

St. Clements University



Mat. No. AC0180

# High Security and Efficient Information Hiding System

A THESIS

SUBMITTED TO St. CLEMETS UNIVERCITY IN PARTIAL  
FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY OF SCIENCE IN INFORMATION  
TECHNOLOGY

By

*Naeem Yaser Salman Al- khafaji*

Supervised By

*Dr. Emad Abbas Kufi*

Baghdad 2014

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

الرَّحْمَنُ \* عَلَّمَ الْقُرْآنَ \*

خَلَقَ الْإِنْسَانَ \* عَلَّمَهُ الْبَيَانَ \*

الْحَمْدُ لِلَّهِ  
الْعَظِيمِ

الرحمن (1-4)

## Supervisor's Certification

I certify that this thesis entitled “ *High Security and Efficient Information Hiding System* ” by *Naeem Yaser Salman* was prepared under my supervision of the St. Clements University in a partial fulfillment of the requirements for the degree of Doctor of Philosophy of Science in Information Technology.


Signature:

Name: **Asst. Prof. Dr. Emad Abbas Kufi**

Date:     /     / 2014



# Dedication



**TO MY DEAR PARENTS,  
WIFE, CHILDREN  
AND MY BROTHERS.**

*Naeem*



# Acknowledgements

It's a great pleasure for me when presenting this thesis to praise God Almighty for all His blessings and His guidance, which greatly helped me finish this research.

I am wholly indebted to my supervisor *Dr. Emad Abbas Kufi* for his supervision, continuous guidance, encouragement, helpful suggestions and patience in answering my questions during the preparation of this project.

My sincere thanks to the dean of the St. Clements University *Dr. Nazar Al-Rubaiee* whose advice and scientific knowledge was quite instrumental.  
are also due.

I'm exceptionally indebted to all the members of the St. Clements University.

Finally, I would like render my sincere acknowledgments to my family for their inspiration, encouragement and support.

*Naeem*

# **ABSTRACT**

As steganography developers and users become more savvy, detection of hidden information will become more challenging. If one cannot detect the possibility of hidden information, then the game of extraction may be lost. Simply embedding information into the LSB of an image provides no protection if the scheme produces artifacts. For this reason, the research is chosen to warn the users who are using hiding tools which depend on LSB techniques.

In this research a new method was suggested and classical ones are developed to detect the hidden message. The classical methods were developed by adding new techniques to increase their performance.

The aim of this thesis is to analyze the hidden information and not only detect it. In general, steganalysis means detection, extraction or distortion of the secret message. The proposed system is designed to deal with Steganalysis tools and to process in three stages: Diagnosis, Breaking and Extraction.

In the diagnosis process, we suggest to use statistical analysis and visual pixel test as a tool for detection. These tests are: Mean Square Error (MSE) for stego-cover images, and Laplace Operator, Chi-Square and Visual Test for stego only attack.

The breaking process includes breaking three stego tools, sequential, jumping and Linear Feedback Shift Register (LFSR) algorithm if a stego and cover images are available. The breaking process is developed to improve its performance to break stego image only for the three mentioned tools.

The last process, is the extracting process includes view the hidden text, if the hidden text is a simple substitution ciphertext, the system is developed to break it, then view plaintext.

# **Contents**

## ***Chapter one Overview***

1-1 Introduction-----	1
1-2 Terminology -----	2
1-3 Image Representation -----	3
1-3-1 24-Bit Image-----	3
1-3-2 8-Bit Images -----	4
1-3-3 1-Bit Images -----	4
1-4 Digital Image File Format -----	5
1-4-1 Bitmap BMP -----	5
1-4-2 Graphics Interchange Format GIF -----	5
1-4-3 Joint Photographic Experts Group JPEG -----	6
1-5 The BMP File Format -----	6
1-5-1 File Header -----	7
1-5-2 Image Header-----	8
1-5-3 Image Data Area -----	10
1-6 Encryption -----	10
1-7 Literature Survey -----	10
1-8 Aim of Thesis -----	12

## ***Chapter Two Information Hiding***

2-1 Introduction -----	13
2-2 Information Hiding Preview -----	13
2.3 Methods of Hiding Information-----	17
2.3.1 Injection -----	17
2.3.2 Substitution -----	18
2.3.3 Generation of new file -----	18
2-4 Basic Model of Steganography System-----	19
2-5 Steganography Through History-----	20
2-6 Steganography Under Various Media -----	21
2-6-1 Hiding in Text-----	22

2-6-2 Hiding in Disk Space-----	23
2-6-3 Hiding in Network Packets -----	24
2-6-4 Hiding in Audio and Images -----	24
2-7 Hiding in Images -----	24
2-7-1 Some Guidelines to Image Steganography -----	25
2-7-2 Image Encoding Techniques -----	26
2-7-2-1 Least Significant Bit Insertion -----	26
2-7-2-2 Masking and Filtering -----	28
2-7-2-3 Algorithms and Transformations -----	29
2-8 Some Applications of Information Hiding -----	29

### ***Chapter Three Steganalysis***

3-1 Introduction-----	31
3-2 Steganalysis -----	31
3-3 Classification of Steganography Attacks -----	32
3-3-1 Passive Attack -----	33
3-3-2 Active Attack-----	35
3-3-3 Malicious Attack-----	35
3-4 Types of Attacks -----	35
3-5 Steganalytic Methods -----	36
3-5-1 Visual Analysis -----	36
3-5-2 Statistical Analysis -----	37
3-5-2-1 Known Cover Attack Using Mean Squared Error-----	38
3-5-2-2 Stego-Only Attack Using Chi-Square Test-----	38
3-6 Steps of Steganalysis of Images -----	43
3-6-1 Detecting Hidden Information -----	46
3-6-2 Extracting -----	47
3-6-3 Disabling and Modification Steganography -----	47
3-7 Differences Between Cryptanalysis and Steganalysis -----	49

### ***Chapter Four Efficient of Hiding System***

4-1 Introduction-----	50
-----------------------	----



4-2 Attacked Steganography Systems -----	50
4-2-1 Sequential Steganography System -----	53
4-2-2 Jumping Steganography System -----	54
4-2-3 LFSR Steganography System -----	55
4-3 Steganalysis System Design -----	57
4-3-1 Hidden Message Diagnosis Stage (HMDS) -----	58
4-3-1-1 Diagnosis by Cover and Stego Images -----	59
4-3-1-2 Diagnosis by Stego Image Only -----	60
4-3-1-2-1 Visual Attack method -----	60
4-3-1-2-2 Laplace Operator method -----	61
4-3-1-2-3 Chi-Square Method -----	62
4-3-2 Breaking Stage (BS) -----	63
4-3-2-1 Breaking Using Cover and Stego Images -----	64
4-3-2-1-1 Breaking the SSA Using Cover Image -----	65
4-3-2-1-2 Breaking the JSA Using Cover Image -----	65
4-3-2-1-3 Breaking the LFSRSA Using Cover Image -----	66
4-3-2-2 Breaking Using Stego Image Only -----	67
4-3-2-2-1 Breaking the SSA Using Stego Image Only -----	68
4-3-2-2-2 Breaking the JSA Using Stego Image Only -----	69
4-3-2-2-3 Breaking the LFSRSA Using Stego Image Only -----	69
4-3-3 Extracting Stage (ES) -----	70
4-3-3-1 Extracting Plain or Cipher Message -----	70
4-3-3-2 Breaking the Encipher Text Message -----	71
4-4 System Requirements -----	73
4-5 System Implementation -----	73
4-5-1 Steganography System Implementation -----	73
4-5-1-1 Hiding RadioGroup -----	75
4-5-1-2 Extracting RadioGroup -----	75
4-5-2 Steganalysis System Implementation -----	75
4-5-2-1 Images Files Submenu -----	77
4-5-2-2 Diagnosis Submenu -----	77
4-5-2-3 Breaking Submenu -----	77

4-5-2-4 Extracting Submenu -----	77
4-6 Experimental Examples -----	77
4-7 Steganalysis System Efficiency Test -----	78

## ***Chapter Five Conclusions and Future Works***

5-1 Introduction-----	80
5-2 Conclusions-----	80
5-3 Suggestions for Future work -----	81
References -----	82

## List of Abbreviation

Symbol	Description
$\nabla$	Laplace Operator
$\Gamma$	Euler Gamma function
$\pi(i)$	Jump by i
BK	Basic Key
BMP	Bitmap
BS	Breaking Stage
DCT	Discrete Cosine Transform
ES	Extracting Stage
FAT	File Allocation Table
GIF	Graphics Interchange Format
HMDS	Hidden Message Diagnosis Stage
HVS	Human Visual System
IC	Index of Coincidence
JPEG	Joint Photographic Experts Group
Keyjmp	Jump as a Key
LFSR	Linear Feedback Shift Register
LSB	Least Significant Bit
MAXIC	Maximum IC
MSE	Mean Squared Error
p	Probability
PoV	Pair of Values
RGB	Red-Green-Blue
$\chi^2$	Chi-Square

## **List of figure**

<b>Figure No.</b>	<b>Figure Description</b>	<b>Page No.</b>
(1.1)	Color image representation .	4
(1.2)	Bitmap File Structure	6
(2.1)	Classification of information hiding techniques	14
(2.2)	General steganography system (stego-system)	19
(3.1)	Histogram of Laplace filtered	34
(3.2)	The histogram of observed values	41
(3.3)	The histogram of expected values	42
(3.4)	General steganography and steganalysis process	44
(3.5)	Basic modules in steganalysis	45
(4.1)	LFSR Steganography system	55
(4.2)	The structure of the proposed steganalysis system	58
(4.3)	Block diagram of steganography system	74
(4.4)	Block diagram of steganolysis system	76

## **List of Tables**

<b>Table Number</b>	<b>Table Description</b>	<b>Page No.</b>
1-1	BITMAPFILEHEADER Structure	7
1-2	BITMAPINFOHEADER Structure	9
1-3	BITMAPCOREHEADER Structure	9
3-1	The results of affected blueberries and fungus	41
3-2	The expected counts $E_{ij}$	42
3-3	Differences between cryptanalysis and steganalysis	49
4.1	Baudot encoding system	51
4.2	MSE results	59
4.3	Letter frequency distributions in English language	72
4.4	MSE test	78
4.5	Visual, Laplace and chi-square tests	79

# *Chapter One*

## **Overview**

# Chapter one

## Overview

### **1-1 Introduction**

Every few years, computer security has to re-invent itself. New technologies and new applications bring new threats, and force us to invent new protection mechanisms. Cryptography became important when businesses started to build networked computer systems; virus epidemics started once large numbers of PC users were swapping programs; and when the Internet took off, the firewall industry was one of the first to benefit [21].

The growth of the Internet has made government intelligence and police agencies nervous. They say that widely available encryption software could make wiretapping more difficult; their common reaction is to try to restrict the strength of encryption algorithms or require that spare copies of the keys are available somewhere for them to seize. Communications can also be hidden using the kind of techniques developed for copyright marking, and these can help criminals evade any laws against using “unapproved” cryptography.

As well as being important for copyright protection and to any long-term resolution of the crypto, information hiding is also important for privacy.

The research in information hiding has grown explosively. The progress made in the last five years is comparable to that in cryptology during 1945-1990. A large number of systems have been proposed; many

of them have been broken; we now have a fair idea of what works, what doesn't, and where the interesting research directions are [21].

## **1-2 Terminology [19, 31, 17]**

The word *Steganography* comes from the Greek *Steganos* (covered or secret) and *Graphy* (writing or drawing) and means literally covered writing. **Cover** is an input to the stego-system, in which the embedded will be hidden. The possible cover carriers are innocent looking carriers (images, audio, video, text, or some other digitally representative code) which will hold the hidden information. **Embedded** is something to be hidden in the cover. A **Message** is the information hidden and may be plaintext, ciphertext, images, or anything that can be embedded into a bit stream. **Embedding** is the process of hiding the embedded message. **Stego** is the output from the stego-system is something that has the embedded message hidden in it. Together the cover carrier and the embedded message create a stego-carrier. Hiding information may require a **Stegokey** which is additional secret information, such as a password, required for embedding the information. For example, when a secret message is hidden within a cover image, the resulting product is a stego-image.

**Extracting** is getting the embedded message out of the stego message again. New terminology with respect to attacks and breaking steganography schemes is similar to cryptographic terminology; however, there are some significant differences. Just as a **Cryptanalyst** applies **Cryptanalysis** in an attempt to decode or crack encrypted messages, the **Steganalyst** is one who applies **Steganalysis** in an attempt to detect the existence of hidden information. With cryptography, comparison is made between portions of the plaintext (possibly none) and portions of the ciphertext. In steganography, comparisons may be made between the cover-media, the stego-media, and possible portions of the message. The end result in cryptography is the ciphertext, while the end result in



steganography is the stego-media. The message in steganography may or may not be encrypted. If it is encrypted, then if the message is extracted, the cryptanalysis technique may be applied.

### **1-3 Image Representation**

A digital image is the most common type of carrier used for steganography. A digital image is produced using a camera, scanner or other device. The digital representation is an approximation of the original image. The system used for producing the image focuses a two-dimensional pattern of varying light intensity and color onto a sensor. The pattern has a co-ordinate system and the origin is the upper left-hand corner of the image. The pattern can be described by a function  $f(x, y)$ . An image can be described as an array of numbers that represent light intensities at various points. These light intensities or instances of color are called pixels. Sampling is the process of measuring the value of the image function  $f(x, y)$  at discrete intervals in space. Each sample is the small square area of the image known as the pixel. The raster data of an image is that part of the image that can be seen i.e. the pixels. The size of an image can be given in pixels, for example an image which is 640x480 pixels contains 307, 200 Pixels are indexed by x and y co-ordinates with x and y having integer values [8].

#### **1-3-1 24-Bit Image**

It can have 16,777,216 ( $2^{24}$ ) possible colors. All color variations for the pixels are derived from three primary color (Red, green, and Blue) as described in figure (1-1). In 24-bit color images each primary color is represented by one byte (8 bit) each byte represents the intensity of the

color and ranges from 0 to 255. The darkest intensity or color value is 0 and the brightest value is 255 [32].

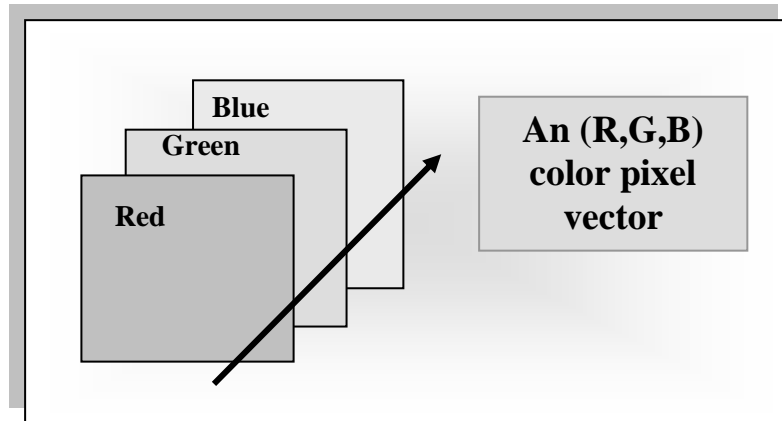


Figure (1-1) Color image representation [36].

### **1-3-2 8-Bit Images**

They may be either 256 color or grayscale images. Gray scale images are referred to as monochrome, or one-color, images they contain brightness information only (no color information). The number of bit used for each pixel determines the number of different brightness levels available. The typical image contains 8 bits/pixel data, which allows is to have 256, (0-255) different brightness (gray) levels this representation provides more than adequate brightness resolution [36].

### **1-3-3 1-Bit Images**

Binary images are the simplest type of images and can take on two values, typically black and white, or '0' and '1'. A binary image is referred to as a 1-bit/pixel image because it takes only 1 binary digit to represent each pixel. These types of images are most frequently used in computer vision applications where the only information required for the task is general shape, or outline, information [36].

## **1-4 Digital Image File Format**

Why do we need so many different types of image file formats? The short answer is that there are many different types of images and applications with varying requirements. A more complete answer also considers market share, proprietary information, and a lack of coordination within the imaging industry [4, 35]. Three types of images will be introduced below:

### **1-4-1 Bitmap (BMP)**

Bitmap (BMP) file format is used for bitmap graphics on the window platform only. Unlike other file formats, which store image data from top to bottom and pixels in Red, Green, Blue order, the BMP format stores image data from bottom to top and pixels in Blue, Green, Red order. This means that if memory is tight, BMP graphics will sometimes appear drawn from bottom to top. Compression of BMP files is not supported, so they are usually very large. When saving a file in the BMP format, add the “.BMP” file extension to the end of its file name.

### **1-4-2 Graphics Interchange Format (GIF)**

Graphics Interchange Format (GIF) was originally developed by CompuServe in 1987. It is one of the most popular file formats for Web graphics and for exchanging graphics files between computers. It is most commonly used for bitmap images composed of line drawings or blocks of a few distinct colors. The GIF 89 a file format supports transparency, allowing you to make a color in your image transparent (CompuServe GIF 87 does not support transparency). This feature makes

GIF a particularly popular format for Web images. When saving an image to the GIF format, add the “.GIF” file extension to the end of its file name.

### **1-4-3 Joint Photographic Experts Group (JPEG)**

JPEG like GIF, the Joint Photographic Experts Group (JPEG) format is one of the most popular formats for Web graphics. It supports 24 bits of color information, and is most commonly used for photographs and similar continuous-tone bitmap images. The JPEG file format stores all of the color information in an RGB image, then reduces the file size by compressing it, or saving only the color information that is essential to the image. Most imaging applications and plug-in let you determine the amount of compression used when saving a graphic in the JPEG format. Unlike GIF, JPEG does not support transparency. When saving a file in the JPEG format, add the “.JPG” file extension to the end of its file name.

## **1-5 The BMP File Format**

The BMP file structure is very simple and is shown in Figure (1-2).

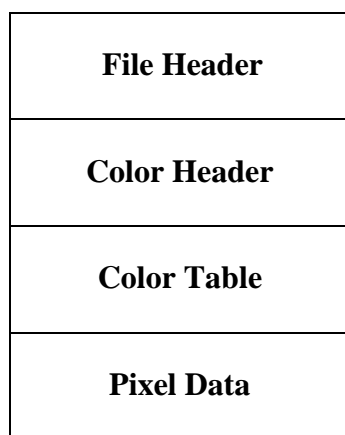


Figure (1-2) Bitmap File Structure [4].

### **1-5-1 File Header**

Every Windows BMP begins with a BITMAPFILEHEADER structure whose layout is shown in table (1-1). The main function of this structure is to serve as the signature that identifies that file format [4].

Three checks can be made to ensure that the file you are reading is in fact a BMP file:

- The first two bytes of the file must contain the ASCII characters “B” followed by “M”.
- If you are using a file system where you can determine the exact file size in bytes, you can compare the file size with the value in the bfSize field.
- The bfReserved1 and bfReserved2 fields must be zero.

The file header also specifies the location of the pixel data in the file. When decoding a BMP file you must use the bfOffbits field to determine the offset from the beginning of the file to where the pixel data starts. Most applications place the pixel data immediately following the BITMAPINFOHEADER structure or palette, if it is present. However, some applications place filler bytes between these structures and the pixel data so you must use the bfOffbits to determine the number of bytes from the BITMAPFILEHEADER structure to the pixel data.

Table (1-1) BITMAPFILEHEADER Structure [4].

<b>Field Name</b>	<b>Size in Bytes</b>	<b>Description</b>
bfType	2	Contains the characters “BM” that identify the file type.
bfsize	4	File Size.
bfReserved1	2	Unused.
bfReserved2	2	Unused.
BfOffBits	4	Offset to Start of Pixel Data.

### **1-5-2 Image Header**

The image header immediately follows the BITMAPFILEHEADER structure. It comes in two distinct formats, defined by the BITMAPINFOHEADER and BITMAPCOREHEADER structures. BITMAPCOREHEADER represents the OS/2 BMP format and BITMAPFILEHEADER is the much more common Windows format.

The only way to determine the type of image structure used in a particular file is to examine the structure's size field, which is the first 4 bytes of both structure types. The size of the BITMAPCOREHEADER structure is 12 bytes; the size of BITMAPINFOHEADER, at least 40 bytes.

The layout of BITMAPINFOHEADER is shown in table (1-2). This structure gives the dimensions and bit depth of the image and tells if the image is compressed. Windows 95 supports a BMP format that uses an enlarged version of this header. Few applications create BMP files using this format; however, a decoder should be implemented so that it knows that header sizes can be larger than 40 bytes.

The image height is an unsigned value. A negative value for the biHeight field specifies that the pixel data is ordered from the top down rather than the normal bottom up. Images with a negative biHeight value may not be compressed [4].

Table (1-2) BITMAPINFOHEADER Structure [4].

Field Name	Size	Description
biSize	4	Header size-Must be at least 40.
biWidth	4	Image width.
biHeight	4	Image height.
biPlanes	2	Must be 1.
biBitCount	2	Bits per pixel-1, 4, 8, 16, 24 or 32.
biCompression	4	Compression type-BI_RGB=0, BI_RLE8=1, BI_RLE4=2, or BI_BITFIELDS=3.
biSizeImage	4	Image Size-May be zero if not compressed.
bixPelsPerMeter	4	Preferred resolution in pixels per meter.
biyPelsPerMeter	4	Preferred resolution in pixels per meter.
biClrUsed	4	Number of entries in the color map that are actually used.
biClrImportant	4	Number of significant colors

The BITMAPCOREHEADER structure is the other image header format. Its layout is shown in table (1-3).

Table (1-3) BITMAPCOREHEADER Structure [4].

Field Name	Size	Description
bcSize	4	Header size-Must be 12
bcwidth	2	Image width
bcHeight	2	Image height
bcPlanes	2	Must be 1
bcBitCount	2	Bit count-1, 4,8, or 24

Notice that it has fewer fields and that all have analogous fields in the Bitmap Info Header structure. If the file uses

BITMAPCOREHEADER rather than BITMAPINFOHEADER, the pixel data cannot be compressed.

### **1-5-3 Image Data Area**

This area contains image data, this data represents the color values for each pixel in the image. The image data are stored from first pixel in the bottom row in the image, from left to right, to the last pixel in the top row, that means the rows of image are stored in reverse form in the file, starting from bottom row to the top row, the pixels are stored inside each row from left to right, therefore the center pixel lies in the left bottom corner.

## **1-6 Encryption**

The art and science of keeping messages secure is cryptography, and it is practiced by cryptographers [34]. Cryptanalysts are practitioners of cryptanalysis, the art and science of breaking ciphertext. There are several approaches in cryptography such as classical methods, stream cipher, and block cipher and public key.

Cryptography involves transforming an original message so that any individual that happens to find the transformed message will not be able to understand it without the correct method that will reverse the transformation, usually through some contact/agreement with the original encryption.

## **1-7 Literature Survey**

Provos [27] carried out an extensive analysis of JPEG images downloaded from eBay. Using his steganalytic software he identified several thousands of “suspicious” images embedded with J-Steg and JP Hide&Seek. A dictionary attack was then applied in an attempt to recover



the hidden message. Although this experiment did not reveal presence of any secret messages, it does not prove that criminals are not using steganography for communication.

Fridrich [13,14] introduced the dual statistics steganalytic method for detection of LSB embedding in uncompressed formats. For high quality images taken with a digital camera or a scanner, the dual statistics steganalysis indicates that the safe bit-rate is less than 0.005 bits per sample, providing a surprisingly stringent upper bound on steganographic capacity of simple LSB embedding.

Pfitzmann and Westfeld [37] introduced a method based on statistical analysis of Pairs of Values (PoVs) that are exchanged during message embedding. Pairs of Values that differ in the LSB only, for example, could form these PoVs. This method provides very reliable results when we know the message placement (such as sequential). Provos [28] noted that the method could still be used for detection of randomly scattered messages by applying the same idea to smaller portions of the image. However, he gives no further details or estimates of false positives and negatives for this generalized approach.

Farid [11] developed a universal blind detection scheme that can be applied to any steganographic scheme after proper training on databases of original and cover-images. He uses an optimal linear predictor for wavelet coefficients and calculates the first four moments of the distribution of the prediction error. The statistics is calculated for a large database of original and stego-images. Fisher linear discriminate

statistical clustering is then used to find a threshold that separates stego-images from original images. Farid demonstrates the performance on J-Steg, both versions of OutGuess [28], and EZ Stego [3]. It appears that the selected statistics is rich enough to cover a very wide range of steganographic methods.

Al-hamami [2] in his work tried to design an efficient system to scan and test suspicious images to find out if it contains a secret message or not. The system try to extract the secret message (the secret message may be text or image) from the suspicious image (if it is possible) and make changes on it. When failed to extract the secret message or to prevent suspicion image for pass a secret message, the system has the ability to destroy the secret messages.

## **1-8 Aim of Thesis**

In this thesis, the following aims are achieved:

1. The most important steganalysis methods and types are introduced in detail.
2. The sequential, non-sequential and algorithm of LSB BMP image hiding are suggested in order to be detected and then broken by new steganalysis methods.
3. A new proposed steganalysis system is constructed to detect and break the LSB BMP images regardless.

## *Chapter Two*

# **Information Hiding**

## **Chapter Two**

### **Information Hiding**

#### **2-1 Introduction**

Digital communication has become an essential part of infrastructure nowadays, a lot of applications are Internet-based and in some cases it is desired that the communication be made secret. Two techniques are available to achieve this goal: one is cryptography, where the sender uses an encryption key to scramble the message, this scrambled message is transmitted through the insecure public channel, and the reconstruction of the original, unencrypted message is possible only if the receiver has the appropriate decryption key. The second method is steganography [33].

Steganography encompasses methods of transmitting secret messages through innocuous cover carriers in such a manner that the very existence of the embedded messages is undetectable. Creative methods have been devised in the hiding process to reduce the visible detection of the embedded messages.

Hiding information, where electronic media are used as such carriers, requires alterations of the media properties which may introduce some form of degradation. If applied to images that degradation, at times, may be visible to the human eye and point to signatures of the steganographic methods and tools used. These signatures may actually broadcast the existence of the embedded message, thus defeating the purpose of steganography, which is hiding the existence of a message [4].

#### **2-2 Information Hiding Preview**

One of the newest hot spots in security research is information hiding. It is driven by two of the bluest policy issues of the information age-copyright protection and state surveillance [21].

Information hiding (literally, covered writing) is the hiding of secret messages within another seemingly innocuous message, called (host signal), data or carrier. The carrier can be anything used to transfer information, including for example, wood or slate tablets hollow heels, images under stamps, and tiny photographs, or word arrangements. Digital carriers include e-mail, audio, and video messages, disk space, disk partitions, and images [18].

The general model of hiding data in other data can be described as follows: The embedded data is the message that one wishes to send secretly. It is usually hidden in an innocuous message referred to as a cover-text.

Cover-image or cover-audio as appropriate, produces the stego-text or other stego-object. A stego-key is used to control the hiding process so as to restrict detection and/or recovery of the embedded data to parties who know it (or who know some derived key value) [10].

One of the information hiding classifications is shown in Figure (2-1).

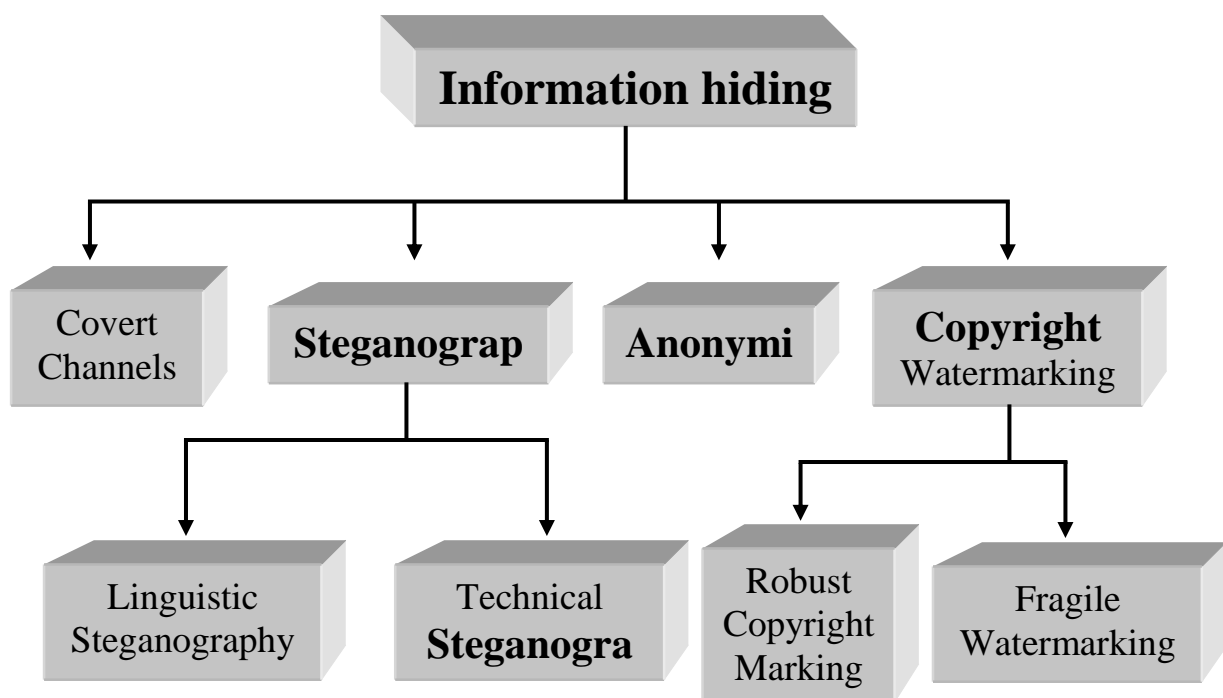


Figure (2-1) A Classification of information hiding techniques [18].

Covert channels have been defined by Lampson [6], in the context of multilevel secure systems (e.g. military computer system), as communication paths that were neither designed nor intended to transfer information at all. These channels are typically used by entrust worthy programs to leak information to their owner while performing a service for another program.

Anonymity is finding ways to hide the met content of messages, that is, the sender and the recipients of a message. Note that there are different variants depending on who is the “anonymized” sender, receiver, or both. Web applications have focused on receiver anonymity while email users are concerned with sender anonymity [21].

Digital watermarking refers to the embedding of unobtrusive marks or labels that can be represented as bits in digital content. Currently watermarking technology is not yet strong enough to withstand potential attacks and there are no watermarking technology standards in place to make it a legitimate form content protection. Historically, a watermark is a slight imprint on paper that is nearly imperceptible unless viewed carefully under the proper condition.

Watermarking as opposed to steganography, has the additional requirement of robustness against possible attacks. The term “robustness” mainly depends on the application, but a successful attack will simply try to make the mark undetectable.

As audio, video, and other works become available in digital form the ease with which perfect copies can be made, may lead to large-scale unauthorized copying which might undermine the music, film, book, and software publishing industries.

These concerns over protecting copyright have triggered significant research to find ways to hide copyright messages and serial numbers into digital media; the ideas that the latter can help to identify copyright violators, and the former to prosecute them.

Watermarking does not always need to be hidden, as some systems use visible digital watermarks. But most of the literature has focused on imperceptible (invisible, transparent, or inaudible, depending on the context) digital watermarks which have wider applications. Visible digital watermarks are strongly linked to the original paper watermarks, which appeared at the end of the 13th century.

The differences between steganography and watermarking is that the information hidden by a watermarking system is always associated to the digital object to be protected to its owner, while a steganography system just hides any information [21].

Steganography is the study of the techniques used to hide one message inside another, without disclosing the existence of the hidden message or making it apparent to an observer that message containing the hidden message has been altered while digital watermarking uses techniques from both cryptography and steganography. The major difference between classical steganography and digital watermarking is that steganography tries to hide a message within some other content, while digital watermarking focuses on preventing attackers from manipulating with the watermark, without necessarily worrying whether the attackers know about the presence of the watermark.

The “robustness” criteria are also different, since steganography is mainly concerned with detection of the hidden message while watermarking concerns potential removal by a pirate.

Finally steganographic communications are usually point-to-point (between sender and receiver) while watermarking techniques are usually one-to-many [27].

A fingerprint, like an embedded serial number, provides the ability to trace illegally distributed content back to the individual who initiated the illegal distribution. When a legal copy is originally obtained from a legitimate distributor, a fingerprint is embedded that links the purchaser to the copy. If the purchaser proceeds to produce and distribute illegal copies, every copy that he distributes will contain his fingerprint [10].

## **2.3 Methods of Hiding Information**

Methods of hiding information can be grouped into three broad categories: injection, substitution and the generation of new Files.

### **2.3.1 Injection**

Injection refers to the insertion of a message into an existing medium. The simple example is the use of the hidden attribute in Microsoft Word, which allows for hiding text with a special, hidden font. This simple technique was used to store notes and references during the creation of the document.

The HTML language allows for the hidden attribute that works in a similar fashion by hiding text from a web browser. Moving up the technology scale and into the security arena it could be argued that the Unicode vulnerability, a technique that has corrupted many web servers by hiding commands in unprintable pieces of web addresses is also a form of stego [2].



### **2.3.2 Substitution**

This technique replaces data in the original File with a coded representation of the original message. The colors of "pixels", tiny elements of digital images are often represented by the value of a number contained in an eight-bit byte of data. For example, three increasingly redder shades of red might be represented as follows:

“00001100” or decimal 12 might represent basic red in a particular 8-bit color palette. Each of the following numbers would then represent a minor increase in the redness.

“00001101” or decimal 13

“00001110” or decimal 14

The likelihood of a casual observer noticing the difference in the shades in the middle of a picture is very slight. The result is that steganographers are able to use the 2 least significant bits to encode messages and while the image does degrade slightly, it is not apparent to the naked eye.

The complexity of the techniques goes well beyond manipulation of least significant bits and extends to sophisticated processes such as manipulating the discrete cosine transformation process used to create JPEG files. Other techniques, such as those used in digital watermarking manipulate other image properties such as luminance. Luminance is often chosen as the Human Visual System (HVS) and has a lower sensitivity to changes in luminance than other image characteristic [2].

### **2.3.3 Generation of New File**

Both insertion and substitution require a host file, sometimes called a container, in references to images, and a host signal in reference to audio

signals. Host files may contain embedded message but may also exhibit characteristics that reveal a pattern that can be used by steganalysis tools to detect the presence of the message. To eliminate this potential weakness, a coded message can be generated as part of an original computer-generated text, audio or image file [2].

## **2-4 Basic Model of Steganography System**

Each steganographic technique consists of an embedding algorithm and a detection function. The embedding algorithm is used to hide secret message inside a cover (or carrier) document. The embedding process is usually protected by a keyword so that the only one who possesses the secret keyword can access the hidden message. The detector function is applied to the stego-document and results the hidden secret message. A possible formula of the process may be represented as:

**Cover medium + embedded message + stegokey = stegomedium**

Figure (2-2) shows the general acceptable model of a steganography system.

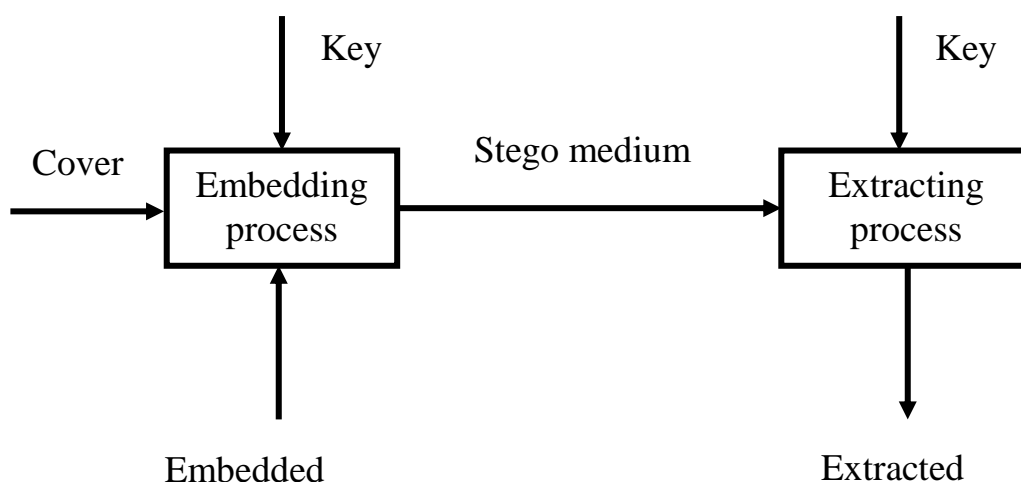


Figure (2-2) General steganography system (stego-system) [35].

For secure covert communication, it is important that by injecting a secret message into a cover document no detectable changes are introduced. The main goal is not to raise suspicion and avoid introducing statistically detectable modifications into the stego-document. The quantity of embedded data and the degree of host signal modification vary from application to application [35].

## **2-5 Steganography Through History**

A story from ancient Greece also comes to us via Herodotus. The writing medium of the time was text, written on wax-covered tablets. Demeratus, a Greek, needed to notify Sparta that Xerxes intended to invade Greece. To avoid capture, he scraped the wax off of the tablets and wrote the message on the underlying wood. Then he covered the tablets with wax again. The tablets appeared to be blank and unused so they passed inspection [4].

Invisible inks have always been a popular method of steganography. Ancient Romans used to write between lines using invisible inks based on readily available substances such as fruit juices, urine and milk. When heated, the invisible inks would darken, and become legible. Invisible inks were used as recently as World War II [21].

Document layout was also used to reveal information. By modulating the position of lines and words, messages could be marked and identified. Techniques such as writing messages in typewriter correction ribbon, and using pin punctures to mark selected letters were used.

With the computer age, steganography has been given a marvelous boost. Old methods, such as hiding messages in images, have been given new leases of life through the computer. We are sure to see a great expansion of steganographical techniques in the coming years [33].

## **2-6 Steganography under Various Media**

Often, although it is not necessary, the hidden messages will be encrypted. This meets a requirement posed by the “Kerckhoff principle” in cryptography. This principle states that the security of the system has to be based on the assumption that the enemy has full knowledge of the design and implementation details of the steganographic system. The only missing information for the enemy is a short, easily exchangeable random number sequence, the secret key. Without this secret key, the enemy should not have the chance to even suspect that on an observed communication channel, hidden communication is taking place.

When embedding data, Bender et al. remind us that it is important to remember the following restrictions and features [19]:

- The cover data should not be significantly degraded by the embedded data, and the embedded data should be as imperceptible as possible. (This does not mean the embedded data needs to be invisible; it is possible for the data to be hidden while it remains in plain sight.)
- The embedded data should be directly encoded into the media, rather than into a header or wrapper, to maintain data consistency across formats.
- The embedded data should be as immune as possible to modifications from intelligent attacks or anticipated manipulations such as filtering and resampling.

- Some distortion or degradation of the embedded data can be expected when the cover data is modified. To minimize this, error-correcting codes should be used.
- The embedded data should be self-clocking or arbitrarily re-entrant. This ensures that the embedded data can still be extracted when only a portion of the cover data is available. For example, if only a part of image is available, the embedded data should still be recoverable.

The Internet is a vast channel for the dissemination of information that includes publications and images to convey ideas for mass communication. Images provide excellent carriers for hidden information and many different techniques have been introduced. The computer technology and the Internet have given new life to Steganography and the creative methods with which it is employed. Computer-based steganographic techniques introduce changes to digital carriers to embed information foreign to the native carriers. Since 1995, interest in steganographic methods and tools as applied to digital media has exploded. Carriers of such message may resemble innocent sounding text, disks and storage devices, network traffic and protocols, the way software or circuits are arranged, audio, images, video, or any other digitally represented code or transmission. These provide excellent carriers for hidden information.

### **2-6-1 Hiding in Text**

Documents may be modified to hide information by manipulating positions of line and words. HTML file can be used to carry information since adding spaces, tabs, “invisible” characters, and extra line breaks are ignored by web browsers. The “extra” spaces and lines are not perceptible until revealing the source of the web page.

Another example of hiding information in text is known as a null cipher or open code. The secret message is camouflaged in an innocent sounding message [14]. The following is such a null cipher:

**Fishing freshwater bends and saltwater coasts rewards anyone  
feeling stressed. Resourceful anglers usually find masterful  
leaders fun and admit swordfish rank overwhelming anyday.**

Taking the third letter in each word the following message emerges.

**Send Lawyers Guns and Money**

If the cover-text is transmitted in a formatted form (like HTML or aPostScript file) information can be embedded in the format rather than in the message itself.

Secret information can be stored in the size of interline or interword spaces [9]. If the space between two lines is smaller than some threshold, a "0" is encoded, otherwise a "1".

### **2-6-2 Hiding in Disk Space**

Other ways to hide information by taking advantage of unused or reserved space to hold covert information provide a means of hiding information without perceptually degrading the carrier. The way operating systems store files typically results in unused space that appears to be allocated to a file. For example, under Windows 95 operating system, drives formatted as FAT16 (MS-DOS compatible) without compression use cluster size of around 32 kilobytes (k). What this means is that the minimum space allocated to file is 32K. If a file is 1K in size, then an additional 31K is "wasted" due to the way storage space is allocated. This "extra" space can be used to hide information without showing up in the directory [18].

### **2-6-3 Hiding in Network Packets**

Characteristics inherent in network protocols can take advantage of hiding information. An uncountable number of data packets are transmitted daily over the Internet, any of which can provide an excellent covert communication channel. For example, TCP/IP packets can be used to transport information across the Internet. The headers of these packets have unused space and other features that can be manipulated to embed information [17].

### **2-6-4 Hiding in Audio and Images**

Many different methods for hiding information in audio and images exist. These methods may include hiding information in unused space in file headers to hold “extra” information. Embedding techniques can range from the placement of information in imperceptible levels (noise), manipulation of compression algorithms, to the modification of carrier properties. In audio, small echoes or slight delays can be added or subtle signals can be masked by sounds of higher amplitude [18].

## **2-7 Hiding in Images**

In this section we deal with data encoding in still digital images. In essence, image steganography is about exploiting the limited powers of the human visual system (HVS) [36]. Within reason, any plaintext, ciphertext, other images, or anything that can be embedded in a bit stream can be hidden in an image. Image steganography has come quite far in recent years with the development of fast, powerful graphical computers, and steganographic software is now readily available over the Internet for everyday users.

### **2-7-1 Some Guidelines to Image Steganography**

To a computer, an *image* is an array of numbers that represent light intensities at various points, or *pixels*. These pixels make up the image's *raster data*. An image size of 640 by 480 pixels, utilizing 256 colors (8 bits per pixel) is fairly common. Such an image would contain around 300 kilobits of data [35].

Digital images are typically stored in either 24-bit or 8-bit per pixel files. 24-bit images are sometimes known as *true color* images. Obviously, a 24-bit image provides more space for hiding information; however, 24-bit images are generally large and not that common. A 24-bit image 1024 pixels wide by 768 pixels high would have a size in excess of 2 megabytes. As such large files would attract attention were they to be transmitted across a network or the Internet, image compression is desirable. However, compression brings with it other problems, as will explained shortly.

Alternatively, 8-bit color images can be used to hide information. In 8-bit color images, (such as GIF files), each pixel is represented as a single byte. Each pixel merely points to a color index table, or *palette*, with 256 possible colors. The pixel's value, then, is between 0 and 255. The image software merely needs to paint the indicated color on the screen at the selected pixel position.

In using an 8-bit image as the cover-image, many steganography experts recommend using images featuring 256 shades of *grey* as the palette, for reasons that will become apparent. Grey-scale images are preferred because the shades change very gradually between palette entries. This increases the image's ability to hide information.



When dealing with 8-bit images, the steganographer will need to consider the image as well as the palette. Obviously, an image with large areas of solid color is a poor choice, as variances created by embedded data might be noticeable. Once a suitable cover image has been selected, an image encoding technique needs to be chosen.

### **2-7-2 Image Encoding Techniques**

Information can be hidden in many different ways in images. Straight message insertion can be done, which will simply encode every bit of information in the image. More complex encoding can be done to embed the message only in “noisy” areas of the image, that will attract less attention. The message may also be scattered randomly throughout the cover image [35].

The most common approaches to information hiding in images are:

- ☐ Least significant bit (LSB) insertion.
- ☐ Masking and filtering techniques.
- ☐ Algorithms and transformations.

Each of these can be applied to various images, with varying degrees of success. Each of them suffers to varying degrees from operations performed on images, such as cropping, or resolution decrementing, or decreases in the color depth.

### **2-7-2-1 Least Significant Bit Insertion**

The least significant bit insertion method is probably the most well-known image steganography technique. It is a common, sample approach to embedding information in a graphical image file. Unfortunately, it is extremely vulnerable to attacks, such as image manipulation. A simple

conversion from GIF or BMP format to a lossy compression format such as JPEG can destroy the hidden information in the image.

When LSB techniques are applied to each byte of a 24-bit image, three bits can be encoded into each pixel. Any changes in the pixel bits will be indiscernible to the human eye. For example, the letter A can be hidden in three pixels. Assume the original three pixels are represented by the three 24-bit words below:

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

The binary value for the letter A is.

0	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---

Inserting the binary value of A into the three pixels, starting from the top left byte, would result in:

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

The emphasized bits are the only bits that actually changed. The main advantage of LSB insertion is that information can be hidden in the least and second two least bits and still the human eye would be unable to notice it [18].

When using LSB techniques on 8-bit images, more care needs to be taken, as 8-bit formats are not as forgiving to data changes as 24-bit formats are. Care needs to be taken in the selection of the cover image, so

that changes to the data will not be visible in the stego-image. Commonly known images, (such as famous paintings, like the Mona Lisa) should be avoided. In fact, a simple picture of your dog would be quite sufficient.

When modifying the LSB bits in 8-bit images, the pointers to entries in the palette are changed. It is important to remember that a change of even one bit could mean the difference between a shade of red and shade of blue. Such a change would be immediately noticeable on the displayed image, and is thus unacceptable. For this reason, data-hiding experts recommend using grey-scale palettes, where the differences between shades is not as pronounced.

### **2-7-2-2 Masking and Filtering**

Masking and filtering techniques hide information by marking an image in a manner similar to paper watermarks. Because watermarking techniques are more integrated into the image, they may be applied without fear of image destruction from lossy compression. By covering, or masking a faint but perceptible signal with another to make the first non-perceptible, we exploit the fact that the human visual system cannot detect slight changes in certain temporal domains of the image.

Technically, watermarking is not a steganographic form. Strictly, steganography conceals data in the image; watermarking extends the image information and becomes an attribute of the cover image, providing license, ownership or copyright details.

Masking techniques are more suitable for use in lossy JPEG images than LSB insertion because of their relative immunity to image operations such as compression and cropping [35].

### **2-7-2-3 Algorithms and Transformations**

Because they are high quality color images with good compression, it is desirable to use JPEG images across networks such as the Internet. Indeed, JPEG images are becoming abundant on the Internet.

JPEG images use the Discrete Cosine Transform (DCT) to achieve compression. DCT is a lousy compression transform, because the cosine values cannot be calculated precisely, and rounding errors may be introduced. Variances between the original data and the recovered data depend on the values and methods used to calculate the DCT.

Images can also be processed using fast Fourier transformation and wavelet transformation. Other properties such as luminance can also be utilised. The HVS has a very low sensitivity to small changes in luminance, being able to discern changes of no less than one part in thirty for random patterns. This figure goes up to one part in 240 for uniform regions of an image.

Modern steganographic systems use spread-spectrum communications to transmit a narrowband signal over a much larger bandwidth so that the spectral density of the signal in the channel looks like noise.

The former hides information by phase-modulating the data signal (carrier) with a pseudo random number sequence that both the sender and the receiver know. The latter divides the available bandwidth into multiple channels and hops between these channels (also triggered by a pseudo random number sequence) [35].

## **2-8 Some Applications of Information Hiding**

There are many applications of information hiding, most of which, tend to hide information from human senses in order to put a level of security in their jobs [24]. Some of these applications are:

1. Copyright protection, which uses techniques of information hiding (like watermarks) to hide a mark or signature of the manufacturer.
2. Currency industries, which hide watermarks, hard curves, or another type of information in order to prevent currency forgery. These types of information cannot be copied.
3. Secure transmission of messages by hiding them in a cover (image, text, audio, video, ...etc.), if any one tries to steal data from the transmission channel, he cannot see the secure hidden information, but only the cover.
4. Secure saving of data inside computer by using the previous techniques and apply it on the saved data. If any one enters to the computer, he will see only the covers, not the hidden information.

## *Chapter Three*

# **Steganalysis**

## Chapter Three

### Steganalysis

#### 3-1 Introduction

Hiding information in digital media requires alterations of the media properties, which may introduce some form of degradation or unusual characteristics. The degradation, at times, may become perceptible. These characteristics may act as signatures that broadcast the existence of the embedded message and steganography tools used, thus defeating the purpose of steganography, which is hiding the existence of a message.

A goal of steganography is to avoid drawing suspicion to the transmission of a hidden message. If suspicion is raised, then this goal is defeated. Steganalysis is the art of discovering and rendering useless such covert messages. The *Steganalyst* is one who applies steganalysis in an attempt to detect the existence of hidden information and/or render it useless. This chapter identifies characteristics in current steganography software that direct the steganalyst to the existence of a hidden message [2].

#### 3-2 Steganalysis [4, 12, 28]

*Steganalysis* is the art of discovering hidden data in cover objects. As in cryptanalysis, we assume that the steganographic method is publicly known with the exception of a secret key. The method is secure if the stego-images do not contain any detectable artifacts due to message embedding. In other words, the set of stego-images should have the same

statistical properties as the set of cover-images. If there exists an algorithm that can guess whether or not a given image contains a secret message with a success rate better than random guessing, the steganographic system is considered broken.

The ability to detect secret messages in images is related to the message length. Obviously, the less information we embed into the cover-image, the smaller the probability of introducing detectable artifacts by the embedding process. Each steganographic method has an upper bound on the maximal safe message length (or the bit-rate expressed in bits per pixel or sample) that tells us how many bits can be safely embedded in a given image without introducing any statistically detectable artifacts.

Some steganographic utilities use secret keys. We can distinguish two kinds of keys: steganographic keys and cryptographic keys. A steganographic key controls the embedding and extracting process. A cryptographic key, however, is used to encrypt the message before it is embedded.

However, electronically each of these tools leaves a fingerprint or signature in the image that can be used to alert an observer to the presence of a hidden message. Discovering a hidden message is the first step in steganalysis and is considered an “attack” on the hidden information.

### **3-3 Classification of Steganography Attacks**

There are three main types of attacks, passive, active and malicious attack. As in most steganography systems, they change parts of the cover



media causing changes in the covers of statistical properties, the embedding process does not pay attention to this fact. A passive attacker could exploit this fact and break the system.

Active attacker is able to change a cover during the communication process. It is a general assumption that an active attacker is not able to change the cover and its semantics entirely; but only make minor changes so that original and the modified cover-object stay perceptually or semantically similar. An attacker is malicious if he forges message or starts steganography protocols under the name of one communication partner.

So, during the design of a steganography system attention has to be paid to the presence of passive attacker, since by detecting that a message is hidden then the other two attacks will start.

### **3-3-1 Passive Attack**

The passive attacker can detect the existence of a secret message by using different ways, the most common method is the discrete Laplace operator. By this operator it is possible to detect secret message in grayscale images:

$$\nabla^2 p(x,y) = p(x+1,y) + p(x-1,y) + p(x,y+1) + p(x,y-1) - 4p(x,y). \quad (3-1)$$

The value of the point (x,y) in (3-1) gives the “Laplace filtered” image. Since we can expect neighboring pixels to have a similar color, the histogram of Laplace filtered is tightly clustered around zero, which is shown in figure (3-1).

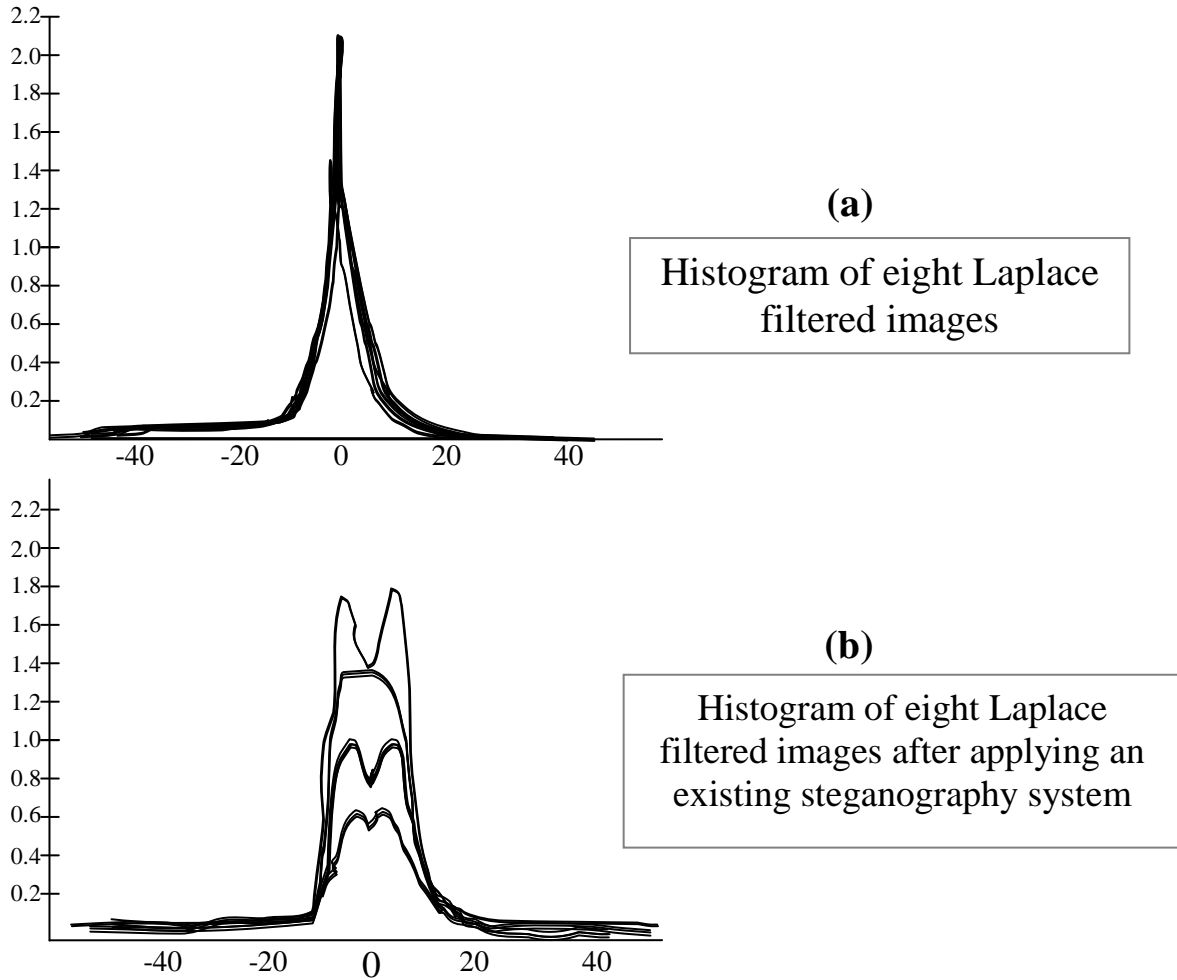


Figure (3-1) Histogram of Laplace filtered [22].

Figure (3-1-a) shows eight histograms of Laplace filtered grayscale images printed in one coordinate system. Figure (3-1-b) shows the histogram of the same image after applying an existing steganography algorithm. Since the embedding process adds noise to the picture, which is statistically quite different from the true random noise, the new histogram differs extremely. Laplace filtering does not prove the existence of a secret, but it will provide strong evidence that the picture was subject to modification.

### 3-3-2 Active Attack

An active attacker, who is not able to extract or prove the existence of a secret message, thus can simply add random noise to the transmitted cover and so tries to destroy the information. In the case of digital images, an attacker could also apply image processing techniques or convert the image to another file format. All of these techniques can be harmful to the secret communication. Another practical requirement for a steganography system, therefore, is robustness. A system is called **Robust** if the embedding information cannot be altered without making drastic changes to the stego-object [22].

### 3-3-3 Malicious Attack

In the presence of a malicious attacker, robustness is not enough. If the embedding method is not dependent on some secret information shared by sender and receiver, an attacker can forge messages, since the recipient is not able to verify the correctness of the sender's identity. Thus, to avoid such an attack, the algorithm must be robust and secure.

### 3-4 Types of Attacks [33, 23]

Attacks may come in several different forms depending on what information is available to the steganalyst:

- **Stego-only attack:** Only the stego-object is available for analysis. For example, only the stego-carrier and hidden information are available.
- **Known cover attack:** The original cover-object is compared with the stego-object and pattern differences are detected. For example, the original image and the image containing hidden information are available and can be compared.

- **Known message attack:** A known message attack is the analysis of known patterns that correspond to hidden information, which may help against attacks in the future. Even with the message, this may be very difficult and may be considered the same as a stego-only attack.
- **Chosen stego attack:** The steganography tool (algorithm) and stego-object are known. For example, the software and the stego-carrier and information are known.
- **Chosen message attack:** The steganalyst generates a stego-object from some steganography tool or algorithm of a chosen message. The goal of this attack is to determine corresponding pattern in the stego-object that may point to the use of specific steganography tools or algorithms.
- **Known stego attack:** The steganography tool (algorithm) is known and both the original and stego-object are available.

### 3-5 Steganalytic Methods [4]

With careful selection of an appropriate cover image and stego-tool it is possible to create a stego-image that does not appear to be different within the limits of human perception. However, electronically each of these tools leaves a fingerprint or signature in the image that can be used to alert an observer to the presence of hidden message. Bellow we explain visual and statistical analytic.

#### 3-5-1 Visual Analysis

The visual attack is a stego-only attack that exploits the assumption of most authors of steganography programs that the least significant bits

of a cover file are random. Relying on a human to judge if an image presented by a filtering algorithm contains hidden data, or does not. The filtering algorithm removes the parts of the image that are covering the message. The output of the filtering algorithm is an image that consists only of the bits that potentially could have been used to embed data. The filtering of the potential stego image is dependent on the steganographic embedding function that is analyzed. However, as most of the embedding functions are similar in most cases only small changes are necessary to adapt an existing filtering algorithm to another steganographic embedding function.

### **3-5-2 Statistical Analysis**

Visual attacks have two important drawbacks. If many images should be analyzed they are very slow or very costly because every image must be filtered, displayed and looked at by a human. The other important drawback is that some (unmodified) images might contain random looking data in its least significant bits. If such an image is used as cover file the visual attack will fail.

Statistical attack exploit- similar to visual attacks the fact that most steganography programs treat the least significant bits of the cover file as random data and therefore assume that they can overwrite these bits with other random data (the encrypted secret message). However as the visual attacks have showed the least significant bits of an image are not random. When a steganography program embeds a bit through overwriting the least significant bit of a pixel is changed to an adjacent color value in the palette (or in the RGB cube if the cover file is a true-color image). These simple tests are not able to decide automatically if an image contains a hidden message.

### 3-5-2-1 Known Cover Attack Using Mean Squared Error

The statistical analysis is very useful for detection of embedded data in an image. These analyses or tests can expose abnormalities in an image that are not visible by the human eyes. The statistical tests need the original and the suspected images for getting the correct results.

To calculate the Mean Squared Error (MSE) between the original image and suspected image, we must know the difference of pixel color in the both images. The result will be the square amount of errors depending on the size of these images.

The equation that is used to calculate (MSE) is:

$$MSE = \frac{I}{XY} \sum_{x,y} (o_{x,y} - \hat{o}_{x,y})^2 \quad (3-2)$$

X = number of rows

Y = number of columns

$O_{x,y}$  = value of pixel in the position x,y of cover-image

$\hat{O}_{x,y}$  = value of pixel in the position x,y of stego-image

### 3-5-2-2 Stego-Only Attack Using Chi-Square Test [18]

We now look at two adjacent color values (a Pair of Values, also referred to as PoV), where adjacent means identical except for the least significant bit: When overwriting the least significant bits of all occurrences of one of these color values with a bit from the secret message, the frequencies of these two color values will essentially be the same. This happens because the data that is embedded is encrypted and therefore equally distributed.

The idea of the statistical attack is to compare the frequency distribution of the colors of a potential stego file with the theoretically

expected frequency distribution for a stego file. The theoretically expected frequency distribution is calculated as follows: Under the assumption that only the least significant bits are overwritten and that the embedded data is equally distributed the expected frequency distribution is that for each PoV the frequencies of the two colors are the same. Due to the fact that the sum of the occurrences of the two colors in a PoV is not changed by the embedding process, the expected frequency can be calculated as the median of the frequencies of a PoV in the potential stego file.

The degree of similarity of the frequencies in the potential stego file and the theoretically expected frequencies is a measure for the probability that the analyzed file contains a hidden message.

Statistical tests can reveal if an image has been modified by steganography by testing whether an image's statistical properties deviate from a norm [4].

The Chi-square test is used to determine whether color frequency distribution in an image shows distortion from embedding hidden data. Because the test uses only the stego medium, the expected distribution  $y_i^*$  for the  $\chi^2$ -test has to be computed from the image. Let  $n_{2i}$  be the frequency of two adjacent color values in the image. We assume that an image with hidden data embedded has similar frequency for two adjacent color values. As a result, we can take the arithmetic mean:

$$y_i^* = \frac{n_{2i} + n_{2i+1}}{2} \quad (3-3)$$

to determine the expected distribution. The expected distribution is compared with the observed distribution

$$y_i = n_{2i} \quad (3-4)$$

the value for the difference between the distributions is given as:

$$\chi^2 = \sum_{i=1}^{v+1} \frac{(y_i - y_i^*)^2}{y_i^*} \quad (3-5)$$

where  $v$  is the degrees of freedom, that is, the number of different categories in the histogram minus one.

The probability  $p$  that the two distributions are equal is given by the complement of the cumulative distribution function,

$$p = 1 - \int_0^{\chi^2} \frac{t^{(v-2)/2} e^{-t/2}}{2^{v/2} \Gamma(v/2)} dt \quad (3-6)$$

where  $\Gamma$  is the Euler Gamma function.

For example the Chi-square test can be explained as follows:

Objective: is there a connection between two categorical variables?

- 1)  $H_0$  : Variable 1 is independent of Variable 2 (no relationship).
- 2)  $H_A$  : Variable 1 is related to Variable 2.
- 3) Choose significance level  $\alpha$  (typically, 0.05).

**Expected counts:** If there were no relationship, ideally we would have expected count for cell in row  $i$  and column  $j$ :

$$E_{ij} = \frac{\text{Row } i \text{ Total} * \text{Column } j \text{ Total}}{\text{Table Total}} \quad (3-7)$$



**Example:**

Blueberries and fungus. 100 berries were untreated, 25 were washed with water and 25 were washed with bleach. Number of those affected by fungi was recorded in table (3-1):

Table (3-1) the results of affected blueberries and fungus

	Affected	Healthy	Total
Untreated	46	54	100
Water	18	7	25
Bleach	10	15	25
Total	74	76	150

Variable 1: Treatment (Untreated/Water/Bleach).

Variable 2: Health of blueberries (Affected or Healthy).

$r = 3$  (number of rows, not counting Totals row).

$c = 2$  (number of columns).

The histogram of observed values shown in figure (3-2).

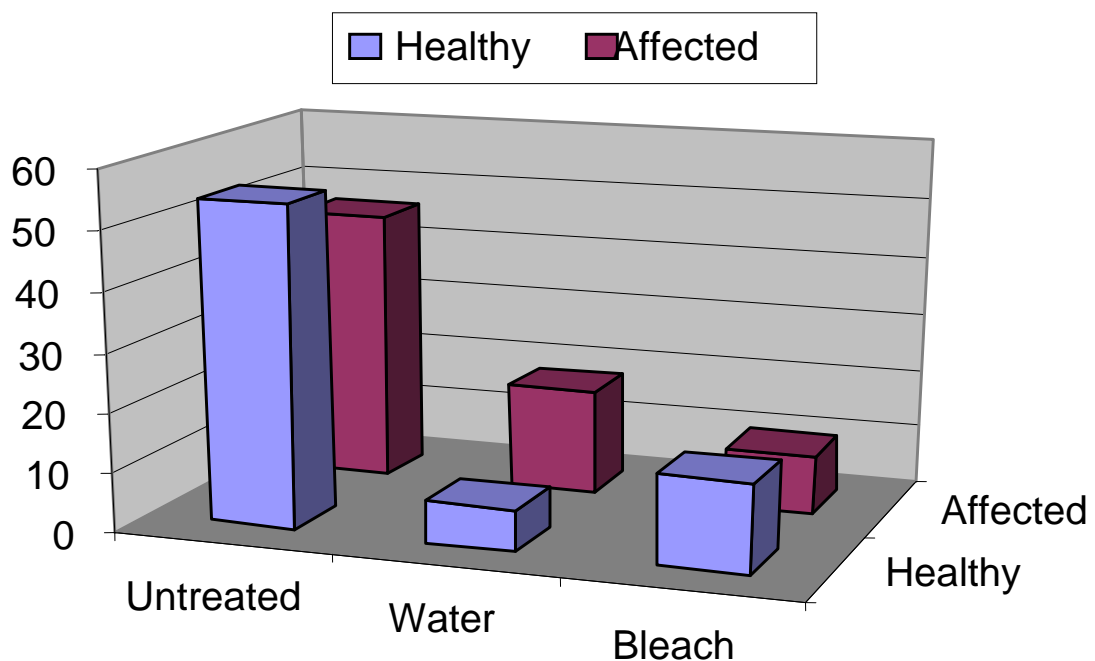


Figure (3-2) The histogram of observed values.

Expected counts  $E_{ij}$  is shown in Table (3-2).

Table (3-2) the expected counts  $E_{ij}$ .

	Affected	Healthy	Total
Untreated	$100 \cdot 74 / 150 = 49.3$	$100 \cdot 76 / 150 = 50.7$	<b>100</b>
Water	$25 \cdot 74 / 150 = 12.3$	$25 \cdot 76 / 150 = 12.7$	<b>25</b>
Bleach	$25 \cdot 74 / 150 = 12.3$	$25 \cdot 76 / 150 = 12.7$	<b>25</b>
Total	<b>74</b>	<b>76</b>	<b>150</b>

The histogram of the expected values is shown in figure (3-3).

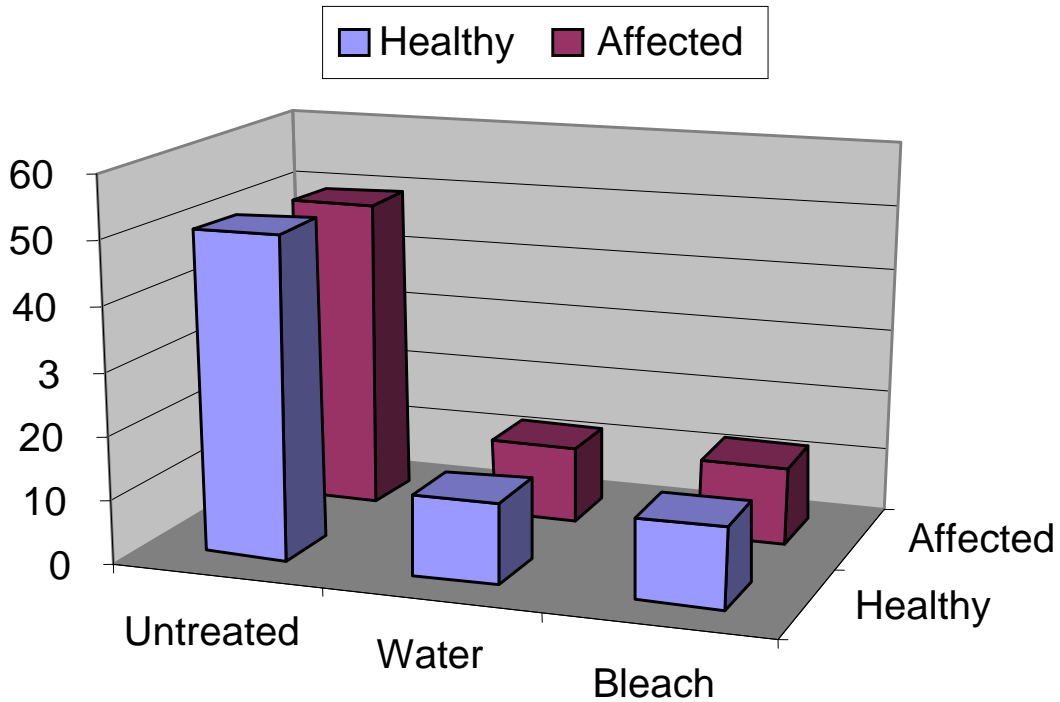


Figure (3-3) The histogram of expected values.

4) Chi-square statistic  $\chi^2 = \sum \frac{(O - E)^2}{E}$

(sum is taken over  $r \cdot c = 3 \cdot 2 = 6$  inner cells)

$$\begin{aligned}
 \chi^2 &= \sum_i \sum_j \frac{(O_{ij} - E_{ij})^2}{E_{ij}} = \frac{(46 - 49.3)^2}{49.3} + \frac{(54 - 50.7)^2}{50.7} + \frac{(18 - 12.3)^2}{12.3} + \\
 &\quad \frac{(7 - 12.7)^2}{12.7} + \frac{(10 - 12.3)^2}{12.3} + \frac{(15 - 12.7)^2}{12.7} \\
 &= 0.225 + 0.219 + 2.604 + 2.535 + 0.441 + 0.430 = 6.454
 \end{aligned}$$

The Chi-square  $\chi^2 = 6.454$ .

Degree of freedom:  $df = (r-1)(c-1) = (3-1)(2-1) = 2$ .

5) p-value: As 6.454 is between 5.99 and 7.38, which are taken from chi-square table the p-value is between 0.05 and 0.025.

When the equation (3-6) is applied by using a freedom degree (as  $v = 2$ ),

$$P = 1 - \int_0^{6.454} \frac{e^{-t/2}}{2\Gamma(1)} dt = 1 + \left[ e^{-t/2} \right]_0^{6.454} \approx 0.04 \quad (3-8)$$

6) p-value  $< 0.05$ . Reject  $H_0$ .

7) Conclusion: the health of berries is affected by their treatment  
(Not all treatments lead to the same result).

We can compute the probability of embedding for different parts of an image. The selection depends on what steganographic system we try to detect. For an image that does not contain any hidden information, we expect the probability of embedding to be zero everywhere.

### **3-6 Steps of Steganalysis of Images**

In general, steganalysis is carried out for breaking the security of a steganographic system. It is assumed that an adversary has the knowledge of the system and has one or many images are intercepted from a public channel. Security of the system lies solely on the secrecy of the key [2]. To make steganalysis difficult, a steganographic system is designed to have a large uncluttered key-space [25]. In the case of steganographic investigation, it is quite likely that details of the system are unknown. Analysis of non-standard systems is a complex activity and requires high level of expertise and massive infrastructure for computation of

steganographic and cryptographic keys. Figure (3-4) shows the general steganography and steganalysis process.

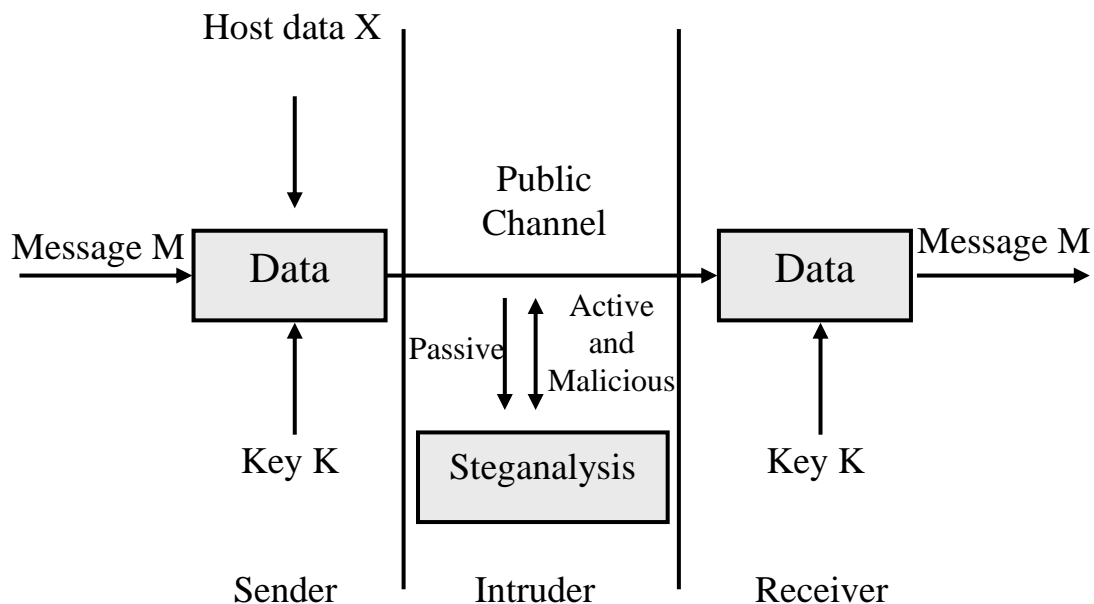


Figure (3-4) General steganography and steganalysis process [25].

Steganalysis consists of the following subtasks, one or more of which needs to be accomplished depending upon the nature of application [17]: detection, extraction, removal\destruction\disabling of the message and meaningful modification\insertion. Detection is the first and the most important part of steganalysis. Extraction of secret messages is required for knowing the contents and also the purpose of the hidden communication. This is required for gathering legal evidence for booking an offender or a criminal. An active adversary may extract the message in an attempt to disable or replace it by another message. The basic modules required for steganalysis are shown in figure (3-5).

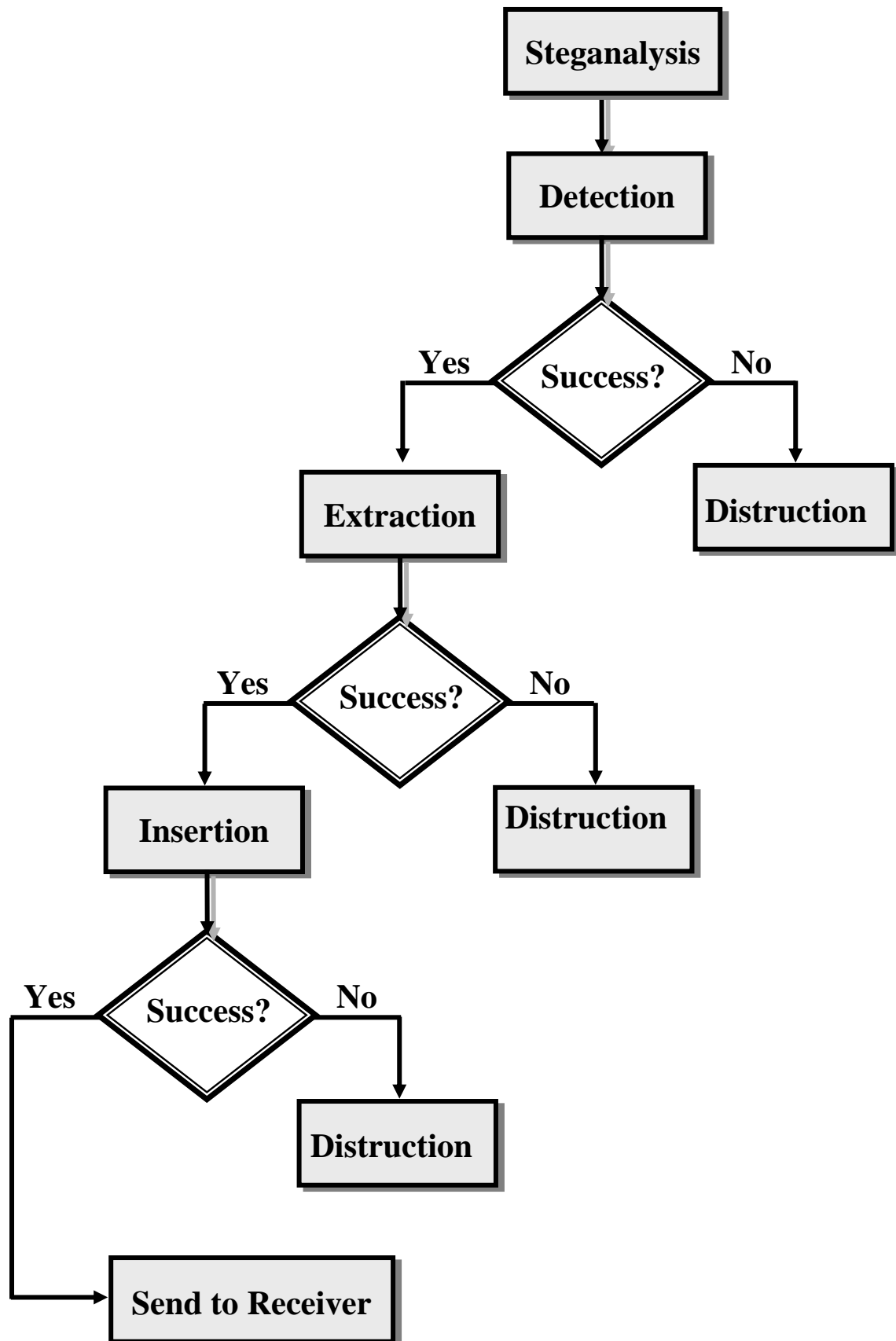


Figure (3-5) Basic modules in steganalysis [17].

Steganographic detection is carried out based on the knowledge of one or more of the following:

1. Stego-media.
2. Host media.
3. Embedded data.
4. Steganographic tool.

Attacks and analysis on hidden information may take several forms:

1. Detecting.
2. Extracting.
3. Disabling or destroying hidden information.

### **3-6-1 Detecting Hidden Information [8]**

Unusual patterns stand out and expose the possibility of hidden information.

In text, small shifts in word and line spacing may be somewhat difficult to detect by the casual observer. However, appended spaces and “invisible” characters can be easily revealed by opening the file with a common word processor. The text may look “normal” if typed out on the screen, but if the file is opened in a word processor, the spaces, tabs, and other characters distort the text’s presentation.

Images may display distortions from hidden information. Selecting the proper combination of steganography tools and carriers is key to successful information hiding. Some images may become grossly degraded with even small amounts of embedded information. This “visible noise” will give away the existence of hidden information. The same is true with audio. Echoes and shadow signals reduce the chance of audible noise, but they can be detected with little processing.

The image format and its size can also provide clues about the type of stego-tools used. The next step is to identify the signatures (if available) of specific tools in the image. Under normal circumstances, it is difficult to obtain the secret message or the host image. Capturing the equipment or the computer used by a suspect can give substantial information to the steganalyst but it is possible only in remote cases. Steganographic detection becomes simpler if the host image and the stego-image are available together [16].

### **3-6-2 Extracting**

Once steganographic contents are detected with high probability, the second step is of message extraction. In many countries it may be possible to obtain legal permission for getting the password required to extract the actual hidden information. Otherwise, the steganalyst may have to use a good amount of infrastructure to carry out a brute force or dictionary attack. Modification\insertion of meaningful messages can only be done after the previous activity is fully or partially successful.

### **3-6-3 Disabling and Modification Steganography [8, 22]**

Steganography systems are extremely sensitive to cover modifications, such as image processing techniques (like smoothing, filtering, and image transformations) in the case of digital image. But even a lousy compression can result in total information loss. Lossy compression techniques try to reduce the amount of information by removing imperceptible signal components and so often remove the secret information which has previously been added.

Detecting the existence of hidden information defeats the steganography's goal of imperceptibility. Methods exist which produce results, which are far more difficult to detect without the original image for comparison. At times the existence of hidden information may be known so detecting it is not always necessary. Disabling and rendering it useless seems to be the next best alternative. With each method of hiding information there is a trade off between the size of the payload (amount of hidden information) that can be embedded and the survivability or robustness of that information to manipulation.

The distortions in text noted by appended spaces and "invisible" characters can be easily revealed by opening the file with a word processor. Extra spaces and characters can be quickly stripped from text documents.

The disabling or removal of hidden information in images comes down to image processing techniques. For LSB method of inserting data, simply using lousy compression techniques, such as JPEG, is enough to render the embedded message useless. Images compressed with such a method are still pleasing to the human eye but no longer contain the hidden information.

Removal or destruction of hidden messages is easier when it is compared to detection and extraction. A number of methods are practiced for disabling a message embedded at unknown locations in a digital image.

These include image conversion, lossy compression, image-processing operations, geometric manipulations and Digital-Analog-Digital (DAD) conversion for which then print and scan operation are commonly used. An impatient attacker may inject noise in every image he encounters. Attacker may write a code (similar to a virus) [7] to modify the LSBs or other spectral coefficients in an image. The probability of message removal in this manner is the least as most of the steganographic systems are robust for simple form of attacks [16].



Conversion between 8-bit and 24-bit image formats may destroy a hidden message. Similarly, lossy compression changes the image data substantially without causing damage to its visual quality. Common signal processing operations used for active attacks [26] include noise insertion/reduction, re-sampling, re-quantization, brightening, darkening, sharpening and softening.

Geometric manipulations include rotation, scaling, translation and cropping. Design of the next generation of steganographic and watermarking tools demand robustness against all these attacks.

### **3-7 Differences Between Cryptanalysis and Steganalysis**

Some significant differences have been concluded between Cryptanalysis and Steganalysis, table (3-3) shows these differences between them [24].

Table (3-3) Differences between cryptanalysis and steganalysis [24].

<b>Cryptanalysis</b>	<b>Steganalysis</b>
Attempt to decode or crack encrypted messages.	Attempt to detect the existence of hidden information.
In cryptography, comparison is made between portions of the plaintext (possibly none) and portions of the ciphertext.	In steganography, comparison may be made between the cover-media, the stego-media, and possible portions of the message.
Attacks available to the cryptanalyst are ciphertext-only, known plaintext, chosen plaintext, and chosen ciphertext.	Attacks available to the steganalyst are stego-only, known cover, known message, chosen stego, chosen message and known stego attack.
Does not need the steganalysis.	If the message in steganography is encrypted, then after the message is extracted the cryptanalysis techniques is applied.

## *Chapter Four*

# **Efficient Information of Hiding System**

## **Chapter Four**

### **Efficient Information of Hiding System**

#### **4-1 Introduction**

Tools used to hide information in images vary in their approaches. Without knowing which tool is used for information hiding, detecting the hidden information may become quite complex. However, some tools produce stego-images with characteristics that act as signatures for the steganography method or tool used.

An approach used to identify such patterns is to compare the original cover-images with stego-images and note visible differences (known-cover attack) [21].

If the adjacent palette colors are very similar, there may be little or no noticeable change. However, if adjacent palette entries are dissimilar, then the noise due to the manipulation of the LSBs is obvious.

Detecting the existence of hidden information defeats the goal of imperceptibility. Some tools that hide information in more significant areas of images are more difficult to detect without the original image for comparison.

In this chapter, the proposed steganalysis system manipulates three different steganography tools, with or without cover image. The proposed steganalysis system is divided into two main subsystems, diagnosis and breaking. The hidden text may be plain or cipher text.

#### **4-2 Attacked Steganography Systems**

In this thesis, three kinds of steganography systems are introduced in true color BMP images, using LSB in image bytes. These systems differ from each other in style of hiding. Of course, the system tests the

size of the message they want to be hidden before hiding process is started and compared with container image size to be sure that the container can contain the data of the message.

It is important to mention that the message characters to be converted to binary are suitable for hiding process. Converting process uses Baudot encoding system [34]. Table (4-1) shows this encoding system.

Table (4-1) Baudot encoding system [34].

Index	Char.	Binary	Index	Char.	Binary
0	1	00000	16	E	10000
1	T	00001	17	Z	10001
2	2	00010	18	D	10010
3	O	00011	19	B	10011
4	3	00100	20	S	10100
5	H	00101	21	Y	10101
6	N	00110	22	F	10110
7	M	00111	23	X	10111
8	4	01000	24	A	11000
9	L	01001	25	W	11001
10	R	01010	26	J	11010
11	G	01011	27	5	11011
12	I	01100	28	U	11100
13	P	01101	29	Q	11101
14	C	01110	30	K	11110
15	V	01111	31	6	11111

The pseudo code of the character to binary converting algorithm is:

```

NAME: Character to Binary Converting Algorithm (C2BCA).
INPUT: Message data;
PROCESS   : Repeat
                Read message character(i);           {i=1..Message length}
                bit(i,j) = Baudot [character(i)];    {j=1..5}
                Write bit(i,j);
                Until EOF (message);
OUTPUT    : Encoded message;
END.

```

The pseudo code of the binary to character converting algorithm is:

```
NAME: Binary to Character Converting Algorithm (B2CCA).
INPUT: Message binary data;
PROCESS   : i = 0, j = 0;
            Repeat
                Read message bit(i);
                i = i+1;
                if i mod 5 = 0 then
                    j = j+1;
                    character(j)=
                        Baudot-1 [bit(i),bit(i-1),bit(i-2),bit(i-3),bit(i-4)];
                    Write character(j);
                endif;
            Until EOF (message);
OUTPUT    : Decoded message;
END.
```

The message could be plain or cipher text. In this thesis, the simple substitution is chosen as an example of an encipher system.

In simple substitutions, the alphabet is scrambled, and each plain text letter maps to a unique cipher text letter. Formally, a permutation is a reordering of the elements of a series.

The pseudo code of the simple substitution encipher algorithm is:

```
NAME: Simple Substitution Encipher Algorithm (SSEA).
INPUT: Plain Message;
PROCESS   : Repeat
                Read plain character(i);           {i=1..Message length}
                cipher character(i) = encipher table [plain character(i)];
                Write cipher character(i);
            Until EOF (message);
OUTPUT    : Encipher message;
END.
```

The pseudo code of the simple substitution decipher algorithm is:

```

NAME: Simple Substitution Decipher Algorithm (SSDA).
INPUT: Cipher Message;
PROCESS   : Repeat
            Read cipher character(i); {i=1..Message length}
            plain character(i) = decipher table [cipher character(i)];
            Write plain character(i);
            Until EOF (message);
OUTPUT    : Plain message;
END.

```

that the start hiding byte is byte number 1000. The three-steganography systems are discussed in the next three subsections.

### 4-2-1 Sequential Steganography System

In this system, the hiding is done in sequential style for every byte of the image data, that means, in every pixel three bits can be embedded of the message to be hidden.

The pseudo code of the sequential steganography algorithm is:

```

NAME: Sequential Steganography Algorithm (SSA).
INPUT: Cover image, plain message;
PROCESS   : If you want to hide encipher message CALL SSEA;
            CALL C2BCA;
            i = 1000, j = 1;
            Repeat
                Read Cbyte(i);    {Cbyte(i) = cover byte i}
                Read bit(j) from message;
                insert bit(j) in LSB of Cbyte(i);
                Write Cbyte(i) in cover;
                i = i+1, j=j+1;
            Until EOF (message);
OUTPUT    : Stego image;
END.

```

While the pseudo code of the sequential extracting algorithm is:

```

NAME: Sequential Extracting Algorithm (SEA).
INPUT: Stego image;
PROCESS   : i = 1000, j = 1;
            Repeat
                Read Sbyte(i);    {Sbyte(i) = stego byte i}
                extract bit(j) from Sbyte(i);
                Write bit(j) in message;
                i=i+1, j=j+1;
            Until EOF (message);
            CALL B2CCA;
            If message encipher CALL SSDA;
OUTPUT    : Plain message;
END.

```

### 4-2-2 Jumping Steganography System

In this system, hiding jump or fixed key jump (Keyjmp) must be specified firstly, the Keyjmp represents the number of jumped bytes from the current hiding byte to the next hiding one. The key jump (Keyjmp) is considered as a stego key of this stego tool.

The pseudo code of the jumping steganography algorithm is:

```

NAME: Jumping Steganography Algorithm (JSA).
INPUT: Cover image, Message plain text, Keyjmp;
PROCESS   : If you want to hide encipher message CALL SSEA;
            CALL C2BCA;
            i = 1000, j = 1;
            Repeat
                read Cbyte(i);
                read bit(j);
                insert bit(j) in LSB of Cbyte(i);
                write Cbyte(i) in cover;
                i=i + Keyjmp +1, j=j + 1;
            Until EOF(message);
OUTPUT    : Stego image;
END.

```

While the pseudo code of the jumping extracting algorithm is:

```

NAME: Jumping Extracting Algorithm (JEA).
INPUT: Stego image, Keyjmp;
PROCESS   : i = 1000, j = 1;
            Repeat
                Read Sbyte(i);
                extract bit(j) from Sbyte(i);
                Write bit(j) in message;
                i=i+Keyjmp+1, j=j+1;
            Until EOF (message);
            CALL B2CCA;
            If message encipher CALL SSDA;
OUTPUT    : Plain message;
END.

```

### 4-2-3 LFSR Steganography System

This system depends on a simple algorithm represented by Linear Feedback Shift Register (LFSR) with 11-stages length. This algorithm uses a stego key called Basic Key (BK), 10-bit length used as an initial value to LFSR and the last stage fills by (1), the BK must be known to both transmitter and receiver, and must be changed every period of time (weekly, daily or every message). When the LFSR is filled with initial values, it starts to move to generate pseudo random value which represents a key jump (Keyjmp) to specify the next hiding position and so on. The diagram of this steganography system is showed in figure (3-1).

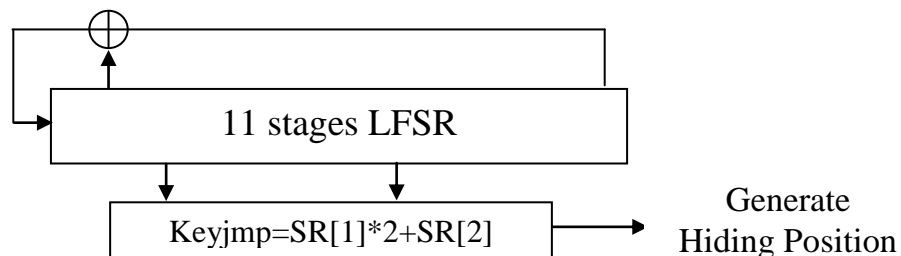


Figure (4-1) LFSR Steganography system.



The pseudo code of the LFSR steganography algorithm is:

```

NAME: LFSR Steganography Algorithm (LFSRSA).
INPUT: Cover image, plain message,
        BK (10 bits) as initial values to LFSR;
PROCESS : If want to hide encipher message CALL SSEA;
        CALL C2BCA;
        i = 1000, j = 1;
        Repeat
            Read Cbyte(i);
            Read bit(j);
            insert bit(j) in LSB of Cbyte(i);
            Write Cbyte(i) in cover;
            using LFSR to generate Keyjmp; {  $0 \leq \text{Keyjmp} \leq 3$  }
            i=i + Keyjmp +1, j=j + 1;
        Until EOF (message);
OUTPUT  : Stego image;
END.

```

While the pseudo code of the LFSR extracting algorithm is:

```

NAME: LFSR Extracting Algorithm (LFSREA).
INPUT: Stego image,
        BK (10 bits) as initial values to LFSR;
PROCESS : i = 1000, j = 1;
        Repeat
            Read Sbyte(i);
            extract bit(j) from Sbyte(i);
            Write bit(j) in message;
            CALL LFSR to generate Keyjmp;
            i= i+ Keyjmp +1, j=j + 1;
        Until EOF (message);
        CALL B2CCA;
        If message encipher CALL SSDA;
OUTPUT  : Plain message;
END.

```

### **4-3 Steganalysis System Design**

As known , it's not easy to specify the hiding positions of the stego image to approach the hidden text, this requires different analytic processes. The proposed steganalysis system divided into some main stages. Every stage implementation is related to implementation success of the previous stage. The implementation of every stage is related to how much the available information is useful .In this thesis, the origin image availability and some information about the stego tools are only the available information, and there is no information about the hidden message, like message length or probable words. One stage may be implemented more than one style or method to guarantee the stage implementation correctly and precisely, that is because one style may not be sufficient to implementation successfully the specified stage.

The steganalysis system consists of three main stages, hidden message diagnosis, breaking and extracting. In this work, the disable process is not mentioned, because there is a fact which says that the disable process will make some one suspicious of the transmitter and receiver, this suspicion makes them believe that their communication channel is under watch. This suspicion make them change their stego style or method (or communication channel or perhaps even abort their activities), this implies more difficulties facing the steganalyst in diagnosis and breaking.

Figure (4-2) shows the structure of the proposed steganalysis system. This system starts with the hidden message existence diagnosis stage, then implements the message and stego tools breaking stage. The system will test the three hiding systems which are mentioned in section

(4-2). If this stage succeeds then it moves to the next stage which is the extracting stage.

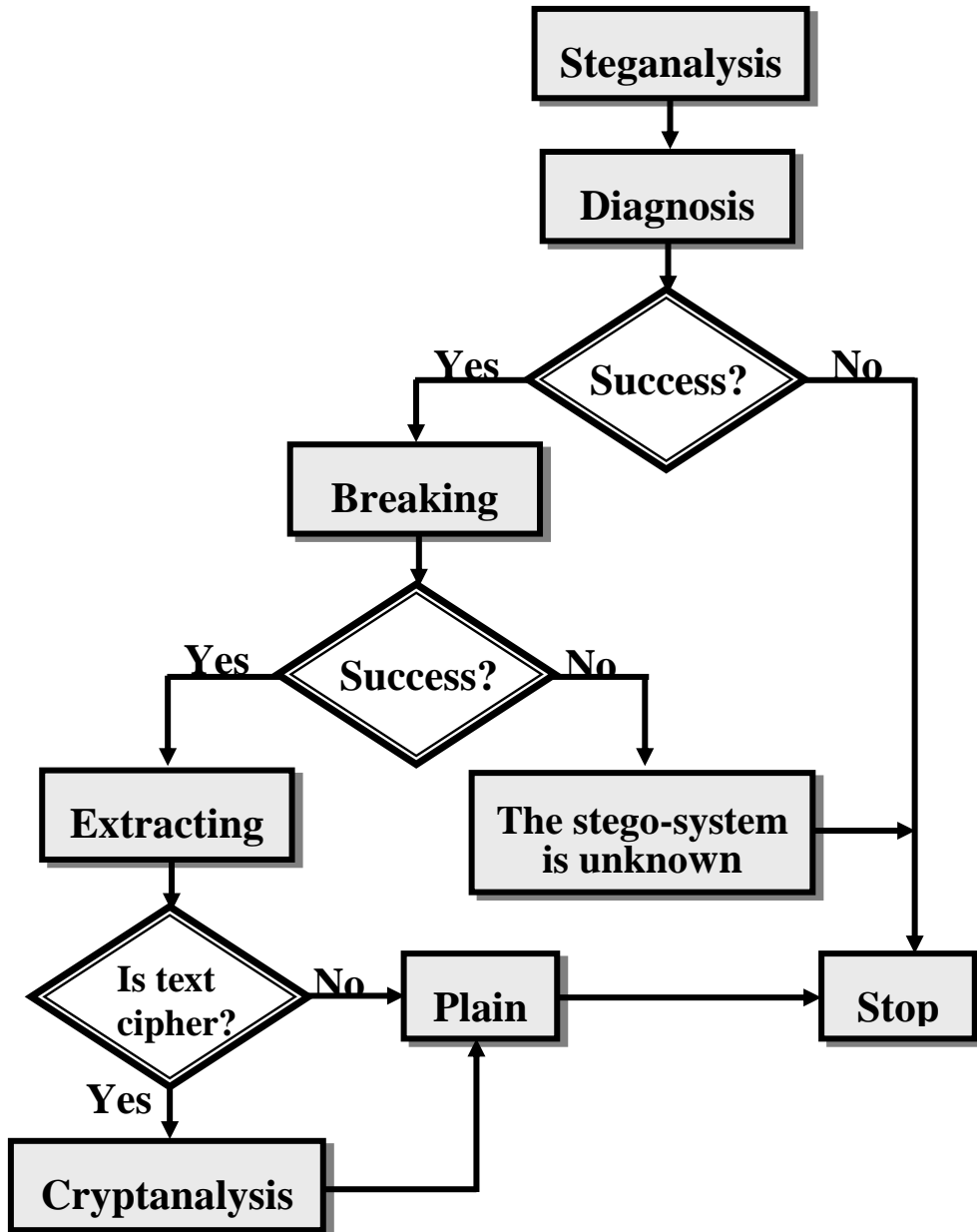


Figure (4-2) The structure of the proposed steganalysis system.

#### 4-3-1 Hidden Message Diagnosis Stage (HMDS)

The HMDS is important, since it's the first stage toward the breaking stage. It saves time, in searching for the hidden message or diagnoses the stego tools. The diagnosis stage may not succeed in specifying the existence of the hidden message, therefore, it must apply

more than one diagnosis style to gain a precise decision. Of course, the successful diagnosis stage implementation is related to the available information. In this manner, more than one diagnosis method is applied in case only the stego image is available.

#### 4-3-1-1 Diagnosis by Cover and Stego Images

In this method, the cover and the stego images are compared, that is done by using Mean Square Error (MSE) which is mentioned in chapter three.

To calculate the MSE, equation (3-2) can be used in bytes of the two images to determine whether the second image contains a hidden message, if  $MSE = 0$ , this means the second image does not contain any embedded message, else it does. As MSE is high, this is a function of size of the hidden message. The type of stego tool has no real effect on MSE value. Table (4-2) shows different cases of hiding and results of MSE.

Table (4-2) MSE results.

Stego tool	Message	MSE	
		1000 bits	2000 bits
Without hiding	No message	0	0
SSA	Plain	0.02045	0.041213
	Cipher	0.02467	0.041213
JSA	Plain	0.02049	0.041213
	Cipher	0.02049	0.049242
LFSRSA	Plain	0.02049	0.041213
	Cipher	0.02467	0.049242

The pseudo code of the calculating MSE Algorithm is:

```
NAME: MSE Algorithm (MSEA).  
INPUT: Cover image, Stego image;  
PROCESS : Calculate X, Calculate Y  
          i = 0..X-1, j = 0..Y-1, MSE = 0;  
          Repeat  
            Read Cbyte(i,j);  
            Read Sbyte(i,j);  
            MSE = MSE + Sqr(Cbyte(i,j)-Sbyte(i,j)) / (X*Y);  
          Until EOF (cover);  
OUTPUT : MSE result;  
END.
```

### 4-3-1-2 Diagnosis by Stego Image Only

Diagnosis by stego image only is more difficult than diagnosis by cover and stego, and sometimes, no final decision can be made as it could do in diagnosis by cover and stego. Therefore, three methods can be applied to help the steganalyst in getting precise decision to move to the next stage of image analysis or abort the steganalysis.

The three diagnosis methods are mentioned in chapter three, visual attack using filter, statistical attack using Laplace operator and Chi-square.

#### 4-3-1-2-1 Visual Attack Method

In this method, a filter can be used through calculating the LSB of image bytes then multiplying by scale value, if the image contains hidden message we expect some noise to appear in the filtered image. This method depends on showing the filtered image so it could be useless in some Bmp images.

The pseudo code of the visual attack algorithm is:

```

NAME: Visual Attack Algorithm (VAA).
INPUT: Stego image, Scale;
PROCESS   : Calculate X, Calculate Y;
            i = 0..X-1, j = 0..Y-1;
            Repeat
                Read Sbyte(i,j);
                LSB(i,j) = Sbyte(i,j) AND 1 ;
                P(i,j) = LSB(i,j)*Scale;
            Until EOF (stego);
OUTPUT    : Show new image (P(i,j));
END.

```

#### 4-3-1-2-2 Laplace Operator Method

Laplace operator is useful in checking if there is a deviation in neighborhood pixels, as usual the neighborhood pixels are approximate to each others.

If there is just one peak then the decision is that the tested image has no hidden message, else when there are more than one peak, the decision is, to find out whether the tested image contains a hidden message.

The pseudo code of the Laplace operator algorithm is:

```

NAME: Laplace Operator Algorithm (LOA).
INPUT: Stego image;
PROCESS   : Calculate X, Calculate Y;
            i = 1..X-2, j = 1..Y-2;
            Repeat
                Read P(i,j), P(i+1,j), P(i-1,j), P(i,j+1), P(i,j-1) from stego;
                 $\nabla^2 P(i,j) = P(i+1,j) + P(i-1,j) + P(i,j+1) + P(i,j-1) - 4 * P(i,j)$ ;
                FRQ[ $\nabla^2 P(i,j)$ ]+1;
            Until EOF(stego);
OUTPUT    : Histogram of (FRQ[ $\nabla^2 P(i,j)$ ],  $\nabla^2 P(i,j)$ );
END.

```

**4-3-1-2-3 Chi-Square Method**

This method is represented by calculating PoV, first, the color frequency must be calculated in every percentage in increasing form (1%, 2%,..., 100%), of course the occurrences of the colors and their frequencies will increase, that means increasing in the number of different categories (degree of freedom).

To calculate the Chi-square value with degree of freedom  $v-1$ , the hiding probability  $p$  can be calculated, if  $p \geq 0.5$  there is a good probability of hiding, in contrast, the image may contain no hidden message. We can use the results to graph a histogram to describe the probability of hiding, and may specify the probable length of the hidden message.

The pseudo code of the Chi-Square algorithm is:

```

NAME: Chi-Square Algorithm (CSA).
INPUT: Stego image;
PROCESS   : i = 0..X-1, j = 0..Y-1, k = 0, L=0, Per = (X*Y) Div 100;
            Repeat
                Read P(i,j) from stego;
                FREQ=FREQ[P(i,j)]+1;
                If L mod Per = 0 then;
                    K=k+1;
                    c = FREQ;
                     $y_n = c_{2n}, y_n^* = (c_{2n} + c_{2n-1}) / 2;$ 
                     $chi = (y_n - y_n^*)^2 / y_n^*;$ 
                     $P_k = chi \text{ Table}(chi_k, V_{k-1})$ 
                    Graph a point (K, Pk);
                endif
            Until EOF(stego);
OUTPUT    : Histogram of (K,P);
END.
```

### 4-3-2 Breaking stage (BS)

Breaking the steganography system means finding the stego tool, the hidden message and stego key.

From previous stage (diagnosis stage) a decision has been made to implement the breaking stage on image which contains a hidden message. In this stage, some information is available to help in this stage. This information may be treated as conditions to help in breaking. These conditions are:

1. The start hiding position is known (byte number 1000).
2. Hiding is done by using one of the three-steganography systems mentioned before.
3. The cover image may be available, in advanced work of this thesis, this condition is dropped.
4. The hidden text may be plain, if it is cipher must be enciphered by SSEA.

Stego image availability is not an additional helping condition in breaking, if it is not available that means no steganalysis be made.

Before we are involved in breaking details, it's important to show how we can calculate the Index of Coincidence value which is very useful in our work.

In an observed sample of  $n$  text letters, suppose there are  $\text{Freq}_i$  instances of the character  $i$ . We want to know the likelihood of picking  $i$  twice at random.

The index of coincidence, written IC, is a way to approximate variance from observed data.

$$\text{IC} = \sum_{i=a}^{i=z} \frac{\text{Freq}_i * (\text{Freq}_i - 1)}{n * (n - 1)}$$



The index of coincidence ranges from 0.0384, for some kinds of encipher system with a perfectly flat distribution, to 0.068, for a simple substitution from common English texts [15].

The pseudo code of calculating the index of coincidence algorithm is:

```

NAME: Index of Coincidence Algorithm (ICA).
INPUT: Message text;
      Sum = 0;
PROCESS   : For i = 1 to n                      { n = message length }
            Read character(i);
            FREQ[character(i)]+1;
        endfor
            For i = 'a' to 'z' sum = sum +FREQ[i]*(FREQ[i]-1);
            IC = sum / (n*(n-1));
OUTPUT    : IC;
END

```

#### 4-3-2-1 Breaking Using Cover and Stego Images

The breaking methods differ from each other because of the stego tools variety and the types of images. In this thesis, the cover image can be used to compare its data with stego data. The comparison process starts from the hiding position until they get similar to each other. Of course, the similarity is not one point, it's a string of bytes, the length of similarity string is related to the used stego tool. The stopping condition is if there is a string of similar LSB's with specified length.

Before we describe the details of breaking, we should show the pseudo code of comparison algorithm.

```

NAME      : Comparison Algorithm (CA).
PROCESS   : Read Cbyte(i);
            Read Sbyte(i);
            Hbit(i) = Sbyte(i) AND 1; { Hbit(i) = hidden bit(i) }
            Write Hbit(i) in text;
            if Sbyte(i) = Cbyte(i) then j=j+1 else j = 0;
END.

```

The breaking of the three steganography system using cover and stego images has been shown in the next three subsections.

#### 4-3-2-1-1 Breaking the SSA Using Cover Image

Breaking the SSA using cover image is considered the easiest method compared with other breaking methods. The process depends on comparing the LSB of the two images in sequential order. The LSB's of the stego image are the hidden message bits.

The pseudo code of the breaking SSA using cover image algorithm is:

```

NAME      : Breaking SSA using Cover Image Algorithm (BSSCIA).
INPUT: Cover image, Stego image
          Simb;                                { number of similar bytes }
PROCESS   : i = 1000, j = 0;
          Repeat
              CALL CA;
              i = i+1;
          Until j > Simb;
OUTPUT    : Hidden Text;
END.
```

#### 4-3-2-1-2 Breaking the JSA Using Cover Image

Breaking the JSA using cover image is considered harder than that of BSSCIA method. Since the length of jump is unknown, all jumps length possibility must be discussed. The process depends on comparing the LSB of the two images in jumping by 1,2,...,10 bytes. The LSB's of the stego image in true jumped bytes are the hidden message bits.

To approach the true jump, the IC value should be calculated first. If it is sure that the stego tool is JSA, then it is certain that one of the

jumping values has  $IC > 0.06$ , this IC represents the plain or simple substitution cipher. Since it starts from  $jmp = 1$ , then the first true jump gives the best IC.

The pseudo code of the breaking JSA using cover image algorithm is:

```

NAME      : Breaking JSA using Cover Image Algorithm (BJSCIA).
INPUT: Cover image, Stego image, Simb;
PROCESS   : jmp = 1;
            Repeat
                i = 1000, j = 0;
                Repeat
                    CALL CA;
                    CALL B2CCA;
                    jmp=i+jmp+1;
                Until j > Simb;
                jmp=jmp+1, CALL ICA;
            Until (j > 10) or (IC > 0.06);
OUTPUT    : Hidden Text;
END.

```

#### 4-3-2-1-3 Breaking the LFSRSA Using Cover Image

The LFSRSA includes random jumping so there is no fixed style to approach the hidden text. Therefore another method is adopted. The new method depends on discussing all possible of stego keys (BK). This is done because the length of the jumps is unknown and they are not fixed, so all initial values possibilities must be discussed. This process is called exhaustive search.

All possible initial values of LFSR are  $2^{10} = 1024$ , the process includes filling the LFSR with one initial value, then moves the LFSR to generate hiding positions of stego data and compare it with cover data to specify the end of the hidden message. In addition to that, we calculate

the IC value, if  $IC > 0.06$  that means we get the hidden message, this implies that the initial value is the correct one, else another possible value must be tried.

One or more false initial values may give good IC, those initial values are very few, therefore, we will show all possible plain text or try to break the cipher text, with good IC.

The pseudo code of breaking the LFSRSA using cover image algorithm is:

```

NAME      : Breaking LFSRSA using Cover Image Algorithm
            (BLFSRSCIA).
INPUT: Cover image, Stego image, Simb;
PROCESS   : P = 0;
            Repeat
                i = 1000, j = 0,    p+1, LFSR = new initial values;
                Repeat
                    using LFSR to generate Keyjmp;
                    CALL CA;
                    CALL B2CCA;
                    i = i + Keyjmp + 1;
                Until j > Simb;
                CALL ICA;
                If IC > 0.06 then show hidden text;
            Until (P > 1024);
OUTPUT    : Hidden Text;
END.

```

#### 4-3-2-2 Breaking Using Stego Image Only

The availability of stego image is not a helping condition in breaking process. The cover availability is very useful in this work to break the three steganography cases mentioned before. In this part of this thesis, the IC value is a good alternative to cover availability.

The IC value is efficient not only in getting precise diagnosis, but it gives a probable length of the embedded message. The process idea is, as we proceed in searching for the embedded message, the IC value will increase or at least be fixed in specified value, unless we get the message end, the IC value will start to decrease because of the noise data added to real message. The decreasing of IC value is the condition to stop the process.

Before we describe the details of breaking, we should show the general formula of pseudo code of breaking stego image only algorithm.

NAME	: General Breaking Stego Image Only Algorithm (GBSIOA)
	( $\pi(i)$ ). { $\pi(i)$ = the jump value }
INPUT	: Stego image;
PROCESS	: $i=1000$ , $MAXIC = 0$ ;
	Repeat
	Read Sbyte(i);
	$Hbit(i) = Sbyte(i) \bmod 2$ ;
	Write Hbit(i) in text;
	$i = i + \pi(i) + 1$ ;
	CALL B2CCA;
	CALL ICA;
	If $MAXIC < IC$ then $MAXIC = IC$ ;
	Until $(IC - MAXIC \geq 0.005) \text{ and } (MAXIC \geq 0.065) \text{ or } (EOF(\text{stego}))$ ;
OUTPUT	: Hidden message;
END.	

The breaking of the three steganography system using stego image only is shown in the next three subsections.

#### 4-3-2-2-1 Breaking the SSA Using Stego Image Only

After the stego image has a certain hidden message, the Hbit(i) is extracted in sequential order from the bytes of the stego image. The IC value is calculated to reach the embedded message end.

The pseudo code of the breaking SSA using stego image only algorithm is:

```

NAME      : Breaking SSA using Stego Image Only Algorithm
            (BSSIOA).
INPUT: Stego image;
PROCESS   : CALL GBSIOA (0);
OUTPUT    : Hidden Text;
END.

```

#### 4-3-2-2-2 Breaking the JSA Using Stego Image Only

As shown before, in subsection (3-3-2-1-2), and previous subsection, all possible jumping will be tested starting from Keyjmp=1 to 10, and helping of IC value.

The pseudo code of the breaking JSA using stego image only algorithm is:

```

NAME      : Breaking JSA using Stego Image Only Algorithm
            (BJSSIOA).
INPUT: Stego image,  $\pi(i)=1$ ;
PROCESS   : Repeat
            CALL GBSIOA ( $\pi(i)$ );
             $\pi(i) = \pi(i)+1$ ;
            Until  $\pi(i) > 10$ ;
OUTPUT    : Hidden Text;
END.

```

#### 4-3-2-2-3 Breaking the LFSRSA Using Stego Image Only

This method depends on discussing all possible stego keys (BK), which represents the initial values of LFSR, as done in subsection (4-3-2-1-2).

The IC value is still the decision point in finding the correct initial values to get the hidden message.

The pseudo code of breaking the LFSRSA using stego image only algorithm is:

```

NAME      : Breaking LFSRSA using Stego Image Only Algorithm
            (BLFSRSSIOA).
INPUT: Stego image;
PROCESS   : P = 0;
            Repeat
                P = P+1, LFSR = New initial values;
                CALL GBSIOA ( $\pi(i)$ );    { include using LFSR
                                            to generate Keyjmp
                                             $\pi(i)=\text{Keyjmp}$  }
            Until (P > 1024);
OUTPUT    : Hidden Text;
END.

```

### 4-3-3 Extracting Stage (ES)

As we mentioned before, the ES implementation depends on BS success. From the previous stage we get the hidden message. The first function of this stage includes showing the extracted message, since the real extracting process is done in previous stage. The second function is breaking the extracted message if it's a cipher message.

#### 4-3-3-1 Extracting Plain or Cipher Message

The proposed steganalysis system shows the extracted message disregard the hidden message is plain or cipher.

If the message is plain, the steganalyst must look at it carefully, to see if it includes any rubbish data especially, in the end of the message,

this is done because of BS. So the steganalyst has to clean the extracted message from this rubbish.

### 4-3-3-2 Breaking the Encipher Text Message

A cryptanalyst's chore is to break an encryption; this means that the cryptanalyst will attempt to deduce the meaning of a cipher text message, or to determine a decrypting algorithm that matches an encrypting algorithm.

To complete the discussion of previous subsection, the message may be enciphered. The steganalyst can't understand the extracted message, so he must find a way to make the message be clear.

In order to make the extracted message readable, the steganalyst must try to decipher the enciphered message, since he knows no information about the encipher key, all he knows that the message is enciphered by SSEA.

Although the SSEA breaking is not in our interest , the system has been developed to break the SSEA. The breaking process includes calculating the frequencies of the encipher message letters.

Now we have to describe Frequency Distributions, in English some letters are used more frequently than others. The letters E, T, and A occur far more frequently than J, Q, and Z, for example. The text being analyzed also affects the distribution.

Table (4-3) shows the counts and relative frequencies of letters in some common plain texts. These frequencies are quite close to published counts from other sources. (The shaded letters in the table (4-3) represent the first 10 high frequencies letters).



Table (4-3) Letter frequency distributions in English language [26].

Letter	English Language		
	Count	Percent	Order
A	3312	7.49	3
B	573	1.29	20
C	1568	3.45	12
D	1602	3.62	10
E	6192	14.00	1
F	966	2.18	16
G	769	1.74	17
H	1869	4.22	9
I	2943	6.65	7
J	119	0.27	24
K	206	0.47	22
L	1579	3.57	11
M	1500	3.39	13
N	2982	6.74	6
O	3261	7.37	4
P	1074	2.43	15
Q	116	0.26	25
R	2716	6.14	8
S	3072	6.95	5
T	4358	9.85	2
U	1329	3.00	14
V	512	1.16	21
W	748	1.69	18
X	123	0.28	23
Y	727	1.64	19
Z	16	0.04	26

The ratios mentioned in table (4-3) are related to the text length and the text subject (or text words). The breaking includes substituting the most high frequency letter in cipher text by ‘E’ and the next by ‘T’ and so on. After finishing the breaking process the deciphered text is shown to ensure that the substitution process is really correct [20].

## **4-4 System Requirements**

The software packages listed below are needed to build the proposed system. These packages are:

1. Microsoft Windows in all new versions Edition as operating system.
2. Delphi version 5.0 ,6.0,7.0 as a programming language.

## **4-5 System Implementation**

In order to show the steganalysis system implementation and how it works, we first show the steganography system how it's implemented.

### **4-5-1 Steganography System Implementation**

The steganography system consists of two main subsystems, these are Hiding and Extracting subsystems. These appear in a radiogroup menu after executing the system program icon. The Block diagram of this system is shown in figure (3-3).

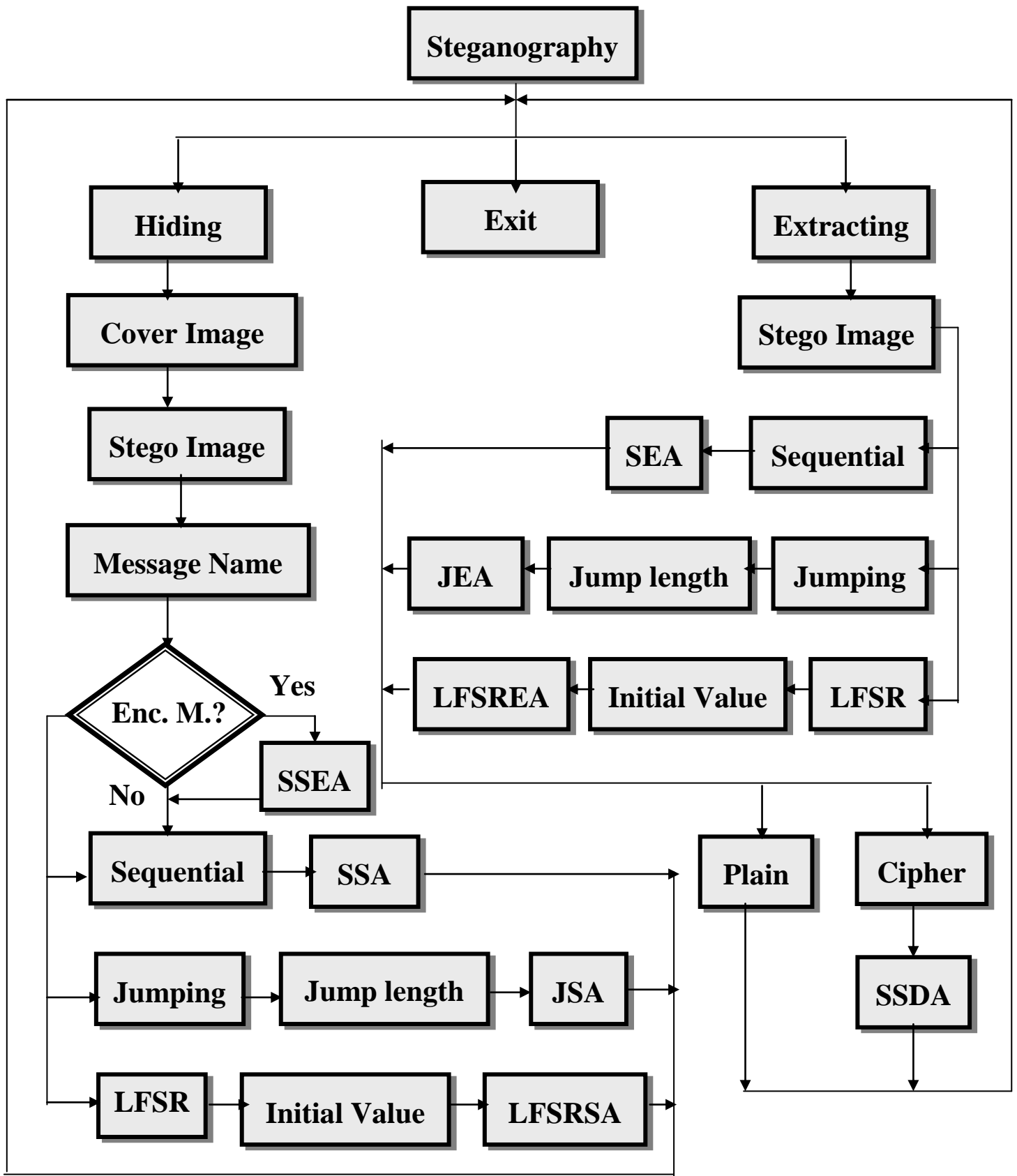


Figure (4-3) Block diagram of steganography system.

#### **4-5-1-1 Hiding RadioGroup**

After choosing the Hiding radiogroup, the system asks for the cover image, new name for stego image and the name of the message to be embed. Then, a submenu appears consisting of three Hiding algorithms. These algorithms are Sequential, Jumping and LFSR. Of course, the user enters some information related to the chosen algorithm and the message. The hiding process starts directly after finishing the choices, the system will inform the user when the process ends.

#### **4-5-1-2 Extracting RadioGroup**

After choosing the Extracting radiogroup, the system asks for the name of stego image. Then, a submenu appears consisting of three Hiding algorithms as mentioned in subsection (4-5-1-2). The extracting process starts directly after finishing the choices, the system will show the embedded message after the process is finished.

#### **4-5-2 Steganalysis System Implementation**

When the program icon is executed, the main window will appear, which includes the main menu which consists of five options. The block diagram of this system is shown in figure (4-4).

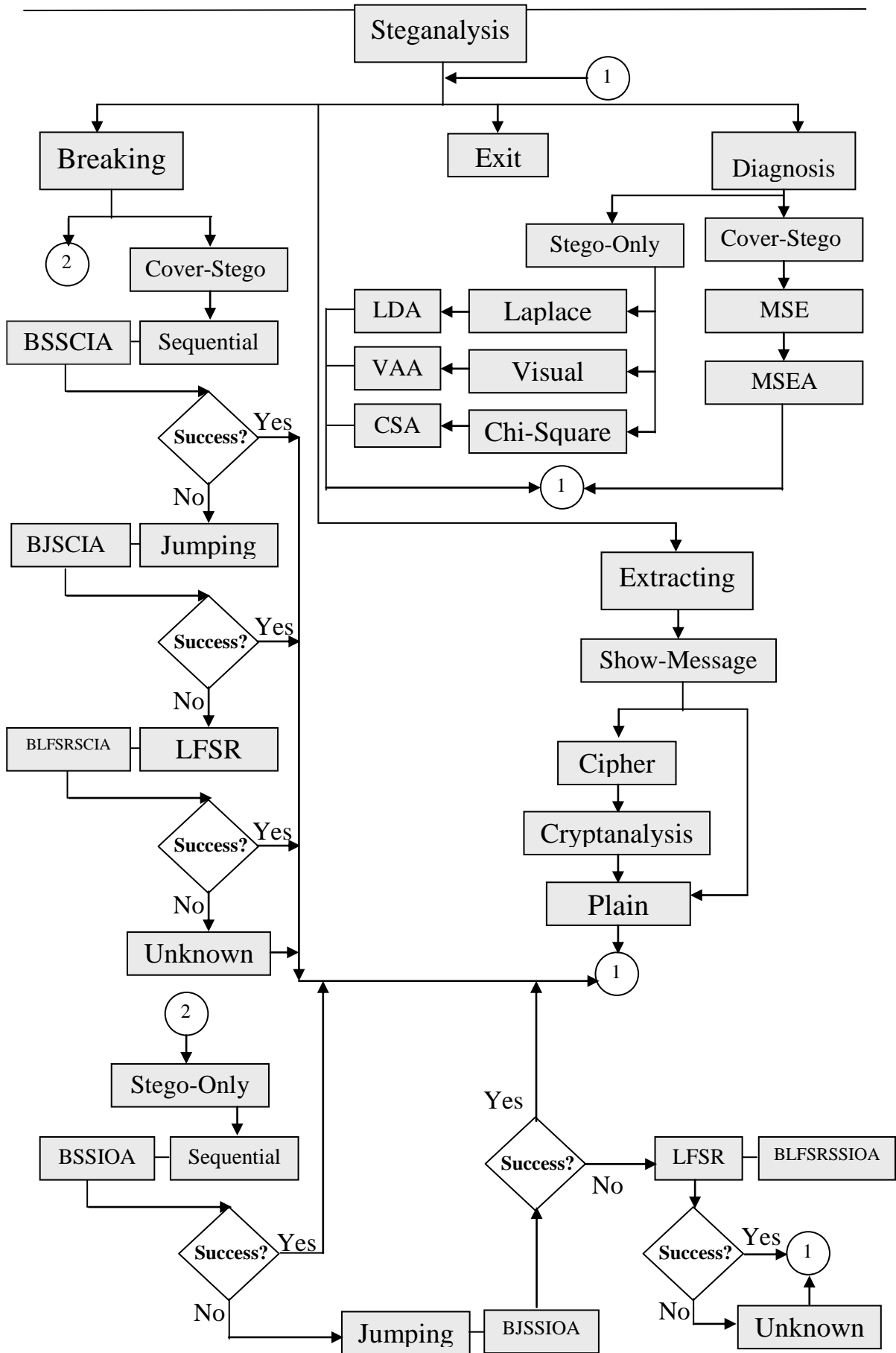


Figure (4-4) Block diagram of steganalysis system

### **4-5-2-1 Images Files Submenu**

When this submenu is chosen, two choices appear, these choices are cover image file and stego image file. Of course the user can select the two choices or only the second one.

### **4-5-2-2 Diagnosis Submenu**

This submenu shows two choices, first one is selected when cover and stego images are available, this selection applies MSE test. The second stego is only available, if it is chosen, three choices appear, which are Visual, Laplace and Chi Square tests.

### **4-5-2-3 Breaking Submenu**

In this Choice, the three-stego tools are tested in the order : sequential, jumping and LFSR algorithms. When first one fails, the system directly tests the second one and so on. If the last one fails this means that the stego tools is unknown.

### **4-5-2-4 Extracting Submenu**

If this submenu is selected, the system shows the extracted message. If the message is enciphered then the user must use the cryptanalysis process to start breaking the enciphered message in order to get the plaintext message.

## **4-6 Experimental Examples**

In this section, three images are introduced, to hide one message (plain and cipher) use three steganography systems. The diagnosis stage results are shown and at last, the breaking stage results.

- 1 - The image before and after hiding process using SSA.
- 2 - The image before and after hiding process using JSA.
- 3 - The image before and after hiding process using LFSRSA.

Table (4-4) MSE test

Stego tools	Successful percentage
Sequential	100%
Jump	100%
LFSR	100%

#### **4-7 Steganalysis System efficiency Test**

We use (14) images to test the steganalysis system efficiency, the tests done as follows:

1. Test hiding for three steganography tools.
2. Test the diagnosis stage for hidden cipher text only.
3. We use the MSE test for cover and stego images.
4. we use the visual, Laplace and chi-square tests for stego image only.

The tests result are describe in table (4-4) and table (4-5). Table (4-4) shows the successful percentage for visual, Laplace and chi-square tests.

Table (4-5) Visual, Laplace and chi-square tests

Stego tools	Successful percentage		
	Visual	Laplace	Chi-square
Sequential	100%	78%	86%
Jump	100%	71%	65%
LFSR	93%	71%	57%



## Chapter Five

### Conclusions and Future Works

#### **5-1 Introduction**

The research contribution, based on investigating steganography in BMP images, includes the steganalysis of LSB data-hiding techniques, and attacks against hidden information. The proposed Steganalysis System suggests attacking and analyzing hidden information in LSB BMP images by processes of three stages: Diagnosis, Breaking and Extracting.

#### **5-2 Conclusions**

In this research, the proposed Steganalysis System tries to extract the hidden information from suspected image by concluding the steganographic techniques that are used in embedding process, the manner of choosing the bytes that are used in embedding process, or the simple substitution encipher system used . The proposed system has the ability to analyze the suspected images whether the original image is available or not. This makes it able to extract the information hidden from suspected image.

The statistical tests, visual tests and histograms are used to decide if suspected image has information hidden or not. A new technique is suggested as a tool to detect if suspected image has secret information. Steganalysis System is built to be more useful for attacking images, the following are some points concluded from this study:

1. The steganography system designer must follow some countermeasures concluded from steganalysis tools to protect his steganography systems.
2. Not every stego image can be detected by using steganalysis attacks.

3. Not every steganalysis attack can give a positive result gotten from stego image, that's obvious from experimental examples shown in Appendix-C.
4. The MSE value could be used as a function of the length of the hidden text.
5. From the results shown in table (4-4) and table (4-5), the visual and Laplace test more efficient than chi-square test, and the sequential and jump stego tools are easier in diagnosis than LFSR algorithm.

### **5-3 Suggestion for Future works**

This thesis recommends the following points to enhance the work of the steganalysis system:

1. In cryptanalysis of SSEA, we suggest using more efficient method, like Genetic Algorithm to break the encipher message.
2. In the proposed steganalysis systems, we suppose that the start hiding byte is known, it is not hard to drop this condition too.
3. The steganalysis system can be developed to add more stego detection tool.
4. In the proposed steganalysis system, a modification (disabling) stage can be added when the breaking stage succeeds, the new stage is specialized in modifying the extracted message then embedding the new message in the same image., then sending the new stego image to the receiver.
5. Increase the number of steganography algorithms that can be analyzed to detect the hidden data in image.
6. Develop the work of the system to deal with text and sound information hiding.
7. Improve the ability of the steganalysis system by using Neural Network and Genetic Algorithm.

# References

## **References**

- [1] Ala Eldin H. Qasim, “**Text Hiding in Image Border**”, Military College of Engineering, M. Sc. Thesis 2002.
- [2] Al-hamami. Mohammed, “**Information Hiding Attack in Image**”, M. Sc. thesis introduced to Ministry of Higher Education & Scientific Research Iraqi Commission for Computer & Informatics, Informatics Institute for Postgraduate Studies, 2002.
- [3] Andersen, R.J. and Petitcolas, F.A.P., “**On the Limits of Steganography**”, IEEE Journal of Selected Areas in Communications, Special Issue on Copyright and Privacy Protection, vol. 16, no. 4, pp. 474-481, 1998.
- [4] Bourke, P. “**BMP Image Format**”, Englewood cliffs: Prentice-hall , July 1998.
- [5] Brown, W., Shepherd, B. J. “**Graphic File Format: Reference and Guide**”, Green Wich, CT: Manning Publications, 1995.
- [6] Buonaiuti M. F., “**The Proper Law of a Tort and the Internet**”, Amicus Curiae, Journal of the Society for Advanced Legal Studies, CCh Edition Ltd, Vol. 13, p.p. 28, June 1999, London, England.
- [7] Cox, I. J., Kilian, J., Leighton, T. and Shamoon, T., “**A Secure Robust Watermark for Multimedia**”, In Proceedings of the First International Workshop on Information Hiding, Cambridge, UK LNCS, 1174, Springer, pp. 185-206, 1996.
- [8] Curran, K. and Bailly, K., “**An Evaluation of Image Based Steganography Methods**”, Internet Technologies Research Group, University of Ulster, 2001.

- [9] Dreier, T., “**The Council Directive of 14 May 1991 on the legal protection of computer programs**”, European Intellectual property Review, vol. 9, 1991, pp.319.
- [10] Fabien A. P. Petitcolas, Ross J. Andeson, And Markus G. Kuhn, “**Information Hiding-A Survey**”, URL, 1999
- [11] Farid, H., “**Detecting Steganographic Message in Digital Images**”, Report TR2001-412, Dartmouth College, Hanover, NH, 2001.
- [12] Fridrich J. and Miroslav G. and Hoge D., “**Steganalysis of JPEG Images: Breaking the F5 Algorithm**”, SUNY Binghamton, Dept. of Computer Science, Binghamton, NY 13902-6000, USA, 2001.
- [13] Fridrich, J., Goljan, M., and Du, R., “**Detecting LSB Steganography in Color and Gray-Scale Images**”, Magazine of IEEE Multimedia, Special Issue on Security, October-November issue, 2001, pp. 22–28.
- [14] Fridrich, J., Goljan, M., and Du, R., “**Reliable Detection of LSB Steganography in Grayscale and Color Images**”, Proc. ACM, Special Session on Multimedia Security and Watermarking, Ottawa, Canada, October 5, 2001, pp. 27–30.
- [15] Friedman W.F., “**The Index of Coincidence and its Application in Cryptography**”, Riverbank Publication No.22, 1987.
- [16] Hansmann, F., “**Steganalysis: Scanning the Web**”, Steganos Website, URL, 2001, <http://www.demcom.deutsch/index.htm>.  
<http://www.cl.cam.ac.uk/~fapp2/publications/IEEE99-infohiding.pdf>.  
<http://www.know.comp.kyutech.ac.jp/STEGO2/Papers/program>.

- [17] Johnson, N. F., and Jajodia, S. “**Steganalysis of Images Created Using Current Steganography Software**”, Proceedings of Information Hiding Second International Workshop, Portland, Oregon, USA, April 1998.
- [18] Johnson, N. F., Duric, Z. and Jajodia, S., “**Information Hiding: Steganography and Watermarking-Attacks and Countermeasures**”. Kluwer Academic Publishers, Boston Dodrecht London, 2000.
- [19] Jonathan, W., “**Steganography - Messages Hidden in Bits**”, Multimedia Systems Coursework, Department of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK, 2000.
- [20] Kahn D., “**The Codebreakers: The story of Secret Writing**”, 2<sup>nd</sup> edition, New York: The MacMillan company, 1996.
- [21] Katzenbeisser, S. and Fabien A. P. Petitcolas, “**Information Hiding Techniques for Steganography and Digital Watermarking**” Artech House, Boston, London, 2000.
- [22] Lala Zareh Avedissian, “**Image in Image Steganography System**”. PH. D. Thesis, University of Technology, 2000.
- [23] Menezes, A. J., Van Orschot, P. C. and Vanstone, S. A., “**Handbook of Applied Cryptology**”, CRC Press, 1997.
- [24] Memon Nasir D., Khalid Sayood, “**Lossless Compression of Color Image in the RGB Domain**”, Computer Science and Mathematics, Arkansas State University, 2001.
- [25] Pal, S. K., Saxena, P. K. and Mutto, S. K., “**A Systematic Approach to Steganographic of Images**”, URL, 2002

- [26] Pfleeger C.P., “**Security in Computing**”, The University of Tennessee, 1989.
- [27] Provos, N. and Honeyman, P., “**Detecting Steganographic Content on the Internet**”, CITI Technical Report 01-11, August 2001.
- [28] Provos, N., “**Defending Against Statistical Steganalysis**”, Center for Information Technology Integration, University of Michigan, 2001.
- [29] Queirolo, F. “**Steganography in Image Final Communications Report**”, R 2001-412, Dartmouth college, Hanover, NH, 2001.
- [30] Richer P., “**Steganalysis: Detecting Hidden Information with Computer Forensic Analysis**”, SANS/GIAC Practical Assignment for GSEC Certification, Version 1.4b, 2002.
- [31] Rifat Z. K. “**Statistical Approach for Steganalysis**” M.Sc. thesis, Applied Sciences of University of Technology 2003.
- [32] Salima B. Abdulah and Hilal M. Yousif, “**Arabic Text Information Hiding**”, Iraqi Commission for Computers and Informatics Institute for Postgraduate Studies, 2003.
- [33] Salomon D., “**Data Compression**”, Department of Computers Science, California State University, 1995.
- [34] Schneier, B. “**Applied Cryptography (Protocols, Algorithms, and Source code in c.)**”, John Wiley & Sons Inc, Second Edition 1997.
- [35] Sellars, D., “**An Introduction to Steganography**”, <http://www.cs.uct.ac.za/courses/CS400W/NIS/papers99/dsellars/stego.ps.gz>, 1999.

- [36] Umbaugh, S. E., “**Computer Vision and Image Processing: A Practical Approach Using CVIP Tools**”, Prentice-Hall PTR, 1998.
- [37] Westfeld A., and Pfitzmann A. “**Attacks on Steganographic systems**” In proceedings of Information Hiding-Third International Workshop, vol.1768, Springer-Verlag, Berlin, 2000, pp. 61-75.
- [38] Zhao, J, and Koch, E., “**Towards Robust and Hidden Image Copyright Labeling**”, Proceeding IEEE Workshop on Non-Linear Signal and Image Processing, Neos MarMaras, Greece pp.452-455, 1995.



## المستخلص

إن الباحثين والمستخدمين لأنظمة الإخفاء أصبحوا على دراية أكثر بان كشف المعلومات المخفية أصبح أكثر تحدياً. فإن لم يتمكن شخص ما من كشف الإخفاء، هذا يعني أن لعبة استخلاص النص المخفي قد انتهت. وبالحقيقة إن تضمين المعلومات في LSB للصورة لا يعطي حماية عالية للمعلومات المخفية في حالة كون نظام الإخفاء المستخدم يعطي مؤشرات على النص المخفي. ولهذا السبب تم اختيار بحث كشف الإخفاء في هذا النوع من تقنيات الإخفاء لتحذير المستخدمين لهذه الأنظمة.

في هذا البحث، تم اقتراح طرق جديدة وتطوير طرق تقليدية لكشف الرسالة المخفية. عملية التطوير للطرق التقليدية تم من خلال إضافة تقنيات جديدة لزيادة إمكانياتها في الكشف.

هدف البحث ليس كشف الإخفاء فحسب، وإنما كسره. إن تحليل الإخفاء بشكل عام يعني كشف الإخفاء، استخلاص النص المخفي ومن ثم عملية تشويه الرسالة السرية. نظام التحليل المقترح يتضمن أدوات ومراحل تحليل جديدة وهي ثلاث: التشخيص، الكسر والاستخلاص.

في عملية التشخيص، تم اقتراح التحليلات الإحصائية و الاختبار البصري كأدوات للكشف. هذه الاختبارات هي: معدل مربع الخطأ MSE في حالة توفر الصورة الأصلية وصورة الإخفاء. وتم استخدام اختبارات أخرى مثل مؤثر لابلاس، واختبار مربع كاي والاختبار البصري في حالة توفر صورة الإخفاء فقط.

عملية الكسر تتضمن كسر ثلاث نظم إخفاء مقترحة في حالة توفر الصورة الأصلية هي: نظام الإخفاء المتسلسل، والقافز وخوارزمية تعتمد على المسجل الزاحف الخطي. تم تطوير عملية الكسر لتحسين أداء النظام من خلال كسر أنظمة الإخفاء المقترحة بتوفر صورة الإخفاء فقط.

العملية الأخيرة، هي عملية استخلاص النص المخفي، وتتضمن إظهار النص المخفي. تم تطوير النظام لكسر النص المشفر المخفي بالنظام التعويضي البسيط وإظهار النص الواضح.



جامعة سانت كلمنتس

# انظمة الاخفاء الكفوّة ذات السرية العالية

رسالة مقدمة الى جامعة سانت كلمنتس وهي جزء من متطلبات  
نيل شهادة دكتوراه فلسفة في علوم الحاسبات

رسالة تقدم بها الباحث

**نعيم ياسر سلمان الخفاجي**

بإشراف

**أ.م.د. عماد عباس كوفي**

بغداد 2014

# **High Security and Efficient Information Hiding System**

## **1. Introduction**

Stenography is the art of hiding the very presence of communication by embedding secret message into innocuous looking cover document, such as digital image, videos, sound files, and other computer files that contain perceptually irrelevant or redundant information as covers or carriers to hide secret messages.

In this thesis, a new embedding technique in 24 or 8 bit colors digital Bmp images, introduced. The idea is using an arbitrarily non least bit of the chose byte to be condition of embedding bit. We called this technique a Controller Bit Technique (CBT).

In the current research a design and implementation for images steganography system based on LSB mechanism was presented and discussed. Different image files stored by using bitmap format were utilized. Some auxiliary processes were suggested and investigated in order to recover some weak aspect inherent with the pure implementation of stego-systems. Besides, the suggested system using crypto-hiding pseudo random key generator. This key generator works for the two purposes to investigate the encryption and embedding processes.

The suggested CBT stego-system can be implemented and tested using visual, Laplace and chi-square tests, the new method based on the idea of increasing the hoping rate due to the HVS-poor sensitivity.

## 2. Cryptography

**Cryptography** is the study of principles and techniques by which information can be concealed in ciphertexts and later revealed by legitimate users employing the secret key, but in which it is either impossible or computationally infeasible for an unauthorized person to do so.

In **stream ciphers**, the message units are bits, and the key is usually produced by a **random bit generator**. The plaintext is encrypted on a bit-by-bit basis. The key is fed into random bit generator to create a long sequence of binary signals. This “key-stream” is then mixed with plaintext, usually by a bit wise XOR to produce the ciphertext stream, using the same random bit generator and seed.

## 3. Basic Efficiency Criteria (BEC) of Key Generators

A **Pseudo Random Bit Generator (PRBG)** is a deterministic algorithm which, given a truly random binary sequence. The input to the PRBG is called the **seed**, while the output of the PRBG is called a **pseudorandom bit sequence**.

As known before, any stream cipher key generator consists of two basic units; they are sequence(s) of bit stream and Combining Function (CF) for the Key Generator (KG). Any weakness in any one of these units means clear weakness in output key generator sequence.

In this section, we will introduce the sequence efficiency in order to use the sequence as encryption key. The basic criteria of key generator efficiency can be defined as the ability of key generator and its sequence to withstand the mathematical analysis which the cryptanalyst can be applied on them.

- Randomness Criterion
- Periodicity Criterion
- Linear Complexity Criterion
- Correlation Immunity Criterion

#### **4. Steganography**

Steganography, from the Greek, means covered, or secret writing, and is a long-practiced form of hiding information. Although related to cryptography, they are not the same. Steganography's intent is to hide the existence of the message, while cryptography scrambles a message so that it cannot be understood.

The most common approaches to information hiding in images are:

1. Least significant bit (LSB) insertion.
2. Masking and filtering techniques.
3. Algorithms and transformations.

#### **5. Classification of Steganography Attacks & Methods**

There are three main types of attacks, passive, active and malicious attack, and there are two main steganalytic methods they are: visual and statistical analytic methods.

#### **6. Design High Efficient of Stego-System**

The aim of this project is applied an information hiding technique using LSB in digital BMP files hybrid with a new suggested technique. The embedding and extracting algorithms supported by encryption algorithm based on a key generator for the encryption and embedding purposes. This

project describes the proposed system design and implementation with experimental examples.

## **7. Design of Crypto-Hiding Key Generator Algorithm**

In this section we will introduce the idea of construct single key generator with two outputs or purposes; first it can be used as crypto key generator to encrypt (decrypt) the message want to be hiding (extract). The second, and in the same time, a hiding key generator to specify the byte from the stego-image to be hide in.

The proposed key generator which it's the heart of the steganography algorithm will be introduced. The proposed key generator consists of the following main components:

1. Main LFSR's System.
2. Combining Function (CF): we suggest to use non-linear function,  $F(x_1, x_2, x_3, x_4)$ .
3. Balance single LFSR.

## **8. Design of Control Bit Stego-System**

In this manner we suggest to design proposed steganography system. First, a new technique introduced by developing the LSB insertion technique, second, we describe how the embedding and extracting for one bit from the ciphertext. Lastly, we will show the extension process of the data embedding in whole the image.

1. Control Bit Technique
2. Embedding and Extracting of Cipher Bit
3. Extending the Embedding Process

## 9. Detection Tests of RCBS System

We have to test the stego-image resulted from the RCBS System using the following stages:

1. Applying the Detection Tests.
2. Detection Tests Results Analysis.

## References

- [1]. Johnson, N.F., "***Steganography***", WWW: <http://www.jjtc.com>, George Mason University, 2003.
- [2]. Alwan, R., Kadhim, F., and Al-Taani, A., "***Data Embedding Based on Better Use of Bits in Image Pixels***". International Journal of Signal Processing. [Online]. (2005) Available: <http://www.enformatika.org/ijsip/v2/v2-2-15.pdf>.
- [3]. Wu, N., and Hwang, M., "***Data Hiding: Current Status and Key Issues***". International Journal of Network Security. [Online]. (2007, Jan.) Available: <http://www.ijns.nchu.edu.tw/ijns-2007-v4-n1-p1-9.pdf>.
- [4]. Provos, N. and Honeyman, P., "***Detecting Steganographic Content on the Internet***", CITI Technical Report 01-11, August 2001.
- [5]. Fridrich, J., Goljan, M., and Du, R., "***Reliable Detection of LSB Steganography in Grayscale and Color Images***", Proc. ACM, Special Session on Multimedia Security and Watermarking, Ottawa, Canada, October 5, 2001, pp. 27–30.
- [6]. Schneier, B., "***Applied Cryptography (Protocols, Algorithms, and Source code in C.)***" Second Edition 1997", John Wiley & Sons Inc.
- [7]. Golomb, S.W., "***Shift Register Sequences***" San Francisco: Holden Day 1967, (Reprinted by Aegean Park Press in 1982).
- [8]. Ahmed, G. A. A., "***Steganalysis System for LSB Image Steganography***", M. Sc. thesis introduced to Iraqi Commission for Computer & Informatics, Informatics Institute for Postgraduate Studies, 2005.

المشرف  
أ.م. د. عماد عباس كوفي

طالب الدكتوراه  
نعيم ياسر سلمان