed and a tree model based on its contents is created. In the second step the data is compressed using the model created.

2.3 Outguess

Outguess is an embedding method proposed by Niels Provos. This method tries to minimize modifications to the cover medium. The statistical properties of the cover media are preserved to avoid attack by statistical tests. In this method an estimate of the amount of data that can be hidden in an image so that the frequency count statistics are maintained, is calculated first. From this an image is chosen to hide a message of a given size safely. This method preserves the statistical properties of a cover medium by applying additional transforms to the redundant data. These transforms correct measurable deviations in the statistics caused by the embedding process without decreasing the embedding capacity of the stego images. The additional transforms are data format independent i.e. , this is a generic concept.

The Outguess embedding process can be divided into these steps:

1) Identification of redundant bits
2) Selection of bits for hiding information

The first step can be changed depending upon the data format which is being used. The second step can be the same for different data formats.

2.3.1 Identification of Redundant Bits

This involves finding the redundant bits that can be modified in the cover media without detectably degrading the cover medium. The redundant bits of an image or any cover media depend on the data format. Here the embedding happens when the cover medium is wriiten in that format. Minimizing modifications to the cover medium requires knowledge of the redundant bits before the actual stego medium is created.

For this purpose of identification of bits a pseudo-random number generator is used that that is initialized to same state for the identification and the final conversion step. Before the identified bits are sent to the selection step, some additional information is added to it. This information includes the locked bits which may not be modified for the embedding process and a heuristic that determines how detectable changes to a bit might be. The locked bit information is more useful when more than one message is being hidden in the cover medium.

2.3.2 Selection of Bits

An RC4 stream cipher is initialized with a secret key The keyed stream cipher is used to encrypt the hidden message and derive a PRNG for the selection of bits from it . The bits that can be replaced are selected with the help of a pseudo-random number generator.

If we need to hide 32 state bits which consist of a 16-bit seed and a 16-bit integer containing length of a hidden message. The selection starts at the beginning of the identified bits. The next bit is determined by computing a random offset within a fixed interval and adding that offset to the current bit position. To compute these random offsets a PRNG is again used. The data at the new position bit is replaced with the message data bit and this is repeated 32 times.

After the state data has been embedded, the PRNG is reseeded with 16-bit seed. The interval out of which the random numbers are drawn is calculated as follows

interval = 2X(remaining redundant bits) / (remaining length of the message)

To make the hidden message more evenly distributed over all the available bits in the cover medium this interval is adjusted for every 8 bits.

The seeding of the PRNG has some considerations. A PRNG selects its own subset of redundant bits. Different selections result in different number of bits that have to be changed. The number of these changed bits follows a binomial distribution.

$$p_k^{(n)} = {}^nC_k p^k (1-p)^{(n-k)} ,$$

where p is the probability that a selected bit in the redundant data has to be changed and $p_k^{(n)}$ is the probability that k of n bits will be changed.

If a seed that represents the changed bits at the lower end of the binomial distribution is selected, this reduces the no. of bits that have to be modified. And also more modifications to the image can be avoided by reseeding the PRNG for smaller hidden than for larger ones.

CHAPTER 3

STEGANALYSIS METHODS

3.1 Steganalysis Based on Statistical Moments of Wavelet Characteristic Functions (39-Dimensional Feature Vectors)

This steganalysis method which has been developed at NJIT uses the features of the original and the stego images calculated from statistical moments of wavelet characteristic functions:

This steganalysis method can be viewed as a matter of pattern recognition. It involves the following two steps:

1) Feature Extraction
2) Classifier

3.1.1 Feature Extraction

The features are selected such that they help in distinguishing between the stego and original images. That means these features should be rather different for the image without hidden message and for the corresponding stego-image. The larger the difference, the better the features are. And the features should be data format independent i.e., we should be able to use these features for any data format (JPEG,GIF,BMP...). These features should also be such that they can be used for different data hiding methods.

In this method, the statistical moments of characteristic functions (CF's) of wavelet subbands are used to form multi-dimensional (M-D) feature vector for steganalysis.

The histogram of a digital image or a wavelet subband is essentially the probability mass function (pmf), if the image grayscale values or the wavelet coefficient values are treated as a random variable. If each component of the histogram is multiplied by a correspondingly shifted unit impulse, we then have the probability density function (pdf). The histogram of the image and the characteristic function form a discrete Fourier transform pair.

The coefficients of different subbands at the same level are kind of independent to each other because of the de-correlation capability of wavelet transform. Therefore, the features generated from different wavelet subbands at the same level are kind of independent to each other as well. The *n*-th statistical moment of a characteristic function , M_n is given by

$$M_n = \left(\sum_{k=0}^{N/2} f_k^n \, | H(f_k) | \right) \Big/ \left(\sum_{k=0}^{N/2} | H(f_k) | \right)$$

where $|H(f_k)|$ is the magnitude of the characteristic function. The histogram of the image and the characteristic function form a discrete fourier transform pair.

These moments of the characteristic functions of wavelet subbands are used as features for steganalysis.

In the 39- D Feature Vector Calculation, a three-level Haar discrete wavelet transformation (DWT) is applied to a test image. The obtained 12 subbands are denoted by LL_i, HL_i, LH_i, HH_i, where i =1, 2, 3 . Denote the test image by LL_0 . We then have 13 subbands, i.e., LL_1 , HL_1 , LH_1 , HH_1 , LL_2 , HL_2 , LH_2 , HH_2 , LL_3 , HL_3 , LH_3 , HH_3 ,

and LL_0 . The 1st, 2nd and 3rd moments of the characteristic function of the 13 subbands are calculated. This makes up the 39- D feature vector. The moments are limited to 3rd order since it is observed that there is not much improvement in performance if we go above order 3.

3.1.2 Classifier

Pattern recognition is also known as classification, pattern classification or statistical classification. It is a field within the area of machine learning and can be defined as "the act of taking in raw data and taking an action based on the category of the data" [13]. As such, it is a collection of methods for supervised learning. Classifiers may either be fixed classifiers or learning classifiers, and learning classifiers may in turn be divided into supervised and unsupervised learning classifiers.

Formally, the problem can be stated as follows: given training data $\{(x_1 , y), \ldots,(x_n ,y)\}$ produce a classifier h: $X \rightarrow Y$ which maps an object $x \in X$ to its classification label $y \in Y$. For example, if the problem is to classify stego and cover images, then x_i is some representation of an image and y takes the values "Stego" or "non-Stego".

The different types of classifiers used are:

- Linear classifiers
- k-nearest neighbor
- Boosting
- Decision trees

- Neural networks
- Bayesian networks
- Support vector machines
- Hidden Markov models

Some examples of the Linear classifiers are:

1. Fisher's linear discriminant
2. Logistic regression
3. Naive Bayes classifier
4. Perceptron

3.1.2.1 Support vector machine. Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression.

3.1.2.2 Linear SVM. When used for classification, the SVM algorithm creates a hyperplane that separates the data into two classes with the maximum-margin. Given training examples labeled either "yes" or "no", a maximum-margin hyperplane splits the "yes" and "no" training examples, such that the distance from the closest examples (the margin) to the hyperplane is maximized.

The use of the maximum-margin hyperplane is motivated by Vapnik Chervonenki's theory, which provides a probabilistic test error bound that is minimized when the margin is maximized. However the utility of this theoretical analysis is sometimes questioned given the large slack associated with these bounds: the bounds often predict more than 100% error rates.

The parameters of the maximum-margin hyperplane are derived by solving a quadratic programming (QP) optimization problem. There exist several specialized algorithms for quickly solving the QP problem that arises from SVMs. The most common method for solving the QP problem is Platt's SMO algorithm.

3.1.2.3 Non-linear SVM. The original optimal hyperplane algorithm proposed by Vladimir Vapnik in 1963 was a linear classifier. However, in 1992, Bernhard Boser, Isabelle Guyon and Vapnik suggested a way to create non-linear classifiers by applying the kernel trick (originally proposed by Aizerman) to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a non-linear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in the transformed feature space. The transformation may be non-linear and the transformed space high dimensional; thus though the classifier is a hyperplane in the high-dimensional feature space it may be non-linear in the original input space.

If the kernel used is a radial basis function, the corresponding feature space is a Hilbert space of infinite dimension. Maximum margin classifiers are well regularized, so the infinite dimension does not spoil the results. Some common kernels include,

1. Polynomial (homogeneous):

$$k(x, x') = (x \cdot x')^d$$

2. Polynomial (inhomogeneous):

$$k(x, x') = (x \cdot x' + 1)^d$$

3. Radial Basis:

$$k(x,x') = \exp(-\frac{\|x-x'\|}{2\sigma^2})$$

4. Sigmoid:

$$k(x,x') = \tanh(kx \cdot x' + c)$$

for k>0,c>0

3.1.2.4 Regression. A version of a SVM for regression was proposed in 1997 by Vapnik, Steven Golowich, and Alex Smola. This method is called Support vector regression (SVR). The model produced by Support Vector Classification (as described above) only depends on a subset of the training data, because the cost function for building the model does not care about training points that lie beyond the margin. Analogously, the model produced by SVR only depends on a subset of the training data, because the cost function for building the model ignores any training data that is close (within a threshold ε) to the model prediction.

In my experimental work, the support vector machine is used for classification of the cover and the stego media. The SVM kernel used is the polynomial SVM kernel.

3.2 Steganalysis Based on Statistical Moments of Wavelet Characteristic Functions (78-Dimensional Feature Vectors)

This steganalysis method which is a modification of the 39D method discussed above also uses the features of the original and the stego images calculated from statistical moments of wavelet characteristic functions. Additionally this 78D method uses the

moments of characteristic functions of the prediction error images. This method can be subdivided into 2 steps:

1) Feature Extraction
2) Classifier

The classification of images is done in a similar fashion as in 39D method as discussed in section 3.1. The feature extraction in this 78D method has some changes when compared to that in the 39D method. The features of an image are calculated using

1) moments of characteristic functions of the wavelet subbands of the test image
2) moments of characteristic functions of the coefficients of the image
3) moments of characteristic functions of the wavelet subbands of prediction error image
4) moments of characteristic functions of the prediction error image

The moments of characteristic functions of the wavelet subbands of the test image are calculated as shown in section 3.1.1. The error statistics are calculated using a linear predictor. A linear predictor predicts the value of a particular coefficient using the surrounding coefficients and the coefficients from the surrounding subbands so that the error between the predicted value and the original value is minimized. The features formed from the error statistics form a 39D feature vector for each coefficient. Combining these with the features extracted form the coefficient statistics, which also form a 39D vector, we get a total of 78D feature vectors for each coefficient.

3.3 Steganalysis Based on Statistical Moments of Wavelet Histograms

Although the human eye cannot detect the presence of embedded messages in some images, they cannot be imperceptible to statistical attacks. It is seen that steganalysis attacks that examine first order statistical properties of images are likely to foiled because some simple counter measures can be made to the Steganography methods. This method is based on building higher-order statistical models for natural images and looks for deviations from these models. For this it uses the wavelet decomposition. According to their paper [6], a broad range of natural images tends to produce similar higher order coefficient statistics. Additionally, alterations such as steganography tend to change those coefficient statistics.

Image Statistics:
The decomposition employed here is based on separable quadrature mirror filters (QMFs). This decomposition splits the frequency space into multiple scales and orientations. The scales are created by filtering the lowpass subbands.

This method uses the mean, variance, skewness, and kurtosis of the coefficients generated at the LH, HL, and HH subbands for all scales. These statistics characterize the one set of coefficient distributions. After decomposition if there are n scales, then the number of individual statistics or the features calculated from the actual coefficients is 12(n - 1). And then a second set of statistics are calculated from errors in an
optimal linear predictor of the coefficient magnitudes. This linear predictor uses linear regression. The value of a particular coefficient is predicted using the surrounding

coefficients and the coefficients from the surrounding subbands. This is done such that the error between the predicted value and the observed value was minimized.

Let the vertical, horizontal, and diagonal subbands at scale i = 1, ..., n be denoted as $V_i(x ,y)$, $H_i (x ,y)$, and $D_i (x ,y)$, respectively. If we consider a vertical band, $V_i(x, y)$, linear predictor for the magnitude of these coefficients in a subset of all possible neighbors is given by

$$\begin{aligned} V_i(x, y) = {} & w_1 V_i(x-1 ,y) + w_2 V_i(x+1 ,y) \\ & + w_3 V_i(x ,y-1) + w_4 V_i(x ,y+1) \\ & + w_5 V_{i+1} (x/2 ,y/2) + w_6 D_i (x ,y) \\ & + w_7 D_{i+1} (x/2 ,y/2) \end{aligned}$$

This linear relationship is expressed in matrix form as:

$$V = Qw$$

where V is the matrix containing the coefficient magnitudes of $V_i(x ,y)$, Q is a matrix containing magnitudes of neighboring coefficients and w is a vector of the weighting factors. The error function is given by

$$E(w) = [V-Qw]^2$$

From this the log error is calculated as

$$E = \log_2 V - \log_2 |Qw|$$

The mean, variance, skewness, and kurtosis are calculated from this value. This is repeated for the other two subbands i.e., horizontal and diagonal. The error statistics give us 12(n-1) features. Totally 24(n-1) features are created which are used for classification of the images.

3.4 Comparison of Farid's 72D method and the NJIT's 39D and 78D methods

As it has been discussed in the previous sections, both the steganalysis approaches use the higher order statistical models of the images and try to observe deviations from these models to detect the presence of any hidden data in images.

The Farid's 72 D method uses the moments of wavelet subbands. And the NJIT's 39D and 78D methods go one step further and they use the moments of characteristic functions of the wavelet subbands. In the Farid's 72D method, only the high-frequency wavelet subbands i.e., LH, HL, HH subbands are used in the calculation of mean, variance, skewness and kurtosis. Whereas the NJIT's 39D and the 78D methods use all the wavelet subbands for the calculation of the moments of wavelet characteristic functions. The most important aspect of the NJIT's 39D and 78D methods is that the moments of characteristic function for the given test image are also considered as a feature. In Farid's method, the first four statistical moments of wavelet coefficients and their prediction errors of nine high frequency subbands are used to form a 72-dimensional (72-D) feature vector for steganalysis.

It has been theoretically shown in [1] that the defined n-th statistical moment of a wavelet characteristic function is related to the n-th derivative of the corresponding wavelet histogram, and hence is sensitive to data embedding. This theoretical analysis is briefly described here.

The inverse transform of characteristic function produces the pdf (here, the histogram) as follows.

$$h(x) = \int_{-\infty}^{\infty} H(f) e^{-j2\pi fx} df$$

We can derive the n-th derivative of the histogram evaluated at the origin, x=0, as follows.

$$\left.\frac{d^n}{dx^n} h(x)\right|_{x=0} = \left.\frac{d^n}{dx^n} \int_{-\infty}^{\infty} H(f) e^{-j2\pi fx} df\right|_{x=0}$$

$$= \begin{cases} (-1)^{n/2} 2(2\pi) \int_0^{\infty} f^n \,\mathrm{Re}(H(f)) df, n \in even \\ (-1)^{(n-1)/2} 2(2\pi)^n \int_0^{\infty} f^n \,\mathrm{Re}(H(f)) df, n \in odd \end{cases}$$

From the above equation, we get

$$\left| \left(\left.\frac{d^n}{dx^n} h(x)\right|_{x=0} \right) \right| \le 2(2\pi)^n \int_0^{\infty} f^n |H(f)| df$$

The right hand side of the above inequality is the moments of CF's, M_n , multiplied by a scalar, which is dependent to the energy of an image or a wavelet subband, from which the moment is generated. This indicates that the features we defined actually are the upper bound (up to a scalar) of the magnitude of the n-th derivative of the histogram

evaluated at the origin of the histogram, i.e., x=0. Furthermore, this observation can easily be extended to the case when $x \neq 0$. Therefore, the defined n-th moment of a CF is closely related to the n-th derivative of the corresponding histogram. And also the n-th moments defined will decrease after data embedding. This decrease lowers the upper bound of the magnitude of the n-th derivative of the histogram. This implies that the moments of CF's defined in NJIT's 39D and 78D methods can sensitively catch the changes caused by data hiding. The effectiveness of these methods has been demonstrated by extensive experimental investigation on a set of 1096 images CorelDraw image database in the paper [1].

From this discussion, we can observe that the feature vectors calculated using NJIT's 39D and 78D methods, based on statistical moments of wavelet characteristic functions, make up a better model for steganalysis than the 72 D method. By a better model means changes between the cover and stego images can be picked up more effectively. Hence, the steganalysis methods based on statistical moments of wavelet characteristic functions show superior performance when compared to the Farid's 72 D method.

CHAPTER 4

EXPERIMENTAL RESULTS

The purpose of this thesis is to study some of the steganographic methods and also steganalysis methods. I have done my experiments with an image set of 3474 images. All these images are of size 768 X 512 pixels and they have a quality factor ranging from 70 to 90. In this work I've implemented the different steganographic methods. The steganographic methods MB1, MB2 and F5 have been implemented. The stego images have been created for different embedding rates. I've concentrated on the implementation of Model-based methods.

The Steganalysis has been done on the stego images created from different steganographic methods. The Steganalysis methods used are Wavelet-based 39 Dimensions method and 78 Dimensions method developed at NJIT [1]. So, the security performance of the different data hiding methods has been studied.

The results of the study are tabulated in Table 1 to Table 16. In the tables 1 to 12, 'cover' stands for correct detection rate in testing cover images, 'stego' stands for correct detection rate in testing stego images. 'Average' (in the right-most column) refers to the average of the detection rates for stego and cover images for each iteration number (mentioned in the 1st column). 'Average' (in the bottom-most row) refers to the arithmetic mean of the detection rates for all the iterations.

Table 4.1 Detection Rates for F5 (with 0.1 bpc) and 39D method

F5(0.1bpc) & 39D method			
Iteration no.	Cover	Stego	Average
1	59.408	62.195	60.801
2	68.293	55.226	61.76
3	61.672	60.976	61.324
4	64.111	61.498	62.805
5	62.369	60.801	61.585
6	64.286	57.666	60.976
7	64.111	59.233	61.672
8	65.505	57.317	61.411
9	66.028	56.62	61.324
10	63.24	58.014	60.627
11	62.195	59.93	61.063
12	64.111	56.969	60.54
13	66.725	56.969	61.847
14	61.498	59.408	60.453
15	61.324	61.324	61.324
16	65.331	58.537	61.934
17	66.202	57.317	61.76
18	60.801	61.324	61.063
19	59.756	61.847	60.801
20	63.763	60.801	62.282
Average	**63.53645**	**59.1986**	**61.3676**

Table 4.2 Detection Rates for F5 (with 0.1 bpc) and 78D method

F5(0.1bpc) & 78D method			
Iteration no.	Cover	Stego	Average
1	65.854	61.847	63.85
2	64.634	63.589	64.111
3	65.157	63.066	64.111
4	66.725	63.763	65.244
5	66.376	62.369	64.373
6	67.073	65.157	66.115
7	67.596	60.976	64.286
8	64.286	66.551	65.418
9	69.338	61.324	65.331
10	61.672	62.544	62.108
11	66.202	64.46	65.331
12	67.422	65.331	66.376
13	64.286	64.983	64.634
14	62.892	65.505	64.199
15	66.551	66.376	66.463
16	62.369	68.467	65.418
17	64.634	66.202	65.418
18	66.202	63.589	64.895
19	68.815	61.847	65.331
20	69.686	58.711	64.199
Average	**65.8885**	**63.83285**	**64.86055**

Table 4.3 Detection Rates for F5 (with 0.2 bpc) and 39D method

F5(0.2bpc) & 39D method			
Iteration no.	Cover	Stego	Average
1	63.589	60.801	62.195
2	66.028	57.143	61.585
3	59.408	62.718	61.063
4	68.118	55.401	61.76
5	61.498	61.15	61.324
6	63.937	61.672	62.805
7	62.544	60.976	61.76
8	64.46	57.84	61.15
9	63.937	59.233	61.585
10	65.679	57.491	61.585
11	65.679	57.491	61.585
12	63.24	57.491	60.366
13	62.195	60.279	61.237
14	63.937	56.794	60.366
15	66.202	57.666	61.934
16	61.324	59.582	60.453
17	61.498	61.847	61.672
18	64.983	59.059	62.021
19	66.551	57.84	62.195
20	60.801	61.324	61.063
Average	**63.7804**	**59.1899**	**61.4852**

Table 4.4 Detection Rates for F5 (with 0.2 bpc) and 78D method

F5(0.2bpc) & 78D method			
Iteration no.	Cover	Stego	Average
1	62.892	64.286	63.589
2	69.338	58.885	64.111
3	59.756	68.293	64.024
4	66.376	62.892	64.634
5	63.24	66.376	64.808
6	63.763	66.202	64.983
7	60.976	66.028	63.502
8	67.596	62.544	65.07
9	65.505	63.415	64.46
10	65.157	63.937	64.547
11	67.422	64.111	65.767
12	66.551	63.415	64.983
13	62.544	66.376	64.46
14	65.854	63.589	64.721
15	66.551	64.46	65.505
16	63.24	67.073	65.157
17	67.77	62.892	65.331
18	64.46	67.073	65.767
19	64.46	63.24	63.85
20	62.369	64.46	63.415
Average	**64.791**	**64.47735**	**64.6342**

Table 4.5 Detection Rates for MB1 (with 0.1 bpc) and 39D method

MB1(0.1bpc) & 39D method			
Iteration no.	Cover	Stego	Average
1	56.098	47.909	52.003
2	59.756	46.69	53.223
3	58.188	49.826	54.007
4	59.233	49.129	54.181
5	53.484	50.348	51.916
6	60.801	45.296	53.049
7	61.324	43.902	52.613
8	61.672	44.077	52.875
9	59.93	45.296	52.613
10	58.362	45.122	51.742
11	58.014	46.167	52.091
12	61.324	45.122	53.223
13	56.446	48.955	52.7
14	59.059	44.077	51.568
15	62.195	44.077	53.136
16	54.878	50	52.439
17	62.195	42.509	52.352
18	57.143	46.864	52.003
19	55.575	49.303	52.439
20	57.84	48.432	53.136
Average	**58.67585**	**46.65505**	**52.66545**

Table 4.6 Detection Rates for MB1 (with 0.1 bpc) and 78D method

MB1(0.1bpc) & 78D method			
Iteration no.	Cover	Stego	Average
1	51.045	60.801	55.923
2	52.962	58.537	55.749
3	54.704	59.582	57.143
4	51.568	58.537	55.052
5	51.568	62.544	57.056
6	54.007	58.362	56.185
7	50.174	60.627	55.401
8	51.742	61.324	56.533
9	50.697	59.756	55.226
10	55.226	54.181	54.704
11	53.484	59.756	56.62
12	52.091	59.408	55.749
13	52.091	58.885	55.488
14	54.53	56.098	55.314
15	50.697	58.885	54.791
16	49.129	61.15	55.139
17	52.787	54.878	53.833
18	52.091	59.756	55.923
19	52.091	58.014	55.052
20	49.477	60.453	54.965
Average	**52.10805**	**59.0767**	**55.5923**

Table 4.7 Detection Rates for MB1 (with 0.2 bpc) and 39D method

MB1(0.2bpc) & 39D method			
Iteration no.	Cover	Stego	Average
1	61.498	52.787	57.143
2	60.801	55.401	58.101
3	57.491	56.446	56.969
4	57.491	55.575	56.533
5	61.498	54.181	57.84
6	61.324	55.575	58.449
7	61.498	51.22	56.359
8	58.362	55.575	56.969
9	61.672	51.045	56.359
10	61.498	54.181	57.84
11	62.021	53.136	57.578
12	57.84	55.749	56.794
13	63.763	49.303	56.533
14	60.453	53.659	57.056
15	54.878	56.794	55.836
16	60.453	51.568	56.01
17	58.711	53.833	56.272
18	60.105	52.787	56.446
19	57.666	58.362	58.014
20	60.453	53.833	57.143
Average	**59.9738**	**54.0505**	**57.0122**

Table 4.8 Detection Rates for MB1 (with 0.2 bpc) and 78D method

MB1(0.2bpc) & 78D method			
Iteration no.	Cover	Stego	Average
1	56.794	65.505	61.15
2	57.84	65.854	61.847
3	58.885	64.634	61.76
4	55.923	67.422	61.672
5	58.885	63.589	61.237
6	59.582	62.892	61.237
7	57.491	66.551	62.021
8	58.537	63.24	60.889
9	56.62	66.028	61.324
10	61.672	59.233	60.453
11	59.756	63.763	61.76
12	58.537	64.111	61.324
13	58.885	64.808	61.847
14	61.15	60.976	61.063
15	58.014	63.763	60.889
16	55.923	67.596	61.76
17	58.014	61.15	59.582
18	59.582	62.718	61.15
19	57.666	62.718	60.192
20	56.969	66.899	61.934
Average	**58.33625**	**64.1725**	**61.25455**

Table 4.9 Detection Rates for MB2 (with 0.1 bpc) and 39D method

MB2(0.1bpc) & 39D method			
Iteration no.	Cover	Stego	Average
1	50.523	55.923	53.223
2	57.317	52.439	54.878
3	54.704	55.226	54.965
4	53.833	55.226	54.53
5	54.704	54.181	54.443
6	55.575	51.916	53.746
7	55.575	52.265	53.92
8	55.401	52.613	54.007
9	52.962	53.31	53.136
10	52.962	52.962	52.962
11	53.659	54.181	53.92
12	55.749	53.484	54.617
13	52.613	53.484	53.049
14	58.537	48.432	53.484
15	56.098	52.091	54.094
16	51.394	54.878	53.136
17	59.233	48.084	53.659
18	56.62	52.091	54.355
19	51.742	57.491	54.617
20	53.484	55.923	54.704
Average	**54.63425**	**53.31**	**53.97225**

Table 4.10 Detection Rates for MB2 (with 0.1 bpc) and 78D method

MB2(0.1bpc) & 78D method			
Iteration no.	Cover	Stego	Average
1	53.833	56.62	55.226
2	53.659	56.794	55.226
3	56.62	60.453	58.537
4	49.129	60.976	55.052
5	53.833	60.453	57.143
6	53.659	60.279	56.969
7	52.439	59.756	56.098
8	55.575	55.749	55.662
9	52.613	58.885	55.749
10	57.491	53.31	55.401
11	53.659	61.324	57.491
12	54.704	57.491	56.098
13	55.749	60.627	58.188
14	57.84	56.098	56.969
15	55.401	59.582	57.491
16	53.31	59.408	56.359
17	55.052	55.401	55.226
18	55.226	58.362	56.794
19	52.439	57.84	55.139
20	50.697	63.24	56.969
Average	**54.1464**	**58.6324**	**56.38935**

Table 4.11 Detection Rates for MB2 (with 0.2 bpc) and 39D method

MB2(0.2bpc) & 39D method			
Iteration no.	Cover	Stego	Average
1	59.408	55.226	57.317
2	62.195	56.098	59.146
3	62.892	54.878	58.885
4	58.362	58.188	58.275
5	60.627	54.007	57.317
6	62.544	55.749	59.146
7	64.46	54.53	59.495
8	62.544	53.484	58.014
9	60.105	56.446	58.275
10	64.46	48.955	56.707
11	63.066	54.355	58.711
12	64.808	52.265	58.537
13	58.537	57.84	58.188
14	65.505	52.962	59.233
15	63.24	53.659	58.449
16	56.272	61.498	58.885
17	59.408	52.439	55.923
18	60.976	53.833	57.404
19	60.453	54.704	57.578
20	59.233	59.93	59.582
Average	**61.45475**	**55.0523**	**58.25335**

Table 4.12 Detection Rates for MB2 (with 0.2 bpc) and 78D method

MB2(0.2bpc) & 78D method			
Iteration no.	Cover	Stego	Average
1	50.523	55.923	53.223
2	57.317	52.439	54.878
3	54.704	55.226	54.965
4	53.833	55.226	54.53
5	54.704	54.181	54.443
6	55.575	51.916	53.746
7	55.575	52.265	53.92
8	55.401	52.613	54.007
9	52.962	53.31	53.136
10	52.962	52.962	52.962
11	53.659	54.181	53.92
12	55.749	53.484	54.617
13	52.613	53.484	53.049
14	58.537	48.432	53.484
15	56.098	52.091	54.094
16	51.394	54.878	53.136
17	59.233	48.084	53.659
18	56.62	52.091	54.355
19	51.742	57.491	54.617
20	53.484	55.923	54.704
Average	**54.63425**	**53.31**	**53.97225**

Table 4.13 Comparison of Detection Rates for F5 and Model-based methods for embedding rate of 0.1bpc and using NJIT's 39 D steganalysis method:

Method	Cover	Stego	Average
F5	63.53645	59.1986	61.3676
MB1	58.67585	46.65505	52.66545
MB2	54.63425	53.31	53.97225

Table 4.14 Comparison of Detection Rates for F5 and Model-based methods for embedding rate of 0.2bpc and using NJIT's 39 D steganalysis method:

Method	Cover	Stego	Average
F5	63.7804	59.1899	61.4852
MB1	59.9738	54.0505	57.0122
MB2	61.45475	55.0523	58.25335

Table 4.15 Comparison of Detection Rates for F5 and Model-based methods for embedding rate of 0.1bpc and using NJIT's 78 D steganalysis method:

Method	Cover	Stego	Average
F5	65.8885	63.83285	64.86055
MB1	52.10805	59.0767	55.5923
MB2	54.1464	58.6324	56.38935

Table 4.16 Comparison of Detection Rates for F5 and Model-based methods for embedding rate of 0.2bpc and using NJIT's 78 D steganalysis method:

Method	Cover	Stego	Average
F5	64.791	64.47735	64.6342
MB1	58.33625	64.1725	61.25455
MB2	54.63425	53.31	53.97225

It is seen that among the steganographic methods used , Model-based Method 2 is the toughest one. It is the most difficult one to break. The detection rates for F5 are the highest followed by those of MB1 or MB2.

CHAPTER 5

CONCLUSIONS AND FUTURE RESEARCH

Conclusions

1) The Steganography methods MB1, MB2 and F5 have been tested, Model Based methods have been found to be the toughest. The detection rates for MB1 for embedding rates of 0.1 bits per non-zero coefficient is 53% and the value for F5 is 61%. The model based methods are the most secure Steganography methods.
2) It is observed as the embedding rates increase for the stego images, the detection also increase.

Future Research

The model based approach can be applied to other file formats i.e., BMP, GIF, PNG, and also to audio and video files. More robust Steganalysis methods are needed to attack Model based methods.

On the Steganalysis side I've concentrated on methods which use supervised training wherein the user (steganalyst) has a copy of the original images. SVM was used as the classifier for classifying the cover and the stego images.

REFERENCES

1. Guorong Xuan, Yun Q. Shi, "Steganalysis Based on Multiple Features Formed by Statistical Moments of Wavelet Characteristic Functions", Information Hiding Workshop (IHW05), June 2005.

2. Simmons, G., "The Prisoners Problem and the Subliminal Channel", Proc. of CRYPTO, 1983.

3. Katzenbeisser, Stefan and Fabien A. P. Petitcolas, "Information Hiding Techniques for Steganography and Digital Watermarking", Boston, Artech House, 2000.

4. Westfeld, Andreas and Andreas Pfitzmann. "Attacks on Steganographic Systems Breaking the Steganographic Utilities EzStego, Jsteg, Steganos, and S-Tools - and Some Lessons Learned," Lecture Notes in Computer Science, 1768:61– 75 (2000).

5. Johnson, Neil F. and others. "Information Hiding: Steganography and Watermarking Attacks and Countermeasures". Boston: Kluwer Academic Publishers, 2001.

6. Farid, Hany, "Detecting Steganographic Messages in Digital Images", Technical Report TR2001-412, Hanover, NH: Dartmouth College, 2001.

7. Neil F. Johnson and Sushil Jajodia, "Steganalysis of Images Created Using Current Steganography Software, Center for Secure Information Systems", George Mason University.

8. Herodotus, "Herodotus: The Histories", Penguin Books, London, 1996, edited by John M. Marincola, translated by Aubrey De Selincourt.

9. R. Chandramouli and K.P. Subbalakshmi, "Current Trends in Steganalysis: a Critical Survey".

10. Westfeld, Andreas, "F5—A Steganographic Algorithm: High Capacity Despite Better Steganalysis", 4th International Workshop of Information Hiding, IH'01, Pittsburgh, USA, April 2001.

11. Ron Crandall, "Some Notes on Steganography". Posted on Steganography Mailing List, 1998. http://os.inf.tu-dresden.de/~westfeld/crandall.pdf

12. Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification", 2nd edition, Wiley, New York, 2001.

13. Mehdi Kharrazi, Husrev T. Sencar, Nasir Memon, "Benchmarking Steganographic and Steganalysis techniques", EI SPIE San Jose, CA, January 16-20, 2005.

14. Phil Sallee, "Model Based Methods for Steganography and Steganalysis", International Journal of Image and Graphics, Vol. 5 , No.1 (2005) 167-189.

15. Bret Dunbar, "A detailed look at Steganographic Techniques and their use in an Open-Systems Environment".