SUDAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

COLLEGE OF GRADUATE STUDIES

COLLEGE OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Enhancement of Image Steganography

Using Compression Techniques

تحسين إخفاء الصورة بأستخدام تقنيات الضغط

A Thesis submitted in partial fulfillment of the requirements for

the Degree of Master of Information Technology

Prepared by:

Badr Khalid GasmElkhalig

Supervised by:

DR.Talaat Mohielddin Wahby

JULY 2017

آيــــــــــــــــه

بسم الله الرحمن الرحيم

اللّٰهُ لَآ اِلٰهَ اِلَّا هُوَ الْحَیُّ الْقَیُّوْمُ

لَا تَاْخُذُهٗ سِنَةٌ وَّلَا نَوْمٌ لَهٗ مَا فِی السَّمٰوٰتِ وَمَا فِی الْاَرْضِ مَنْ ذَا الَّذِیْ یَشْفَعُ عِنْدَهٗٓ
اِلَّا بِاِذْنِهٖ یَعْلَمُ مَا بَیْنَ اَیْدِیْهِمْ وَمَا خَلْفَهُمْ وَلَا یُحِیْطُوْنَ بِشَیْءٍ مِّنْ عِلْمِهٖٓ اِلَّا بِمَا شَآءَ
وَسِعَ كُرْسِیُّهُ السَّمٰوٰتِ وَالْاَرْضَ وَلَا یَئُوْدُهٗ حِفْظُهُمَا وَهُوَ الْعَلِیُّ الْعَظِیْمُ

الحمـــــــــــــــــــــــد

الحمد لله اللهم ربنا لك الحمد بما خلقتنا ورزقتنا وهديتنا وعلمتنا وأنقذتنا وفرجت عنا ،لك الحمد بالإيمان ولك الحمد بالإسلام ولك الحمد بالقرآن ولك الحمد بالأهل والمال والمعافاة ، اللهم لك الحمد بكل نعمة أنعمت بها علينا في قديم أو حديث أو سر أو علانية أو خاصة أو عامة أو حي أو ميت أو شاهد أو غائب. نحمد الله تبارك وتعالى ان تفضل علينا بأن زودنا بأدوات العلم من السمع والبصر والفؤاد فعلمنا مالم نكن نعلم وزادنا من العلم بسطة بفضله مما أعاننا على إخراج هذا البحث ، لك الحمد حتى ترضى ولك الحمد إذا رضيت ولك الحمد بعد الرضى.

وصلي اللهم وسلم وبارك على سيدنا محمد وعلي آله و صحبه وسلم تسليما كثيرا.

DEDICATION

I DEDICATE THIS RESEARCH FOR

MY FAMILY

MY SUPERVISOR

MY SIBLINGS

MY FRIENDS

AND TO ALL THE PEOPLE WHO HELPED ME BRING THIS PROJECT TO LIFE.

AKNOWLEDAGEMENT

I WOULD LIKE TO THANK MY FAMILY FOR THE GREAT

SUPPORT THAT YOU GAVE ME.

“A TEACHERS AFFECTS ETERNITY; THEY CAN NEVER TELL

WHERE HIS INFLUENCE STOPS.”

THANKS TO ALL MY TEACHERS ESPECIALLY MY SUPERVISOR

” DR.TALAAT MOHIELDDIN WAHBY”

FOR TEACHING ME HOW TO BE ACCURATE IN ALL THE THINGS

THAT I DO.

THANK TO ALL MY FRIENDS FOR THE UNCONDITIONAL

SUPPORT ALONG THE WAY.

TABLES OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

In a world of digital technology, maintaining the security of the secret data has become a great challenge. One way to achieve this is to encrypt the message before it is sent. But encryption draws the attention of third parties, which may cause the third party to seek to break the encryption and to detect the original message. Another way is steganography, steganography is the art and science of writing hidden Messages in such a way that no one, apart from the sender and intended recipient, Suspects the existence of the message.

In this research apply Multilevel Steganography for image steganography was presented. MLS consists of at least two steganography methods utilized respectively. Two-levels of steganography have been applied; the first level is called (the upper-level), and it has been applied using enhanced LSB (secure LSB-L1) image steganography, the secret data in this level are English text, and the cover is Bitmap image, the output is a stego_image called (intermediate image).

And the second level is called (the lower-level); it has been applied using another enhance LSB (secure LSB-L2) based image steganography. In this level, another Bitmap (BMP) image has been used as a cover image and embeds (the BMP image output from level one) as a secure data and generates the new BMP image as stego image.

A lossless data compression technique using Huffman, LZW algorithm and Winrar Application between the First and Second levels of steganography are applied. The improved embedding capacity of the image possible due to preprocessing the stego-images from level one in which lossless data compression techniques are applied

After completing the proposed method implementation, many experiments have been conducted. Different sizes of secret messages and different sizes of cover images have been experimenting. Finally, comparative analysis of compression has been done on parameters of Quality Image Measurement like MSE, PSNR to experiment results and presented good results for the proposed method.

المستخلص

في عالم التكنولجيا الرقمية أصبح الحفاظ علي أمن البيانات السرية تحدي كبير.احدي الطرق المستخدمة هي تشفير الرسالة قبل ارسالها ، ولكن التشفير قد يلفت انتباه طرف ثالث ،و هذا قد يتسبب في السعي الي اكتشاف الرسالة الاصلية.هنالك طريقة اخري هي إخفاء المعلومات .إخفاء المعلومات هو فن وعلم كتابة الرسائل المخفية، في مثل هذه الطريقة لا أحد عدا المرسل و المستقبل المعني ، يشك في وجود الرسالة.

في هذا البحث، تم تطبيق مفهوم جديد لإخفاء البيانات السرية،يسمي بعلم إخفاء الصور متعدد المستويات. الاخفاء متعدد المستويات يحتوي علي مستويين علي الاقل من طرق إخفاء البيانات، ويتم تطبيقهم علي التوالي. في هذا البحث تم استخدام مستويان . المستوي الاول يسمي ب(المستوي العلوي) و يتم تطبيقه بتحسين طريقة LSB وتم تسميتها ب (secure LSB-L1) لإخفاء الصور بحيث يتم اخفاء الرسالة بطريقة عشوائية داخل بتات الLSB وذلك بإستخدام مفتاح سري. البيانات السرية عبارة عن نص باللغة الانجليزية يتم إخفائه في صورة نقطية (Bitmap), الناتج يكون عبارة عن صورة بها نص مخفي(stego_image) تسمي ب(الصورة الوسيطة)

المستوي الثاني يسمي ب(المستوي الادني) وتم تطبيقه ايضا بإستخدام و تحسين طريقة LSB للإخفاء في الصور وتم تسميتها (secure LSB-L2) وهنا نقوم باختيار 3 × 3 نقاط بكسل لاخفاء الرسالة . في هذا المستوي ايضا يتم استخدام صورة نقطية أخري (Bitmap) كغطاء يتم فيها اخفاء الصورة الناتجة من المستوي الأولي و الناتج يكون عبارة عن صورة (Bitmap) جديدة (stego_image).

تم ضغط الصورة المخرجة من مستوى الاول بإستخدام خوارزمية برنامج (winrar)مرة و خوارزمية (Huffman) مرة اخري و خوارزمية (LZW) مرة ثالثة . و تحسنت عملية أخفاء الصورة في المستوى الثاني بسبب إستخدام خوارزميات الضغط علي الصورة المخرجة من عملية الاخفاء الاولي .

هنالك العديد من التجارب تم إجرائها بعد الإنتهاء من تطبيق النظام المقترح ، وهذه التجارب تم إجرائها بإستخدام أحجام مختلفة من الرسائل السرية و أيضا أحجام مختلفة من الصور التي يتم الإخفاء فيها

واخيرا تم اجراء عملية مقارنة للنتائج بناءا علي عوامل قياس جودة الصور مثل قيم (MSE) و (PSNR) و قد اظهرت نتائج جيدة للنظام المقترح.

CHAPTER ONE

INTRODUCTION

CHAPTER ONE

Introduction

1.1 Background

The security of information is one of the most important factors of information technology and communication. The security of information often lies in the secrecy of its existence and/or the secrecy of how to decode it. Cryptography, watermarking and Steganography can be used in information security. The cryptography techniques hide secret information by encrypts it using an encryption key(s), the output of encryption is chipper text or the secret information in an unreadable format, and this may draw the attention of attackers to the existence of confidential information. The digital watermarking is the process of embedding information into digital multimedia content such that the information (the watermark) can later be extracted or detected for a variety of purposes including copy prevention and control. The proposed method of information security in the research is steganography (Singla and Rupali 2013).

Steganography is defined as "the art and science of communicating in a way which hides the existence of the communication". Methods of steganography have existed for centuries, though with the advent of digital technology, have taken on a new form. Embedding data within the redundancy and noise of media files is among these digital techniques. (Beau Grantham 2007)

Steganography can be classified into image, text, audio and video steganography based on the cover media used to embed secret data. Images are the most popular cover objects used for steganography. In the domain of digital images many different image file formats exist, most of them for specific applications. For these different image file formats, different steganography algorithms exist. (Samir Kumar (2012)

Steganography (from Greek Steganos, or "covered," and graphic, or "writing") is the hiding of a secret message within an ordinary message and the extraction of it at its destination. Steganography takes cryptography a step further by hiding.

The steganography goal is to hide the presence of a message within another message called cover message, so steganography can be seen as the complement of cryptography whose goal is to hide the content of a message. (Eltyeb E.Abedelgabar)

Least Significant Bit (LSB) image steganography one of the earliest techniques studied in the information hiding of digital image (as well as other media types) is Least Significant Bit modification coding technique. In this technique, LSB of binary sequences of each sample of the digitized image file is replaced with the binary equivalent of a secret message. The advantage of this technique, it is the simplest way to embed information in a digital audio file. It allows a large amount of data to be concealed within an image file. This method is more widely used as modifications to LSBs usually not create audible changes to the sounds. The LSB has disadvantages, it has considerably low robustness against attacks (Sridevi and Narasimham. 2009)

1.2 Problem Statement

When hiding all Message into single image some problem may occur, it is possible the hackers obtained to all text quite easily if detected. Systems that use only one level of Steganography are usually more vulnerable, due to the fact that they lack the complexity to keep the data secure., Furthermore the most commonly used Steganography algorithm which is the normal (LSB) algorithm is proved to be weak and the secret data is easy to retrieve.

1.3 Propose Solution

For this reason Above, the proposed system uses a modified more secure version of LSB called the (secure LSB-L1), and a separate message over more than one images. The last step in this level, adding a key string to secure the information.

A lossless data compression technique uses Huffman, LZW algorithms and Winrar Application between the First and Second levels of steganography. The second Steganography level also employs another strong algorithm called (secure LSB-L2). In this level (secure LSB-L2) provides using several layers lieu of using only LSB layer of the image. Writing data starts from the last layer (8th or LSB layer); because significant of this layer is least and every upper layer has doubled significant from its down layer. So every step we go to upper layer image quality decreases and image retouching transpires.

The secret text message used here is the English language text, in field text design by c# under Microsoft Visual Studio 2010 Express. The images used are BMP image Finally, study the effect of these algorithms on image steganography.

Figures 1.1, 1.2 and 1.3 explain the scope of the proposed method in details.

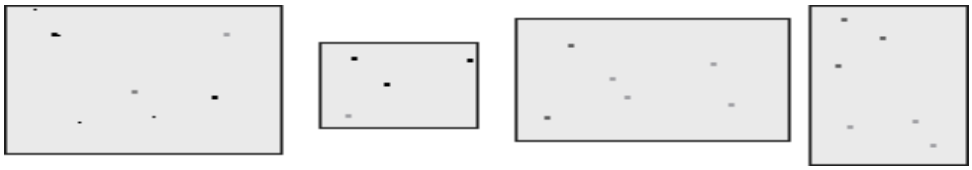

Figure 1.1: spreading the pixels over multiple images.

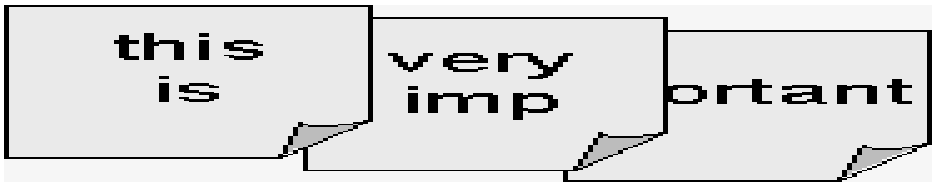

Figure 1.2: Spread the information over the images

And the following figure 1.3 shows the proposed algorithm to hide information:

1.4 Research Scope

The scope of this research will be steganography technique, especially multilevel steganography (MLS) focusing on Image steganography. The hiding of secret information (text) will be achieved by two levels of image steganography.

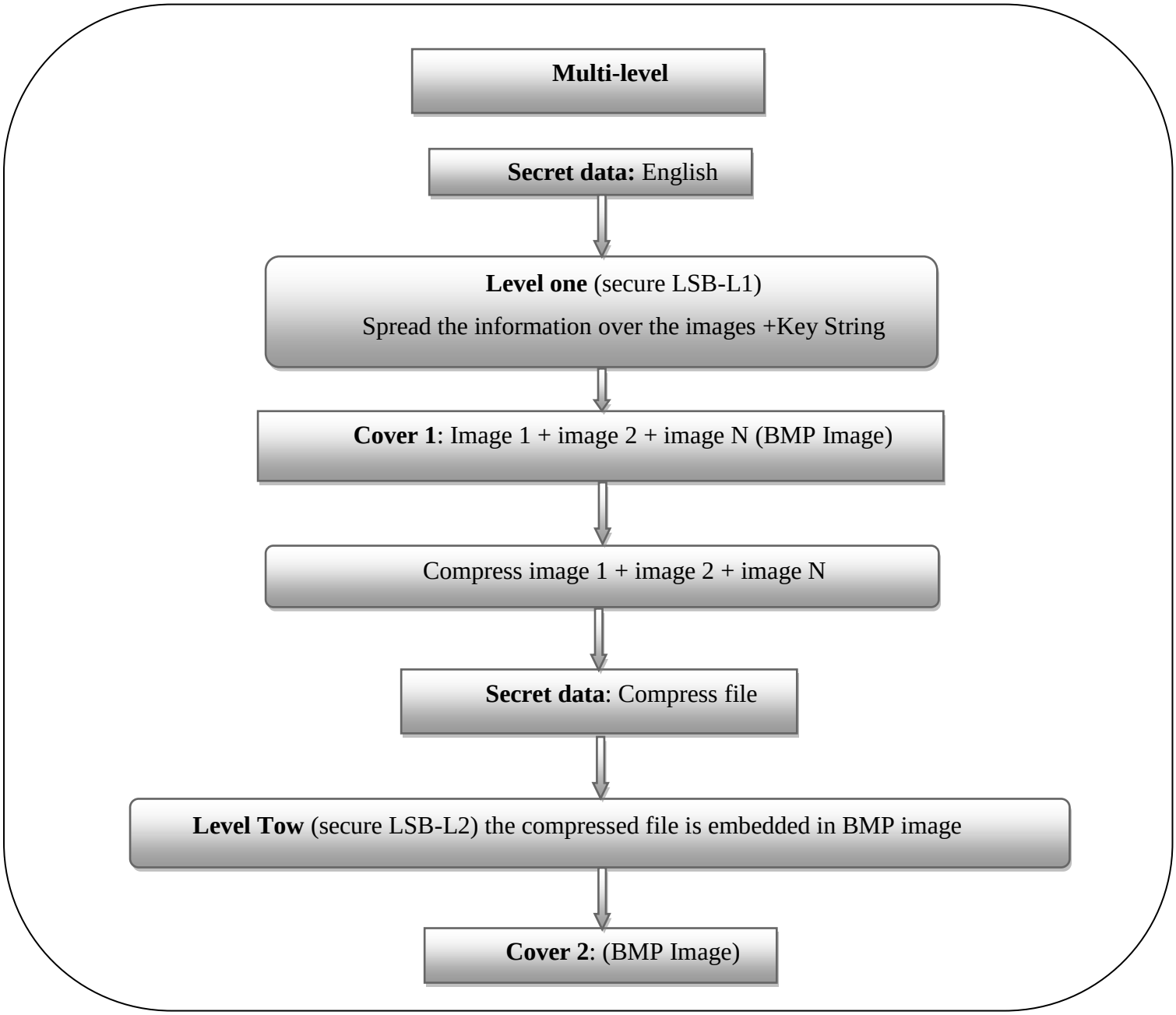

Figure 1.3 shows the proposed algorithm to hide information

1.5 Objective of the Research

- Hide information and avoid suspicion of a hidden message.
- Improve the way to hide the information division the text on more BMP images.
- An attempt to add some complexity to hide information by using Tow levels of steganography.
- Enhancing the confidentiality of the secret information**.**

1.6 Research Questions

- How the proposed method helps in hiding the secret information (text) to protect it from unauthorized disclosure?
- What are the results that can be accessed at the end of the research?

1.7 Research Methodology and Tools

There are a lot of studies done knowingly about steganography and separate studies on encryption and studies collected between encryption and steganography.

In this research by applying deep study in one-level LSB image steganography techniques, we discovered the existence of vulnerabilities in LSB image steganography, multi-level steganography can meet the vulnerabilities found in LSB by adding another level of image steganography Using compression techniques and Spread the information over the images in level One. And the following figure 1.4 shows the proposed algorithm to hide information:

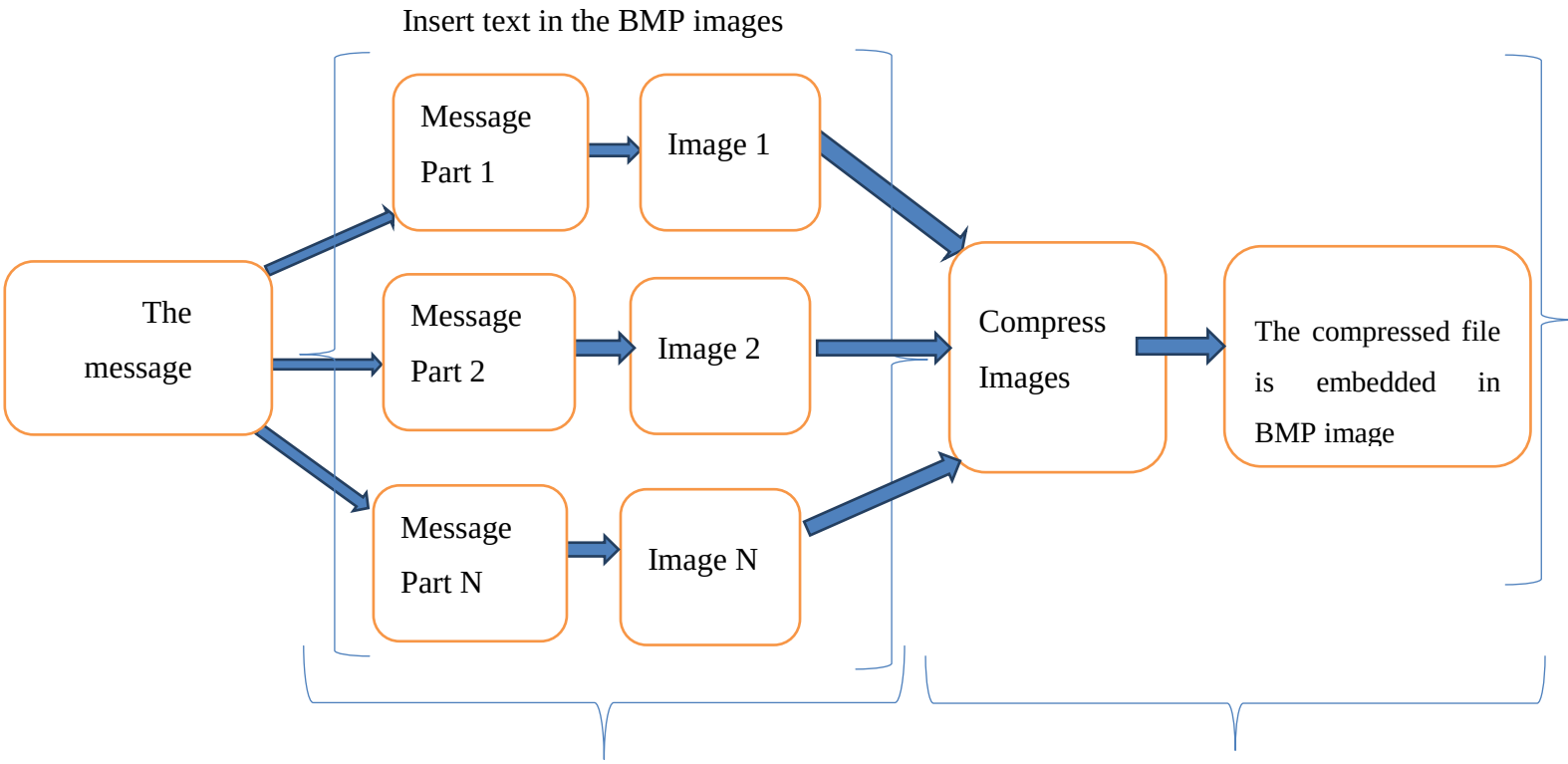

Figure 1.4 shows the proposed algorithm to hide information

1.7 Research Organization

Chapter one gives an introduction and brief history about the steganography, defining the types of steganography and multilevel steganography. Recently literature review will be explained in chapter two. Chapter three explains the proposed algorithm, tools and techniques used in the project. The analysis of the proposed algorithm and discussion of the results appears in chapter four and finally, Chapter five contains the conclusion, recommendations, and future work.

CHAPTER 2

LITERATURE REVIEW AND RELATED WORK

CHAPTER TWO

Literature Review

2.1 Overview

Steganography is the art of hiding and transmitting data through apparently innocuous carriers in an effort to conceal the existence of the data, the word Steganography literally means covered or hiding writing as derived from Greek. Steganography has its place in securing. It is not intended to replace cryptography, but supplement it. Hiding a message with Steganography methods reduces the chance of a message being detected. If the message is also encrypted then it provides another layer of protection (Johnson, 2006).

Therefore, some Steganographic methods combine traditional Cryptography with Steganography; the sender encrypts the secret message prior to the overall communication process, as it is more difficult for an attacker to detect embedded cipher text in a cover (Sellar, 2003).

In the field of Steganography, some terminology has developed. The adjectives 'cover', 'embedded', and 'stego' were defined at the information hiding workshop held in Cambridge, England. The term "cover" refers to a description of the original, innocent message, data, audio, video, and so on. Steganography is not a new science; it dates back to ancient times. It has been used through the ages by ordinary people, spies, rulers, government, and armies (Sellars, 2002).

There are many stories about Steganography. For example, ancient Greece used methods for hiding messages such as hiding it in the belly of a hare (a kind of rabbits), using invisible ink and pigeons. Another ingenious method was to shave the head of a messenger and tattoo a message or image on the messenger's head. After allowing his hair to grow, the message would be undetected until the head was shaved again. While the Egyptian used illustrations to conceal message (Davern, 2002).

2.2 Methods for Hiding Information.

Described below are some examples (Inoue, 2006; Noel, 2005) of those methods:

2.2.1 Hiding in Text

Most of the methods for hiding information into text process text as image essentially, text document, images, however, are quite special types of images, which have large blank areas, structured frequency spectra, and small meaningful subunits (words and letters). A variety of methods are available for hiding information in a text by introducing variations in letters, words, and line spacing.

2.2.2 Hiding in Disk Space

Another way to hide information relies on hiding unused space that is not readily apparent to an observer. Taking advantage of an unused or reserved space to hold covert information provides a mean of hiding information without perceptually degrading the carrier. The way operating system stores files typically result in an unused space that appears to be allocated to files. Another method of hiding information in a file system is to create hidden partitions. These partitions are not seen if the system is started normally.

2.2.3 Hiding in Network Packets

With the rapid development of Internet technologies, the number of data packets sent and received electronically is increasing greatly. As the technology of transmitting information on the network in securely, the importance of information hiding as a field of information security comes to be recognized widely. Any of these packets can provide a covert communication channel. For example, TCP/IP packets are used to transport information over the internet. The packet headers have unused space or other features that can be manipulated to hide information.

2.2.4 Hiding in Software and Circuitry

Data can also be hidden based on the physical arrangement of the carrier. The arrangement itself may be an embedded signature that is unique to the creator. An example is in the layout of code distribution in a program or the layout of electronic circuits on a board. This type of ‘marking’ can be used to uniquely identify the origin and design cannot be removed without significant change in the work.

2.2.5 Hiding in Image and Audio

Many different methods enable hiding information in audio and image. 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 level (noise), manipulation of compression algorithms, and the modification of carrier properties. In audio, small echoes or slight delays can be added or subtle signals can be masked by the sound of higher amplitude. Information (data) can be hidden in different ways in the image. To hide information; straight message insertion may encode every bit of information in the image or selectively embed the message in busy areas where it would be less perceptible. A message may also be scattered randomly or repeated several times throughout the image. The following below table 2.1 shown and explain Weaknesses of Method for Hiding Information.

Table 2.1: Weaknesses of Method for Hiding Information. Aos, A.Z.Ansaef (2009)

Methods for Hiding Information.	Weakness
Hiding in Text	Methods of concealment in the text are weak and inefficient not suitable for the application and the main disadvantage include, need large text to hide small message that leads to an increase in the size of the cover.
Hiding in Disk Space	Methods of concealment in the unused areas weak and inefficient not suitable for the application and the main disadvantage include, this is not efficient since its easy detection by using some service software such as Norton antivirus.
Hiding in Network Packets	Methods of concealment in the network packets are weak and inefficient not suitable for the application and the main disadvantage include, Limitation of size for hiding information and It requires other methods to hide the data.
Hiding in Software and Circuitry	Methods of concealment in the (software or circuitry) are weak and inefficient not suitable for the application and the main disadvantage include, It is not possible to combine the (maximize/ maximum) strength of Robustness with maximize/maximum amount of hidden data comparing to data cover, which the hiding method cannot makes the relation between the cover and the message independent.
Hiding in Image and Audio	Methods of concealment in the (image or audio) are weak and inefficient not suitable for the application and the main disadvantage include: a) The size of the output of the hidden data file is larger comparing to the encoded data. In its most efficient possible case, it may reach double the size of encoded data or a bet less . b) In some situations output file may reach eight times larger than the encoded data, as well as certain files of media images, files may reach fifty times larger when they are encoded. c) In the case of using a cover environment with equal value spaces as in the pictures with constant value colour spaces (week texture) or sounds with constant intensity sound intervals, that may lead to discovery or differentiation at these sectors.

From the table above, it is clear that the main weaknesses of those methods are the size of the hidden data depends on the size of the cover file and any changes made are easily detected by antivirus software. Thus, through this research, these weaknesses will be solved through the use of the EXE file.

2.3 Cryptography and Steganography.

Since the advent of computers, there has been a vast dissemination of information, some of which needs to be kept private, some of which doesn't. The information may be hidden in two basic ways (Cryptography and Steganography) (Al-Dieimy, 2002). The methods of Cryptography does not conceal the presence of secret information, but render it unintelligible to an outsider by various transformations of the information that is to be put into secret form, while methods of Steganography conceal the very existence of the secret information. The following table shows the comparison between Cryptography and Steganography; (Dorothy, 2000). Table 2.2 below show comparison between Cryptography and Steganography

Table 2.2: Comparison between Cryptography and Steganography Aos, A.Z.Ansaef (2009)

Cryptography	Steganography
1-The encrypted letter could be seen by anyone but cryptography make the message not understandable.	1-Steganography is hiding the massage in another median so that nobody will notice the message.
2-The end result in cryptography is the cipher text.	2-The end result of information hiding is the stego-media.
3-The goal of a secure cryptographic is to prevent an interceptor from gaining any information about the plaintext from the intercepted cipher text.	3-The goal of secure Steganographic methods is to prevent an observant intermediary from even obtaining knowledge of the mere presence of the secret data.
4-Any person has the ability of detecting and modifying the encrypted message.	4-The hidden message is imperceptible to anyone.
5-Steganography cannot be used to adapt the robustness of cryptographic system.	5-Steganography can be used inconjunction with cryptography by hiding an encrypted massage.

2.4 General Steganography Systems

A general Steganography system is shown in Figure 2.1. It is assumed that the sender wishes to send via Steganographic transmission, a message to a receiver. The sender starts with a cover message, which is an input to the stego-system, in which the embedded message will be hidden. The hidden message is called the embedded message. A Steganographic algorithm combines the cover message with the embedded message, which is something to be hidden in the cover (Nspw, 2006). The algorithm may, or may not, use a Steganographic key (stego- key), which is additional secret data that may be needed in the hidden process. The same key (or a related one) is usually needed to extract the embedded message again. The output of the Steganographic algorithm is the stego message. The cover message and Stego-message must be of the same data type, but the embedded message may be of another data type. The receiver reverses the embedding process to extract the embedded message (Avedissian, 2005).

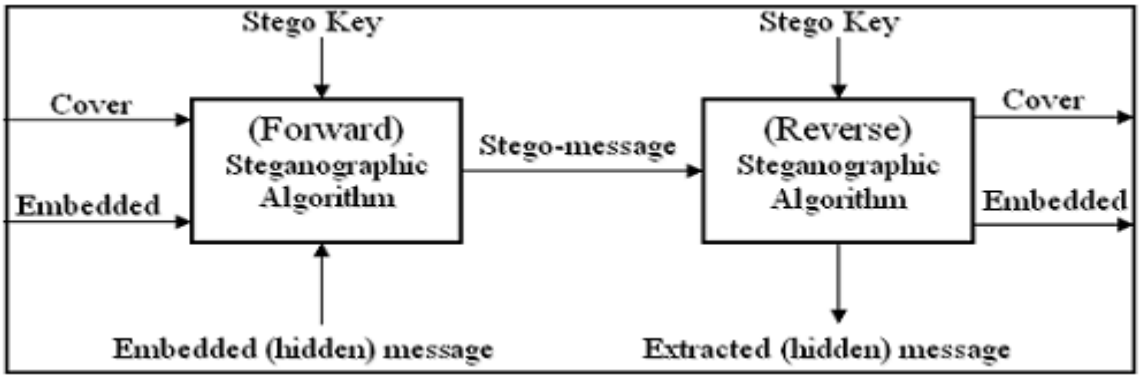

Figure 2.1: General Steganography System.

2.5 Characterization of Steganography Systems

Steganographic techniques embed a message inside a cover. Various features characterize the strength and weaknesses of the methods. The relative importance of each feature depends on the application (Brigit, 1996).

2.5.1 Capacity

The notion of capacity in data hiding indicates the total number of bits hidden and successfully recovered by the Stego system (Ross, 2005).

2.5.2 Robustness

Robustness refers to the ability of the embedded data to remain intact if the stego-system undergoes a transformation, such as linear and nonlinear filtering; the addition of random noise; and scaling, rotation, and lose compression (Brigit, 1996).

2.5.3 Undetectable

The embedded algorithm is undetectable if the image with the embedded message is consistent with a model of the source from which images are drawn.

For example, if a Steganography method uses the noise component of digital images to embed a secret message, it should do so while not making statistical changes to the noise in the carrier. Undetectability is directly affected by the size of the secret message and the format of the content of the cover image (Ross, 2005).

2.5.4 Invisibility (Perceptual Transparency)

This concept is based on the properties of the human visual system or the human audio system. The embedded information is imperceptible if an average human subject is unable to distinguish between carriers that do contain hidden information and those that do not. (Ross, 2005) It is important that the embedding occurs without a significant degradation or loss of perceptual quality of the cover (Brigit, 1996).

2.5.5 Security

It is said that the embedded algorithm is secure if the embedded information, is not subject to removal after being discovered by the attacker and it depends on the total information about the embedded algorithm and a secret key (Lin, 2005).

2.6 Classification of Steganography Techniques

There are several approaches in classifying Steganographic systems. One could categorize them according to the type of covers used for secret communication or according to the cover modifications applied in the embedding process. The second approach will be followed in this section, and the Steganographic methods are grouped into six categories, although in some cases an exact classification is not possible. Figure 2.2 presents the Steganography classification (Katzenbisser, 2000).

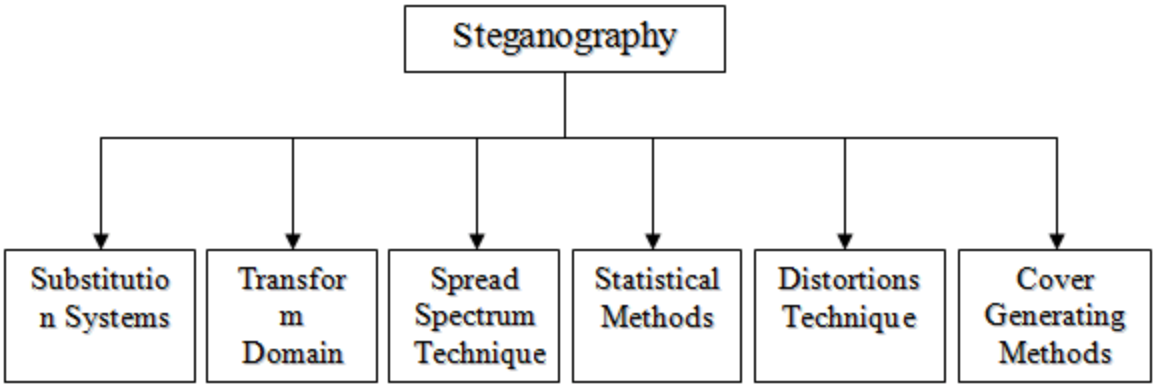

Figure 2.2: Steganography Classification (Sellars, 2002).

2.6.1 Substitution Systems

Basic substitution systems try to encode secret information by substituting insignificant parts of the cover by secret message bits. The receiver can extract the information if he has knowledge of the positions where secret information has been embedded. Since only minor modifications are made in the embedding process, the sender assumes that they will not be noticed by an attacker. It consists of several techniques that will be discussed in more detail, in the following subsection (Lin, 2005):

2.6.1.1 Least Significant Bit Substitution (LSB)

The embedding process consists of choosing a subset {j1… Jl (m)} of cover elements and performing the substitution operation CJI _ mi on them, which exchange the LSB of 'CJI' by mi (mi can be either 1 or 0).In the extraction process, the LSB of the selected cover element is extracted and lined up to reconstruct the secret message (Ross, 2005).

2.6.1.2 Pseudorandom Permutation

If all cover bits are accessed in the embedding process, the cover is a random-access cover, and the secret message bits can be distributed randomly over the whole cover. This technique further increases the complexity for the attacker, since it is not guaranteed that the subsequent message bits are embedded in the same order (Ross, 2005).

2.6.1.3 Image Downgrading and Cover Channels

Image downgrading is a special case of a substitution system in which image acts both as a secret message and a cover. Given cover-image and secret image of equal dimensions, the sender exchanges the four least significant bits of the cover gray scale (or color) values with the four most significant bits of the secret image. The receiver extracts the four least significant bits out of the stego-image, thereby gaining access to the most significant bits of the stego-image.

Whereas the degradation of the cover is not visually noticeable in many cases, four bits are sufficient to transmit a rough approximation of the secret image (Katzenbisser, 2005).

2.6.1.4 Cover Regions and Parity Bits

Any nonempty subset of {c1,……., cI (c)} is called a cover region. By dividing the cover into several disjoint regions, it is possible to store one bit of information in a whole cover-region rather than in a single element (Al-Hamami, 2006). A parity bit of a region I can be calculated by B (I) ==_LSB (cj) mod2

2.6.1.5 Palette-Based Image

There are two ways to encode information in a palette-based image; either the palette or the image data can be manipulated. The LSB of the color vectors could be used for information transfer, just like the substitution methods presented. Alternatively, since the palette does not need to be sorted in any way, information can be encoded in the way the colors are stored in the palette. For N colors since there are N! Different ways to sort the palette, there is enough capacity to encode a small message. However, all methods which use the order of a palette to store information are not robust, since an attacker can simply sort the entries in a different way and destroy the secret message (Lin, 2005).

2.6.2 Transform Domain Techniques

It has been seen that the substitution and modification techniques are easy ways to embed information, but they are highly vulnerable to even small modification. An attacker can simply apply signal processing techniques in order to destroy the secret information. In many cases, even the small changes resulting out of loose compression systems yield total information loss. It has been noted in the development of Steganographic systems that embedding information in the frequency domain of a signal can be much more robust than embedding rules operating in the time domain. Most robust Steganographic systems known today actually operate in some sort of transform domain. Transformation domain methods hide messages in a significant area of the cover image which makes them more robust to attack, such as adding noise, compression, cropping some image processing. However, whereas they are more robust to various kinds of signal processing, they remain imperceptible to the human sensory system. Many transform domain variations exist. One method is to use the Discrete Cosine Transformation (DCT) as a vehicle to embeds information in an image. Another method would be the use of wavelet transforms (Katzenbisser, 2000). Transforms embedding embeds a message by modification (selected) transform (e.g., frequency) coefficient of the cover message. Ideally, transform embedding has an effect on the spatial domain of apportioning the hidden information through different order bits in a manner that is robust, but yet hard to detect. Since an attack, such as image processing, usually affects a certain band of transforming coefficient, the remaining

coefficient would remain largely intact. Hence, transform embedding is, in general, more robust than other embedding methods (Katzenbisser, 2000).

2.6.3 Spread Spectrum (SS) Techniques

Spread spectrum techniques are defined as "Means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information". The band spread is accomplished by means of a code which is independent of the data, and a synchronized reception with the code at the receiver is used for despreading and subsequent data recovery. Although the power of the signal to be transmitted can be large, the signal–to–noise ratio in every frequency band will be small, even if parts of the signal could be removed in several frequency bands, enough information should be present in the other bands to recover the signal. This situation is very similar to a Steganography system which tries to spread a secret message over a cover in order to make it impossible to perceive. Since spread signals tend to be difficult to remove, embedding methods based on SS should provide a considerable level of robustness (Sellars, 2002). In information hiding, two special variants of spread spectrum techniques are generally used: direct sequence, and frequency–hopping scheme. In the direct-sequence scheme, the secret signal is spread by a constant called ship rate, modulated with a pseudorandom signal and added to the cover.

On the other hand, in the frequency–hopping schemes the frequency of the carrier signal is altered in a way that it hopes rapidly from one frequency to another. SS are widely used in the context of watermarking (Al_Mayyahee, 2005).

2.6.4 Distortion Techniques

In contrast to substitution systems, distortion requires the knowledge of the original cover in the decoding process. The sender applies a sequence of modifications to the cover in order to get a stego-system. A sequence of modification is chosen in such a way that it corresponds to a specific secret message to be transmitted. The receiver measures the difference in the original cover in order to reconstruct the sequence of modification applied by the sender, which corresponds to the secret message. An early approach to hiding information is in the text. Most text-based hiding methods are of distortion type (i.e., the

arrangement of words or the layout of a document may reveal information). One technique is by modulating the positions of line and words, which will be detailed in the next subsection. Adding spaces and “invisible” characters to text provide a method to pass hidden information HTML files are good candidates for including extra spaces, tabs, and line breaks. Web browsers ignore these “extra” spaces and lines and they go unnoticed until the source of the web page is revealed (Sellar, 2003).

2.6.5 Cover Generation Techniques

In contrast to all embedding methods presented above, when secret information is added to a specific cover by applying an embedding algorithm, some Steganographic applications generate a digital object only for the purpose of being a cover for secret communication (Katzenbisser, 2000). And the below table 2.4 shown Weaknesses of Steganography Techniques.

Table 2.3: Weaknesses of Steganography Techniques (Aos, A.Z.Ansaef 2009)

Steganography Techniques	Weakness
Substitution Systems	Low robustness: filtering, lossy compression attacks, format file dependand.
Transform Domain Techniques	An attacker can simply apply signal processing techniques in order to destroy the secret information. In many cases even the small changes resulting out of loose compression systems yield total information loss.
Spread Spectrum (SS) Techniques	There are increases in the complexity, higher costs and more stringent timing requirements. a) Direct-Sequence Scheme:The circuitry required to produce the spectrum is complex, it requires a large bandwidth channel with relatively small phase distortions and requires a long acquisition time since the PN codes are long. b) Frequency–Hopping Scheme: Weakness with both slow and fast hopping. With slow hopping, coherent data detection is possible, but data can be lost if a single frequency hop channel is jammed. To overcome this, it is necessary to use error correcting codes. Fast hopping disposes of the need for error codes since one bit of data is spread over a number of hops. However, fast hopping has the disadvantage that due to phase discontinuities, coherent data detection is not possible.
Distortion Techniques	In many applications, such systems are not useful, since the receiver must have access to the original cover.It is weakness point.So if the attacker also has access to them, he/she can easily detect the cover modification and has evidence for a secret communication.If the embedding and extraction functions are public and do not depend on a stego-key, it is also possible for the attacker to reconstruct secret message entirely.
Cover Generation Techniques	They have heavy and complexity process for algorithems compersion with other techniques. This point due to dealy time for finished (hiding or extract) process operation .Example: Automated Generation of English Text. Use a large dictionary of words categorised by different types, and a style source which describes how words of different types can be used to form a meaningful sentence. Transform message bits into sentences by selecting words out of the dictionary which conforms to a sentence structure given in the style source.

From the above table, most of the techniques are very complex and not suitable to be used with the EXE file. In order to use the EXE file, a simple technique needs to be applied so that changes made to the file will not be detected by antivirus software. Thus, we choose to apply statistical techniques because it is not complex and suitable to be implemented with the structure and characteristic of the EXE file.

2.6.6 Statistical Steganography

Statistical Steganography techniques utilize the existence of "1-bits"Steganography schemes, which embed one bit of information in a digital carrier.

This is done by modifying the cover in such a way that some statistical characteristics change significantly if a "1" is transmitted. Otherwise, the cover is left UN changed. So, the receiver must be able to distinguish unmodified covers from modified ones. A cover is divided into l (m) disjoint blocks B1... B l (m). A secret bit, MI is inserted into the ith block by placing "1" into Bi if MI=1.Otherwise, the block is not changed in the embedding process (Katzenbisser, 2000).

2.7 Types of steganography:

The different types of steganographic techniques that are available:

1. Pure steganography
2. Public key steganography
3. Secret key steganography

1-Pure steganography:

Pure steganography is the process of embedding the data into the object without using any private keys. This type of steganography entirely depends upon the secrecy. This type of steganography uses a cover image in which data is to be embedded, personal information to be transmitted, and encryption-decryption algorithms embed the message into an image.

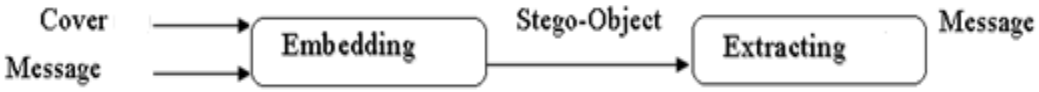

Figure 2.3: pure steganography process (Zaidoon, 2010).

This type of steganography can‟t provides better security because it is easy for extracting the message if the unauthorized person knows the embedding method. It has one advantage that it reduces the difficulty in key sharing (Zaidoon, 2010).

2-Secret key steganography:

Secret key steganography is another process of steganography, which uses the same procedure other than using secure keys. It uses the individual key for embedding the data into the object which is similar to asymmetric key. For decryption, it uses the same key which is used for encryption.

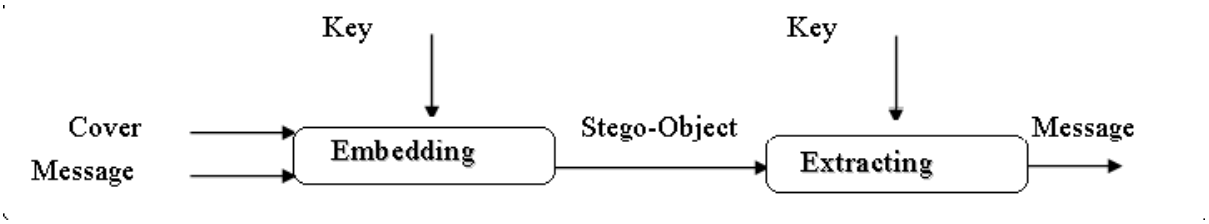

Figure 2.4: secret key steganography (Zaidoon, 2010)

This type of steganography provides better security compared to pure steganography. The main problem of using this type of steganographic system is sharing the secret key. If the attacker knows the key it will be easier to decrypt and access original information (Zaidoon, 2010).

3-Public key steganography:

Public key steganography uses two types of keys: one for encryption and another for decryption. The key used for encryption is a private key and for decryption, it is a „public key‟ and is stored in a public database (Zaidoon, 2010).

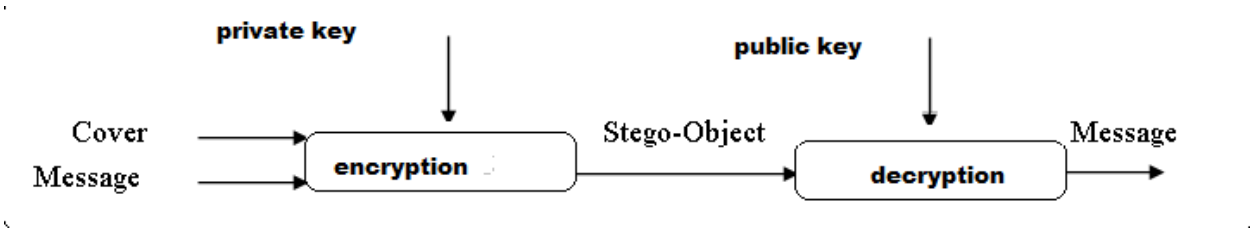

Figure 2.5: public key steganography (Zaidoon, 2010).

For encryption and decryption of text messages using the secret keys, the steganographic system uses algorithms known as steganographic algorithms. The most used algorithms for embedding data into images are (Zaidoon, 2010).

- LSB (Least Significant Bit) Algorithm
- JSteg Algorithm.
- F5 Algorithm.

2.8 Steganography Advantages and Disadvantages

Steganography is the art of hiding information inside another media; this is basically done by using unnecessary bits in an innocent file to store the secret data. The techniques used make it impossible to detect that there is anything in the innocent file, but the intended recipient can get the hidden data.

Steganography has its place on security, but not instead of Cryptography, it supplements it. Hiding a message with Steganography methods reduces the chance of detecting the message (Jafer, 2006).

It can and will be used for two purposes, good and evil. Both government and private industries need to worry about the use of Steganography inside of their respective buildings (Katzenbisser, 2000; Jafer, 2006). The main advantages of Steganography are:

• It can be used for secretly transmitting a message without being discovered.

• It does not allow the enemy to detect that there is secret information inside the cover file.

• It can be used by anyone using the internet.

However, Steganography has several disadvantages. They are:

• It generally requires a lot of overhead to hide a relatively few bit of information. However, there are ways around this.

Once a Steganography system is discovered, it is rendered useless. This problem, too, can be overcome if the hidden data depend on some sort of key for its insertion and extraction.

2.9 Image Steganography

Image Steganography has many applications, especially in today's modern, high-tech world. Privacy and anonymity are a concern for most people on the internet. Image Steganography allows for two parties to communicate secretly and covertly. It allows for some morally-conscious people to safely whistle blow on internal actions; it allows for copyright protection for digital files using the message as a digital watermark. One of the other main uses for Image Steganography is for the transportation of high-level or top-secret documents between international governments. While Image Steganography has many legitimate uses, it can also be quite nefarious. It can be used by hackers to send viruses and Trojans to compromise machines, and also by terrorists and other organizations that rely on covert operations to communicate secretly and safely.

Image steganography techniques can be divided into two groups the spatial Domain or Image and Transform Domain or frequency domain.

2.10 Multilevel Steganography (MLS)

Multilevel Steganography can be utilized to achieve various aims – it all depends on how it will be used. Here we present several of the most interesting MLS applications, in our opinion. The benefits of MLS of hidden data exchange are summarized in the below Table.2.4:

In MLS, at least two steganographic methods are utilized simultaneously, in such a way that one method (called the upper-level) serves as a carrier for the second one (called the lower-level). Such a relationship between two (or more) information hiding solutions, has several potential benefits. The most important are that the lower-level method steganographic bandwidth can be utilized to make the steganogram unreadable even after the detection of the upper-level method: e.g., it can carry a cryptographic key that deciphers the steganogram carried by the upper level one. It can also be used to provide the steganogram with integrity. Another important benefit is that the lower-layer method may be used as a signaling channel in which to exchange information that affects the way that the upper-level method functions, thus possibly making the steganographic communication harder to detect. Table. 2.4: The benefits of MLS for hidden data exchange are summarized.

Table 2.4: MLS benefits and possible applications.

MLS benefit	Described MLS application
Increased steganography bandwidth for user data	Using two or more steganography. methods increases the total Stenograhic. bandwidth achieved for user data compared with a single Steganography method.
Increased undetectability	An upper-level method controlled by the information carried by the lower-level method
Steganogram transmission reliability	Lower-level method carrying information for steganogram integrity verification.
Harder steganogram extraction and analysis	1. Cryptographic key carried by the lower level method and upper-level method steganogram ciphered. 2. Parts of the steganogram sent using the upper-level and others by the lower-level method. 3. Steganogram carried only by the lower-level method; upper-level steganogram only for masking
Steganography cost unchanged	In best case scenario, depends on the upper- and lower-level methods used, but can be the same as for utilization Of the upper-level Method alone.

Let us consider the above-mentioned MLS applications based on where the steganogram is inserted. There are three possible cases:

- Steganogram is carried only by upper-level method.
- Steganogram is carried only by the lower - level method.
- Steganogram is carried by both upper- and lower-level methods.

2.11 Data compression

Data compression is a process by which a file (Text, Audio, and Video) may be transformed to another (compressed) file, such that the original file may be fully recovered from the original file without any loss of actual information. This process may be useful if one wants to save the storage space. For example, if one wants to store a 4MB file, it may be preferable to first compress it to a smaller size to save the storage space. (Er. Meenakshi Garg 2014)

Also, compressed files are much more easily exchanged over the internet since they upload and download much faster. We require the ability to reconstitute the original file from the compressed version at any time. Data compression is a method of encoding rules that allows a substantial reduction in the total number of bits to store or transmit a file. The more information being dealt with, the more it costs in terms of storage and transmission costs. In short, Data Compression is the process of encoding data to fewer bits than the original representation so that it takes less storage space and less transmission time while communicating over a network. . (Er. Meenakshi Garg 2014)

Data Compression is possible because most of the real-world data is very redundant. Data Compression is basically defined as a technique that reduces the size of data by applying different methods that can either be Lossy or Lossless. A compression program is used to convert data from an easy - to -use format to one optimized for compactness. Likewise, an uncompressing program returns the information to its original form. . (Er. Meenakshi Garg 2014)

Figure: 2.6 Data Compression and Decompression. (Er. Meenakshi Garg 2014)

2.12 Types of Data Compression

Currently, two basic classes of data compression are applied in different areas. One of these is a lossy data compression, which is widely used to compress image data files for communication or archive purposes. The other is a lossless data compression that is commonly used to transmit or archive text or binary files required to keep their information intact at any time. . (Er. Meenakshi Garg 2014)

There are two mainly two types of Data Compression:

- Lossy Compression.
- Lossless Compression.

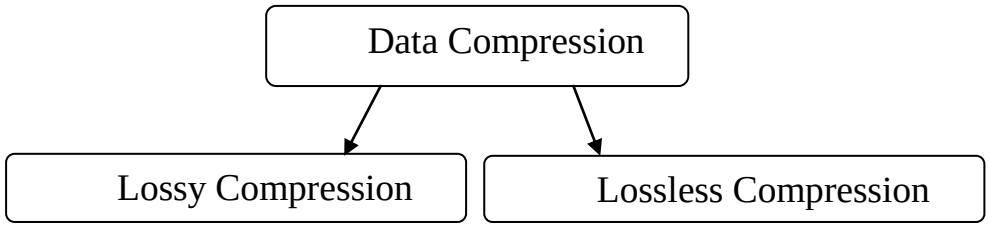

Figure: 2.7 Classification of Data Compression

2.12.1 Lossy Data Compression

A lossy data compression method is one where the data retrieves after decompression may not be exactly same as the original data, but is "close enough" to be useful for specific purposes. After one applies lossy data compression to a message, the message can never be recovered exactly as it was before it was compressed. When the compressed message is decoded it does not give back the original message. The data has been lost. Because lossy

compression cannot be decoded to yield the exact original message, it is not a good method of compression for critical data, such as textual data. It is most useful for Digitally Sampled Analog Data (DSAD). DSAD consists mostly of sound, video, graphics, or picture files. In a sound file, for example, the very high and low frequencies, which the human ear cannot hear, may be truncated from the file. The examples of the frequent use of Lossy data compression are on the Internet and especially in the streaming media and telephony applications. Some examples of lossy data compression algorithms are JPEG, MPEG, MP3. Most of the lossy data compression techniques suffer from generation loss which means decreasing the quality of text because of repeatedly compressing and decompressing the file. Lossy image compression can be used in digital cameras to increase storage capacities with minimal degradation of picture quality. . (Er. Meenakshi Garg 2014)

2.12.2 Lossless Data Compression

Lossless data compression is a technique that allows the use of data compression algorithms to compress the text data and allows the exact original data to be reconstructed from the compressed data. This is in contrary to the lossy data compression in which the exact original data cannot be reconstructed from the compressed data. The popular ZIP file format that is being used for the compression of data files is also an application of the lossless data compression approach. (Er. Meenakshi Garg 2014)

Lossless compression is used when it is important that the original data and the decompressed data be identical. Lossless text data compression algorithms usually exploit statistical redundancy in such a way so as to represent the sender's data more concisely without any error or any sort of loss of important information contained within the text input data. Since most of the real-world data have statistical redundancy, therefore lossless data compression is possible. For instance, In English text, the letter ’a’ is much more common than the letter 'z', and the probability that the letter’t’ will be followed by the letter 'z' is very small. So, this type of redundancy can be removed using lossless compression. Lossless compression methods may be categorized according to the type of data they are designed to compress. Compression algorithms are basically used for the compression of text, images, and sound. (Er. Meenakshi Garg 2014)

2.13 Lossless vs. lossy compression

Lossless compression algorithms usually exploit statistical redundancy in such a way as to represent the sender's data more concisely, but nevertheless perfectly. Lossless compression is possible because most real-world data has statistical redundancy. For example, in English text, the letter 'e' is much more common than the letter 'z', and the probability that the letter 'q' will be followed by the letter 'z' is very small. (NILESH and SANKAR 2007)

Another kind of compression, called lossy data compression, is possible if some loss of fidelity is acceptable. For example, a person viewing a picture or television video scene might not notice if some of its finest details are removed or not represented perfectly (i.e. May not even notice compression artifacts). Similarly, two clips of audio may be perceived as the same to a listener even though one is missing details found in the other. Lossy data compression algorithms introduce relatively minor differences and represent the picture, video, or audio using fewer bits. (NILESH and SANKAR 2007)

Lossless compression schemes are reversible, so that the original data can be reconstructed, while lossy schemes accept some loss of data in order to achieve higher compression. (NILESH and SANKAR 2007)

However, lossless data compression algorithms will always fail to compress some files; indeed, any compression algorithm will necessarily fail to compress any data containing no discernible patterns. Attempts to compress data that has been compressed already will therefore usually result in an expansion, as will attempts to compress encrypted data. (NILESH and SANKAR 2007)

ddccIn practice, lossy data compression will also come to a point where compressing again does not work, although an extremely lossy algorithm, which for example always removes the last byte of a file, will always compress a file up to the point where it is empty. (NILESH and SANKAR 2007)

A good example of lossless vs. lossy compression is the following string 888883333333. What you just saw was the string written in an uncompressed form. However, you could save space by writing it. By saying "5 eights, 7 threes", you still have the original string, just written in a smaller form. In a lossy system, using 83 instead, you cannot get the original data back (at the benefit of a smaller file size). (NILESH and SANKAR 2007)

2.14 Huffman algorithm

The Huffman algorithm is an encoding technique developed by D. A. Huffman which compresses data into a minimal form where values get an encoding depending on the value's frequency. Frequent values get a shorter encoding and infrequent values get a longer encoding. For a given dataset, the algorithm creates a table of the encountered values and their frequency. From this VFT, the Huffman algorithm generates individualized de_ned bit codes for the different values. These bit codes are of variable length depending on the value's frequency. (JONAS NIKLAS June 2015)

To guarantee that decompression will be possible, no encoding is allowed to be encoded so that it is a pre_x to any other encoding. Also, its pre_xes may not be used as an encoding for any other values. The Huffman algorithm creates a binary tree structure from the bottom up, using the frequencies to decide which nodes combine into a parent. Once the tree is built, binary encodings are assigned to the different nodes from the top down. The following example shows how the Huffman algorithm can be used to compress a simple data set. The dataset has been evaluated and four values were found in the dataset. We call these four values: A, B, C, and D. The dataset's VFT can be seen in Table 2.5(JONAS NIKLAS June 2015)

Table 2.5: VFT of example data set(JONAS NIKLAS June 2015)

Value	Frequency
A	0.40
B	0.35
C	0.20
D	0.05

From the VFT, the binary tree seen in Figure 2.14 is created. The steps included in the creation of the tree structure are listed below.

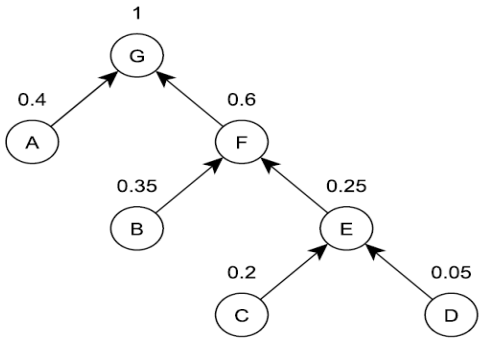

Figure 2.8:Creating a binary tree from the bottom-up.The two nodes with lowest Frequencies are combined into a single parent node (JONAS ANDERSSON and NIKLASDOVERBO 2015)

2.14.1 Achievable Huffman compression of states

Huffman compression performs best when exact probabilities of values in state vectors are known. These probabilities are unknown during verification of new models, but we can compute them for a set of known models to get a more exact estimate of the achievable compression ratio. We performed this computation on all models from Appendix A. As you can see in figure 3.5, average achievable compression ratio with full knowledge of model is

approximately 40 %. That is almost twice more efficient than the current approach implemented in DiVinE, using a static generated codebook.

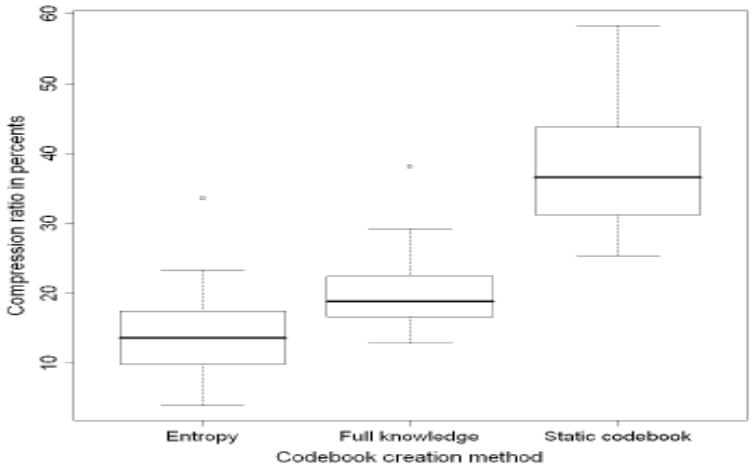

Figure 2.9: Achievable compression ratios and state vector entropy Huffman compression of states in DiVinE by Jaroslav Šeděnka Brno, 2007

2.15 LZW Algorithm

The best a lossless compression algorithm can do is to encode the output of a source such that the average number of bits equals the entropy of the source

2.15.1 LZ77 Algorithm

The LZ77 [Ziv77] algorithm is the first text compression algorithm proposed in the LZ (Zev-Lempel) family compression algorithms. It is incorporated by most commercial compression utilities such as ZIP and ARJ. The lz77 algorithm is a dictionary-based algorithm and the dictionary are both implicit and dynamic, as can be seen from the followings. The algorithm is very simple: *if the current symbol(s) has appeared somewhere before, then output a reference to that position; otherwise, output a null-reference and the first mismatched symbol*. Therefore, an LZ77 code is denoted as *(b, l) c,* which tells the decoder to "go back *b* characters, copy *l* characters from there and then append a *c* character".

The lz77 algorithm provides good compression performance, but the encoding takes a long time since a large number of comparisons need to be done. In contrast, the decoder is very fast since it only does simple "copy and paste".

For example, string "***AABCBBABC***", will be encoded as shown below:

Table 2.6: LZ77 Algorithm encoding table

Position	1	2	3	4	5	6	7	8	9
Character	A	A	B	C	B	B	A	B	C
Code	(0,0)A	(1,1)B		(0,0)C	(2,1)B		(5,2)C		

2.15.2 LZSS Algorithm

LZSS algorithm [Ziv77] is a modified version of theLZ77 algorithm. The following problems may affect the efficiency of the LZ77 algorithm:

1. When there is no match or only one match, it is too expensive to represent the characters (one character when no match and two characters when one match is found) using code (*b, l*) *c*.

2. Using code "(*b, l*) *c*" is probably not efficient since *c* might be the first symbol of the next match.

To solve the above problems, the LZSS algorithm simply sends a reference *(b, l)* when a match is found and it sends the original symbol when no match is found. Furthermore, a minimal matching length *M* is applied.

When an encoder looks for match symbols in the "dictionary", only the match that has a length larger than M is considered.

For the same example, if the minimal matching length is 2. The encoding of LZSS is shown below.

Table 2.7: LZSS Algorithm encoding table

Position	1	2	3	4	5	6	7	8	9
Character	A	A	B	C	B	B	A	B	C
Code	A	A	B	C	B	B	(5,2)		C

To be able to distinguish between a reference code and a symbol in the compressed data, an ID-bit is used in the code. Decoding in LZSS is as easy and quick as in LZ77.

LZSS algorithm is almost the same as an LZ77 algorithm in nature. Therefore, the discussion of compressed pattern matching we had for LZ77 algorithm should apply for LZSS algorithm too.

2.15.3 LZ78 Algorithm

Different than LZ77 and LZSS algorithms, LZ78 algorithm [Ziv78] uses an explicit dictionary. The idea is very simple: “Why not store the strings in a dictionary and outputs its index in the dictionary when a match is found. Anyway, an index such as 7 is shorter than a reference such as (5, 2)”. At the beginning of encoding, the dictionary is empty. Every time when a mismatch happens, i.e. no match is found in the dictionary in the current string, the encoder inserts the current string into the dictionary and outputs the index of the matched part in the dictionary, followed by the mismatched symbol the encoding is illustrated in the following table.

Table 2.8: LZ78 Algorithm encoding table.

Position	1	2	3	4	5	6	7	8	9
Character	A	A	B	C	B	B	A	B	C
Code	0A	1B		0C	0B	4A		4C	
Dictionary	A	AB		C	B	BA		BC	

The dictionary will not be sent to the decoder since the decoder can reconstruct the same dictionary in the same fashion.

The compression performance of LZ78 is close to LZ77 and LZSS. However, since a tree structure is utilized in LZ78 to store the dictionary, the search speed is improved.

Although LZ78 constructs the dictionary explicitly, the way it constructs the dictionary is similar to that of LZ77 in that they both use the symbols that have been coded. Therefore, randomly accessing the compressed text is still impossible for LZ78.

2.15.4 LZW Algorithm

LZ78 algorithm presents the following problems:

1. Since the alphabet is a small finite set and the symbols from the alphabet frequently appear in the text, it might be better to store the alphabet in the dictionary at the very beginning.

2. Using code *"index, c"* may not be efficient if *c* is frequently the first symbol of the next match. Let $S=c1c2c3 \ldots Cu$ be the uncompressed text of length *u* over alphabet $\Sigma = \{a1, a2, a3, \ldots, aq\}$, where *q* is the size of the alphabet. We denote the LZW compressed format of *S* as *S.Z* and each code in *S.Z* as *S.Z*[*i*], where $ni \leq\leq 1$.

The LZW compression algorithm [Welch84] uses a tree-like data structure called a "tree" to store the dictionary generated during the compression processes. Each node on the tree contains:

- A node number, which is a unique ID in the range of [0, *n*+*q*] and,
- A label, which is a symbol from the alphabet Σ.
- A chunk, which is defined as the string on the path from the root to the node.

At the beginning of the compression, the tree has *q*+1 nodes, including a root node with node number 0 and a *NULL* label and *q* child nodes each labeled with a unique symbol from the 'alphabet. During the compression, LZW algorithm scans the text and finds the longest substring that appears in the tree as the chunk of some node *N* and outputs *N* to *S.Z*. The tree then grows by adding a new node under *N* and the new node's label is the next unencoded symbol in the text. Obviously, the new node's chunk is a node *N*'s chunk appended by the new node's label. Figure 2-12 illustrates the tree structure.

S=a abbaab b ab c c c c c c;

S.Z = 1,1,2,2,4,6,5,3,11,12;

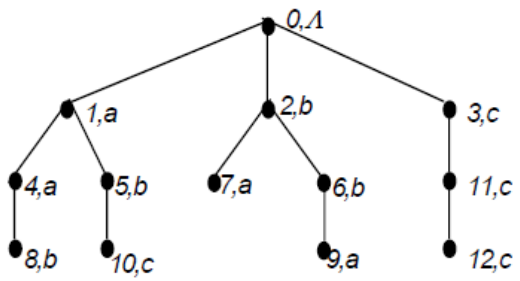

Figure 2.10 LZW tree example

The decoder constructs the same tree and uses it to decode S.Z. Both the compression and decompression (and thus and tree construction) can be done in time O (u).

The LZW tree can be reconstructed from S.Z in time O (n) without explicitly decoding S.Z in the following manner [5]: When the decoder receives a code S.Z [I], assuming S.Z [I-1] has already been received in the previous step (ni≤≤2), a new node is created and added as

a child of node S.Z [I-1]. The node number of the new node is i-1+q and the label of the new node is the first symbol of node S.Z[i]'s chunk.

A popular compression utility that is based on LZW is COMPRESS.

Amir [Amir96] proposed an algorithm to implicitly decode the text and build the dictionary for compressed pattern matching. (TAO TAO.1994)

J. Ziv and A. Lempel, in 1977 and 1978, presented several papers describing the mathematical foundation for a universal data compression algorithm, which is based on the idea of an incremental parser. Welch later extended this idea of the greedy incremental parser to describe a dictionary-based compression scheme that he labeled the LZW method. Later, developers of the UNIX operating system enhanced Welch's work to create a practical file compression/decompression utility.

The LZW method is a single pass algorithm, which is extremely adaptable, and hence requires no a priori knowledge of the source. In most cases, it provides a reasonable compression for a diverse range of input streams. This results from LZW's ability to make use of redundancy in character repetition, like RLE. And, at the same time, it also makes use of character frequency redundancy and string pattern redundancy.

However, this algorithm is not very effective for the compression of data that contains significant positional redundancy. LZW has typical compression ratios for English text, and program source code, which are around 2. Huffman coding and Arithmetic coding are also capable of producing similar results, although not as flexible. Since LZW takes from the best of both worlds, that is to say, it has the speed, and simplicity of Huffman while preserving the adaptive nature of the PPM model for the Arithmetic coding algorithm. (Lempel Ziv Welch 1998)

2.16 WinRAR compression

WinRAR uses a proprietary compression implementation developed by Eugene Roshal. This implementation includes several well-known compression algorithms such as Lempel-

Ziv-Storer-Szymanski (LZSS), PPM with Information Inheritance (PPMII), Intel IA-32 and delta encoding. These methods will be discussed in detail below. (Kristine Arthur-Durett 2014).

2.16.1 LZSS

LZSS is the primary compression method for WinRAR. It is a lossless data compression algorithm derived from LZ77. LZSS is a dictionary coding technique that utilizes previously seen the text as a dictionary. A string of symbols, S, is replaced by pointers to substrings of S in the dictionary along with the length of the substring. The pointers are original if they point to a substring of the original source. Similarly, a compressed pointer references the compressed representation. The references can be either left or right-pointing and the scheme allows for recursion. (Kristine Arthur-Durett 2014).

Storer and Szymanski's scheme addresses a flaw in the original LZ77 algorithm. LZ77 would occasionally generate a reference longer than the target string, resulting in poor compression. To correct this, LZSS omits references that are longer than a specific point. This scheme also uses one-bit flags to indicate whether the following string of data is the original source or a reference. (Kristine Arthur-Durett 2014).

2.16.2 PPMII

PPMII was integrated into WinRAR as of version 2.9 to further reduce compression ratios. PPMII was developed by Dmitry Shkarin as an improvement to the Prediction by Partial Matching model. Broadly, the n^{th} symbol of a string is predicted based on the previous $n-1$ symbols. The compression of a string is defined by code conditional probability distributions and based on the following assumption:

The larger the common (initial) part of contexts s, the larger (on the average) the closeness of their conditional probability distributions.

This notes that the greater number of common characters two strings have, the greater the probability of predicting the N^{th} symbol. This is desirable as a higher probability requires

fewer bits to encode. To efficiently store the contexts, a M−ary tree is utilized. This is particularly efficient if a text consists of large numbers of short strings. (Kristine Arthur-Durett 2014).

2.16.3 Intel IA-32

Intel IA-32 is a compression scheme introduced in response to the observation that database processing correlates with the hardware constraints of storage I/O. It provides lightweight compression and decompression using single instruction, multiple data (SIMD) command to optimize database queries. Data is compressed quickly by reducing the dynamic range of data. This is an accomplished by applying a mask, packed shift, and finally stitching the data together. (Kristine Arthur-Durett 2014).

2.16.4 Delta encodes

This is the second new technique introduced to optimize compression performance in the new version. Delta encoding encompasses several techniques that stores data as the difference between successive samples. This is an alternative to directly storing the samples themselves. Generally, the first value in the encoded file is equal to the first value in the original data. The subsequent values are equal to the difference between the current and the previous value in the input. That is, for an encoded value YN with original inputs XN:

$$YN = XN - xn-1$$

This approach is best suited when the values in the original file have only small changes between adjacent symbols. It is, therefore, ideal for file representation of a signal, but performs poorly with text files and executable code. (Kristine Arthur-Durett 2014).

2.17 Literature Survey

Most of the published articles are concerned with the description of some software tools designed and built to perform Steganography on some text, image, and audio cover media. The publication about the scheme of the stego system is very primitive and mostly does not

offer a key solution for some weak aspects which may face the discussed (proposed) system. Among a large number of published articles, the following table shows the current researches being done, in comparison to this research:

Table 2.9: Shows Current Research being done:

Table 2.9: Shows Current Research being done (Aos, A.Z.Ansaef 2009)

Date	Author	Title	Description	Published
2001	Andre	Zero knowledge Watermark Detection and Proof of Ownership	The goal is; allow prove to soundly convince a verifier of the presence of a watermark in certain stegodata without revealing any information, which the verifier can use t remove the watermark. In this paper, they define zero-knowledge watermark detection precisely. Then they propose efficient and probably secure zero-knowledge protocols for watermarking scheme.	Journal of Computer Science
2002	S. Joshua and C. Barrett	Introduce Several Spread Spectrum Data-Hiding Methods	Introduce several spread spectrum data-hiding methods. These techniques use the message data to modulate a carrier signal, which is then combined with the cover image in section of non overlapping blocks. The message is extracted via cross correlation between the stego image and the regenerated carrier; hence, cover image escrow is not necessary. A threshold operation is then performed on the resulting cross correlation to determine the binary value of the embedded data bits.	IEEE And International Conference
2005	Ross	El.proposed the Steganogra-phic File System	This is a storage mechanism designed to give the user a very high level of protection against being compelled to disclose the file content. It will deliver a file to any user whose name and password is known, but an attacker who does not possess this information and cannot guess it, can gain no information about whether the file is present, even given complete access to all the hardware and assumptions.	Journal of Computer Science And IEEE
2005	N. Al_Mayyahee	New Robust Information Hiding Technique	The research aim's to hide small color image inside another bigger color image. It uses the transform domain in the steganography process to increase its robustness against the changes and treatment it's done for the cover image. The research depends on substituting the similar block of the embedded image within a cover image The system includes six stages: • Test stage, • Transformation stage, • Matching stage, • Substituting stage, • Inverse transformation stage • Key generating-stage. The proposed system is a secret key steganography system, where The key is secret between the two parties and stored inside the stego image to increase the quality of the system. The proposed system gives good results in imperceptibility, security, robust, capacity and quality	International Conference Listed of IEEE and ISI Thomson

2006	M. Al-Hamami	Information Hiding Attack in Image	The aim of this research is to analyze the hidden information and not only detect it. Analysis means detection, extraction or distortion of the secret message. The proposed system is designed to deal with analysis tools and to process in three stages: Detection, extraction and distortion. In detection process the intention will be on: • Statistical analysis or tests. • Visual pixel detection. The suggestion that using statistical test as a tool for Detection. These tests are: Average Absolute Difference (AAD), Mean Square Error (MSE), Laplacian Mean Squared Error (LSME), Signal-To-Noise Ratio(SNR), Peak Signal-To-Noise Ratio (PSNR),Normalized cross-Correlation (NCC), Correlation Quality (CQ), Histogram Similarity (HS),and Chi-Square Test.	Journal of Computer Science And International Conference
2006	S. Abdullah	Presents a Method to Hide Small Arabic Texts in Two Cover Types	Presents a method to hide small Arabic texts in two cover types: the first cover is another Arabic text, where the embedding depends on the natural feature of Arabic, the second cover is an image, and the process uses three methods: hiding by modulo 2 of LSB block, hiding by modulo 2 with encryption and hiding in blue channel of pixel.	Academy Publisher And Journal of Computer Science
2008	AOS.A.Z. Ansaef	Securing Cover-File of Hidden Data Using Statistical Technique and AES Encryption Algorithm	The aims of this research is to study the different types of steganography systems design and implementation of steganography system which embeds information in an .EXE files, The system tries to find a solution to the size of the cover file and making it undetectable by anti-virus software. The system includes two main functions; first is the hiding of the information in a PE-file (.EXE file), through the execution of four process (specify the cover file, specify the information file, encryption of the information, and hiding the information) and the second function is the extraction of the hiding information through three process (specify the steno file, extract the information, and decryption of the information).The system has achieved the main goals, such as, make the relation of the size of the cover file and the size of information independent, and the result file does not make any conflict with anti-virus software.	Two International Conference Listed of IEEE and ISI Thomson And Local Conference

2.18 Conclusion

The hurried development of multimedia and internet allows wide distribution of digital media data. It becomes much easier to edit, modify and duplicate digital information. In additional, a digital document is also easy to copy and distribute, therefore it may face many threats. It became necessary to find an appropriate protection due to the significance, accuracy, and sensitivity of the information. Nowadays, protection system can be classified into more specific as hiding information (Steganography) or encryption information (Cryptography) or a combination of them. Cryptography is the practice of 'scrambling' messages so that even if detected, they are very difficult to decipher. The purpose of Steganography is to conceal the message such that the very existence of the hidden is 'camouflaged'. However, the two techniques are not mutually exclusive. Steganography and Cryptography are in fact complementary techniques. No matter how strong algorithm, if an encrypted message is discovered, it will be subject to cryptanalysis. Likewise, no matter how well concealed a message is, it is always possible that it will be discovered. By combining Steganography with Cryptography, we can conceal the existence of an encrypted message. In doing this, we make it far less likely that an encrypted message will be found. Also, if a message concealed through Steganography is discovered, the discoverer is still faced with the formidable task of deciphering it. Also, the strength of the combination between hiding and encryption science is due to the non-existence of standard algorithms to be used in (hiding and encryption) secret messages.

Also, there is randomness in hiding methods such as combining several media (covers) with different methods to pass a secret message. Furthermore, there is no formal method to be followed to discover a hidden data.

2.19 Related works

Section two in this chapter presents the related work in image steganography both LSB techniques and pixel intensity techniques and present compression table between them in the latter part of this section.

In Dr. (AL-NAJJAR 2008) presented “Multilevel Digital Multimedia Steganography Mode”. This paper focuses on two folds: to develop an abstract multilevel model and to illustrate the model by hiding text represented using a black and white image into a gray decoy image and then into a color image in the RGB format. Four objects are defined, the message object (M), the intermediate object (I), the cover object (C), and the stego-object (S). The elements of M are given by the set {M} of size |M|, similarly, {I} of size |I| and so forth.

The message {M} is passed through the transformation T1 that can include many possibilities. It can be compression, private-key or public-key encryption, or a combination of techniques, as required by the particular application. The same can be said about the other transformations T2 and T3. Figure 2.16 demonstrates the proposed model.

Embedding and recovery are controlled by the embedding/recovery function pairs f/g. The embedding function from {M} to {I} (or {D}) and from {D} to {C} can be different, improving one or more of the three steganography attributes: Capacity, Robustness, and Transparency.(Dr. AL-NAJJAR 2008)

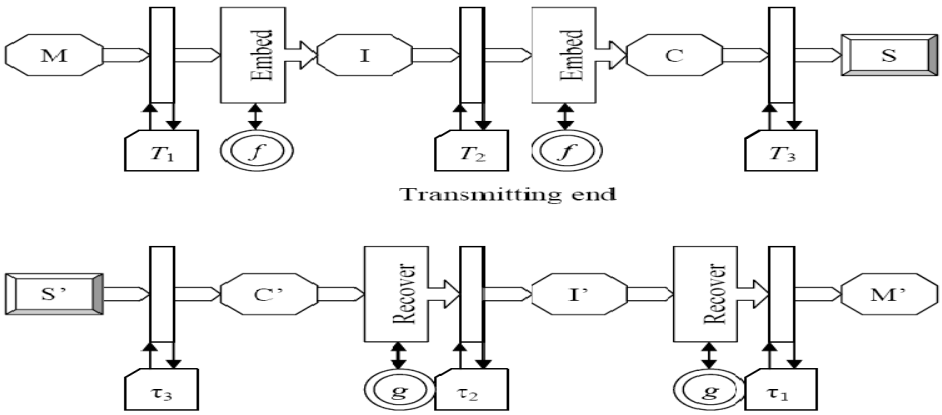

Figure 2.11: multi-level steganography model .(Dr. AL-NAJJAR 2008).

Comparative analysis is made to demonstrate the effectiveness of the proposed methods. The effectiveness of the proposed methods has been estimated by computing Mean square error (MSE) and Peak Signal to Noise Ratio (PSNR) and processing time.

The algorithm of steganography is divided into two section, section one focus in embed the text message using LSB steganography, this section have six steps, in step one read the cover image and text message which is to be hidden in the cover image, in step two convert the color image into grey image, after that in step three convert text message in binary then in step four calculate LSB of each pixels of cover image and in step five Replace LSB of cover image with each bit of secret message one by one , finally in step six Write stego image.

The section two in steganography algorithm is retrieved text message from stegoimage, this section have eight steps, in step one read the stego image, and then in step two calculate LSB of each pixels of stego image, after that in step three retrieve bits and convert each 8 bit into character and the cover image is broken into 8×8 block of pixels and then in step four working from left to right, top to bottom subtract 128 in each block of pixels after that in step five DCT is applied to each block, in step six each block is compressed through quantization table, additionally in step seven calculate LSB of each DC coefficient and replace with each bit of secret message, Finally in step eight write stego image.(Gurmeet Kaur)

In another proposed algorithm is embedded text message based on DCT steganography in this algorithm in step one is read cover image and in step Two read the secret message and convert it to binary.

The analysis of LSB based and DCT based steganography has been done based on parameters like PSNR, MSE, Processing time, security. PSNR computes the peak signal to noise ratio, in decibels, between two images. This ratio is used as a quality measurement between two images. If PSNR ratio is high then images are best of quality (Gurmeet Kaur)

Table 2.10: Simulation results for LSB & DCT Method

Method	PSNR	MSE	PROCESSING TIME	SIZE OF COVER IMAGE
LSB	51.1 109	0.50 35	0.133777 Seconds	256x256
LSB	51.1 109	0.49 93	0.084754 seconds	256x256
DCT	40.6735	5.56 84	1.0140 seconds	256x256
DCT	39.3 983	7.4687	1.3260 seconds	256x256

In this paper analysis of LSB & DCT methods has been successfully implemented and results are delivered. The MSE and PSNR of the methods are also compared and this paper presented a background discussion and implementation of the major algorithms of steganography deployed in digital imaging. From the results, it is clear that as PSNR in LSB is the best, but as we know that security is much more important in today's communication system. So security wise DCT is the best.

(Souvik Bhattacharyya 2011) is presented "Data Hiding through Multi Level Steganography and SSCE". They proposed that a steganography model combining the features of both text and image based steganography technique for communicating information more securely between two locations. The authors incorporated the idea of a secret key for authentication at both ends in order to achieve a high level of security.

As a further improvement of security level, the information has been encoded through SSCE values and embedded into the cover text using the proposed text steganography method to form the stego text. This encoding technique has been used at both ends in order to achieve a high level of security. Next, the stego text has been embedded through PMM method into the cover image to form the stego image. At the receiver side, the different reverse operation has been carried out to get back the original information. (Souvik Bhattacharyya)

The input message is first encoded through SSCE (Secret steganography code for embedding) values and embedded into the cover text using the proposed text.

steganography method. This encrypted message generates the secret key. The encrypted message is then embedded in the cover text using the mapping techniquemethod to form the stego text which in turn embedded into the cover image through PMM (Pixel Mapping Method) to form the stego image and transmit to the receiver side. At the receiver side, the stego image will be tested first for a specific feature. If that feature matches, the extraction process starts by extracting the stego text from the stego image. Next, the stego text goes through the text extraction and decryption method and finally the receiver may be able to see the embedded message with the help of same secret key generated at the sender side.

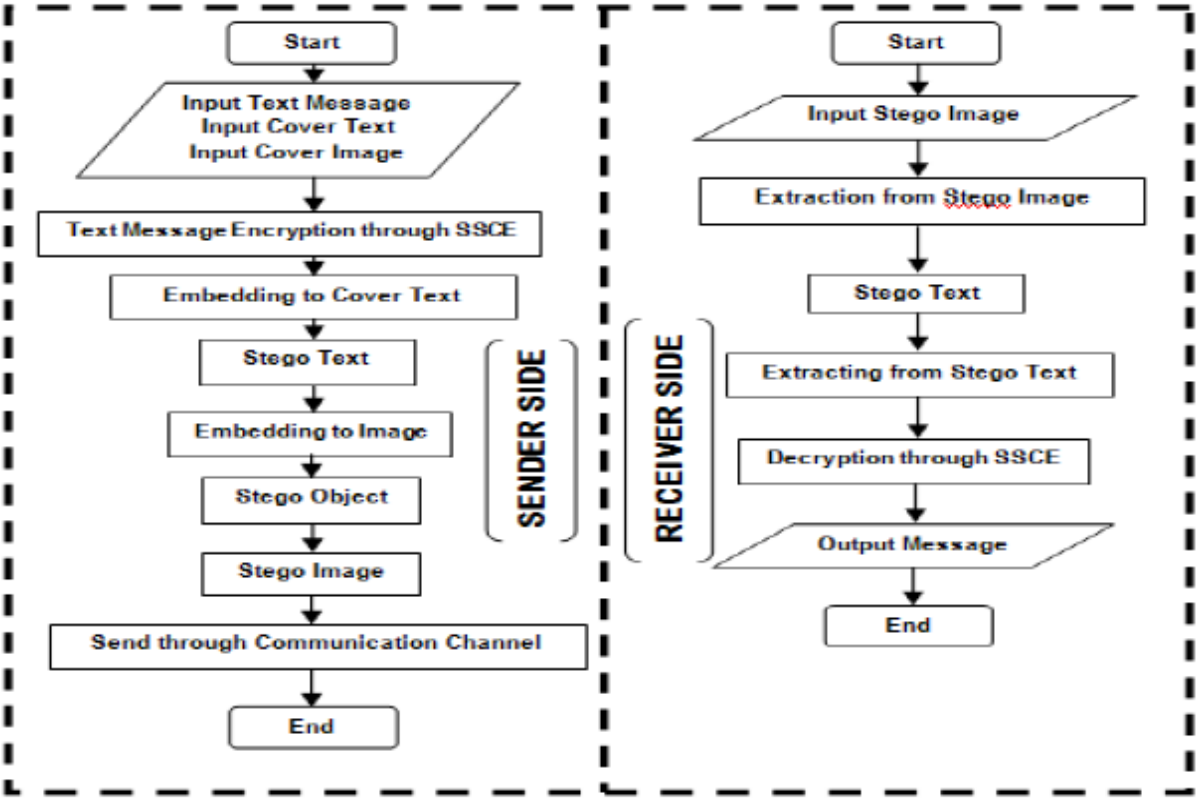

Figure 2.12: proposed algorithm for the steganography model (Souvik Bhattacharyya.2011).

In (Mohammad Tanvir Parvez)presents a new algorithm for RGB imagebased steganography. The algorithm introduces the concept of storing a variable number of bits in each channel (R, G or B) of pixel based on the actual color values of that pixel: lower color component stores thehigher number of bits. Our algorithm offers very high capacity for cover media compared to other existing algorithms.

The proposed is splitting pixel value to three channels (Red, Green, and Blue)

Use one of the three channels as the indicator. The indicator sequence can be made randomly, based on a shared key between sender and receiver, and then in the embedding process, the data is stored in one of the two channels other than the indicator. The channel, whose color value is lowest among the two channels other than the indicator, will store the data in its least significant bits. Instead of storing a fixed no of data bits per channel, no of bits to be stored will depend on the color value of the channel. The lower value higher data bits to be stored. Therefore, a partition of the color values is needed. Through experimentations, we show that optimal partition may depend on the actual cover image used. To retrieve the data in this algorithm, we need to know which channel stores the data bits. This is done by looking at the least significant bits of the two channels other than the indicator, if the bits are same, then the channel following the indicator in cyclic order stores the data, otherwise, the channel which precedes the indicator in cyclic order stores the data. The Flowing is Flow charts explain the encoding and decoding parts of this algorithm:

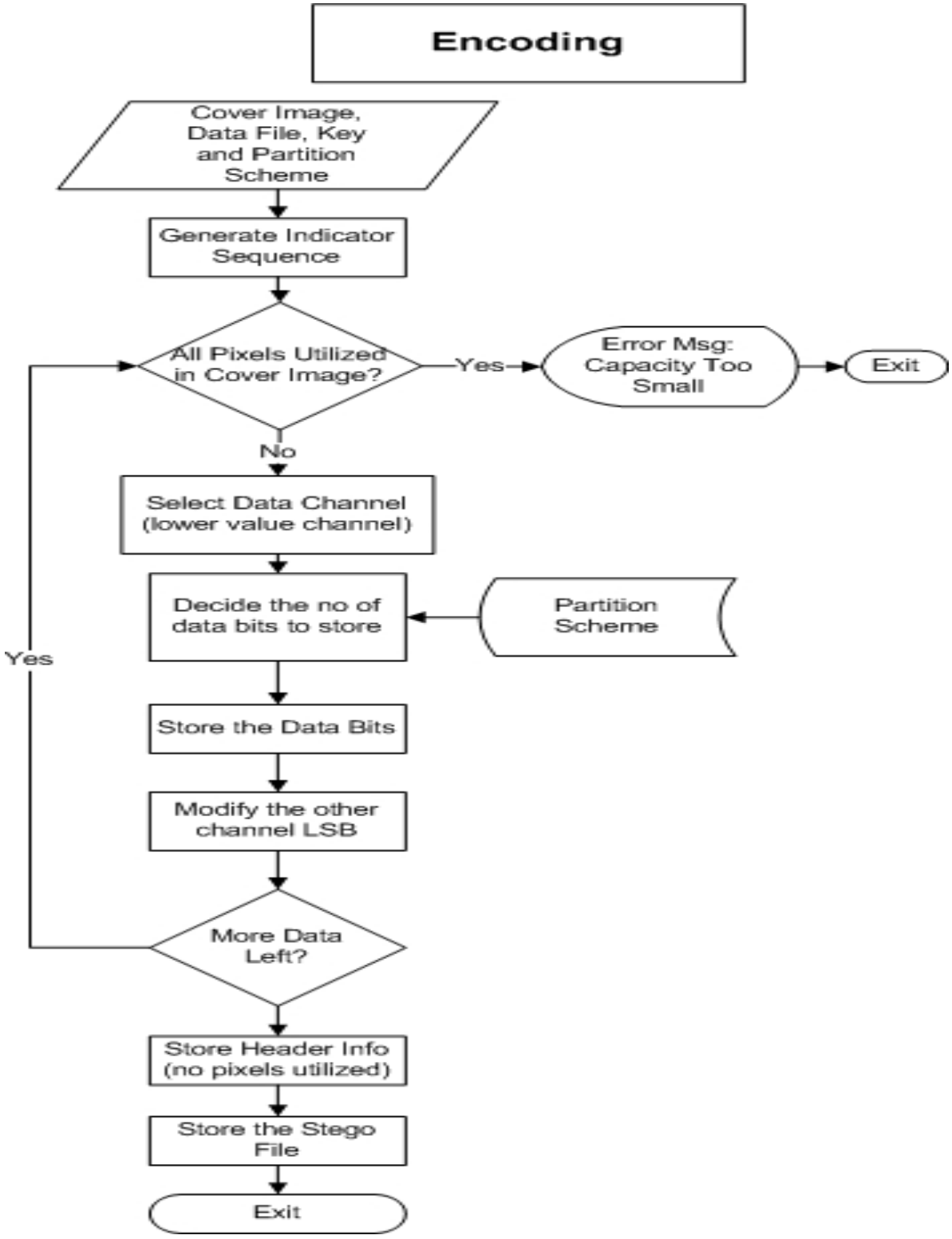

Figure 2.13: Flow charts of the encoding part of the algorithm(Mohammad Tanvir Parvez 2009).

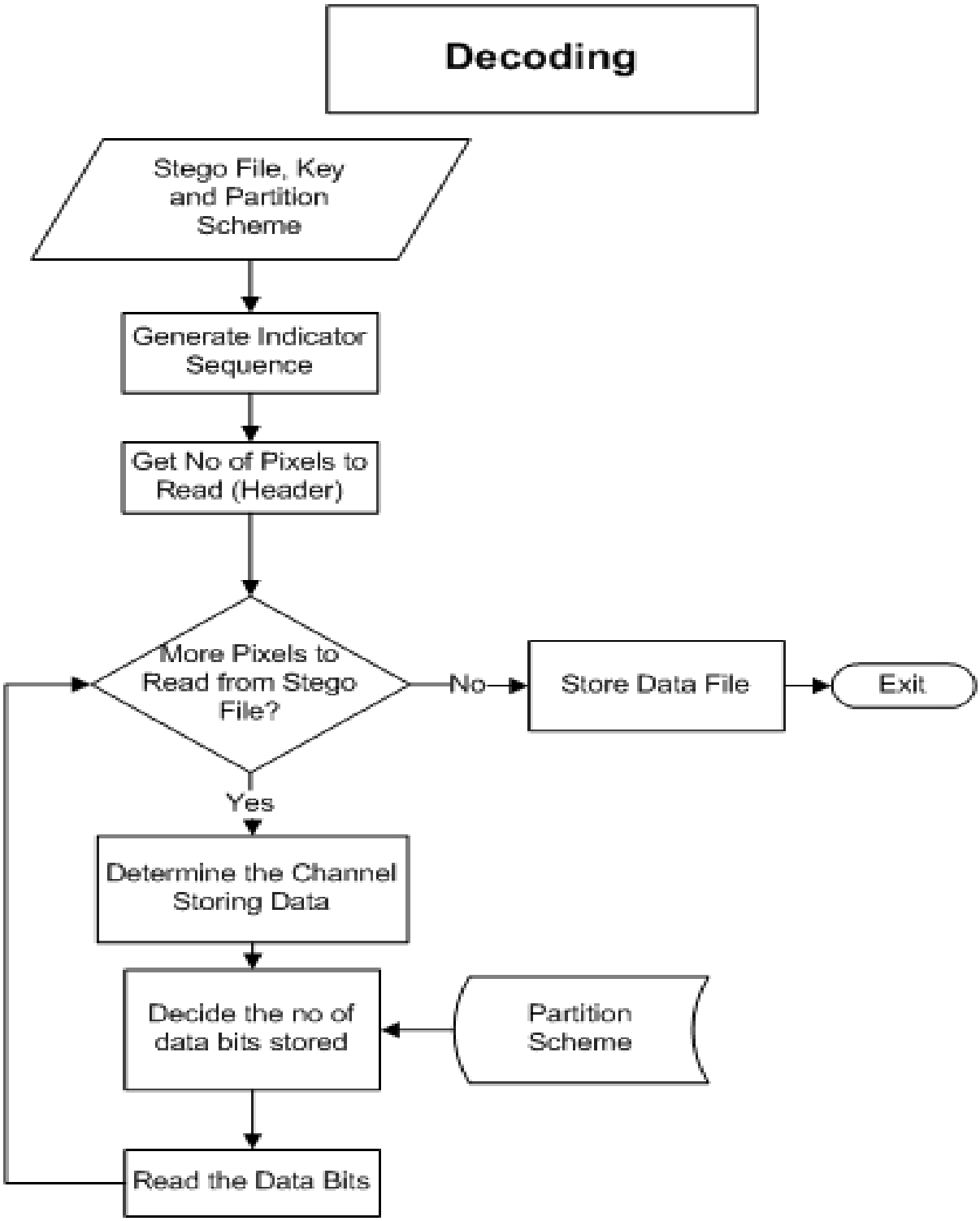

Figure 2.14: Flow charts of the decoding part of the algorithm (Mohammad Tanvir Parvez 2009).

Table 2.11: the summarization of related work.

Paper name	Message Object	Number of Levels	Cover Object	Techniques
The Decoy:Multilevel Digital Multimedia Steganography Model	Text Represented by black and white images	Two– Levels	A Grayscale image and RGB image	LSB in both Levels
A Steganography Implementation based on LSB & DCT	Text	One– Level	Image	LSB & DCT
Hiding through Multi Level Steganography and SSCE	Text message	Two-levels	Image	PMM (Pixel Mapping Method) Inserting non-specific or non-particular nouns in English
RGB Intensity Based Variable-Bits Image Steganography	Text Message	One– Level	Image	Pixel intensity

CHAPTER THREE

PROPOSED SYSTEM ANALYSIS AND WORK ENVIRONMENT

CHAPTER THREE

Proposed System Analysis and Work Environment

3.1 Overview

This chapter describes the proposed method (Multilevel Image Steganography Using compression techniques) and explains the diagrams that clarify the proposed method. In level one modified Least Significant Bit (secure LSB-L1) Image steganography BMP image is used as a cover Image with a secure data (text) converted to long bit-stream before Hiding, while level two (secure LSB-L2) another BMP image is used as a cover image with a secure data also converted to long bit-stream before canceling.

The programming language will be used in the implementation of the two levels (Level one and level two) is a C# programming language.

3.2 Proposed Method

The proposed method uses multilevel image steganography (two levels) level one will be done by embedding the secret message (text) into the cover image (cover one) which is a colored image (BMP image) using Least Significant Bit (LSB) image steganography. And then using key to private message, and finally using compression techniques to compress output result that comes from level one

In level two improve the LSB scheme. It overcomes the sequence-mapping problem by embedding the message into a set of random pixels, which are scattered on the cover image. Figure 3.1 below explains the general overview of the proposed method (embedding process)

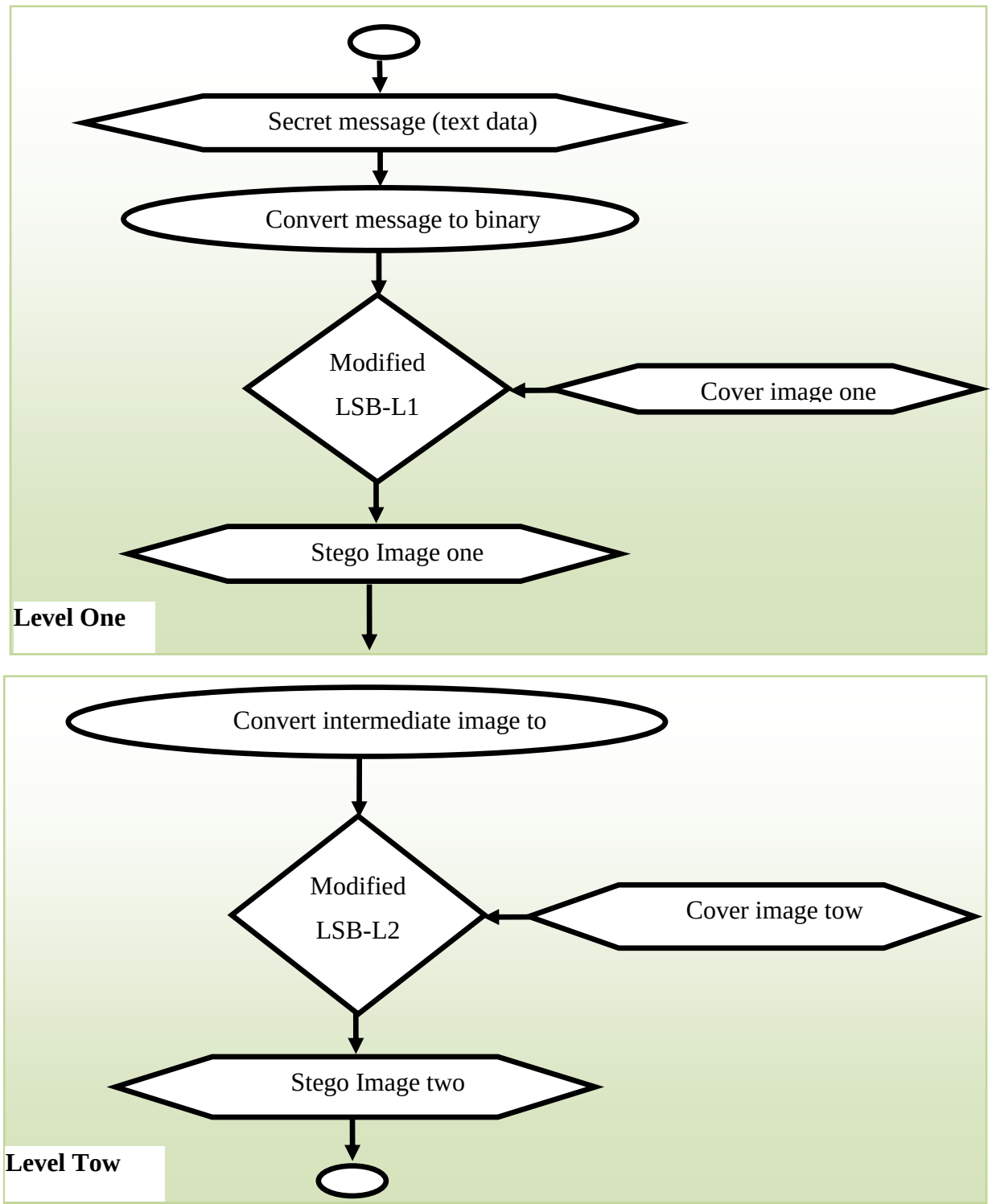

Figure 3.1: the proposed method.

3.3 The algorithm process

3.3.1 Level one process

Firstly, writes the length of the message in bytes into the first pixel. After that, it reads a byte from the message, reads another byte from the key, and calculates the coordinates of the pixel to use for the message-byte. It increments or resets the color component index, to switch between the R, G, and B component. Then it replaces the R, G or B component of the pixel (according to the color component index) with the message-byte, and repeats the procedure with the next byte of the message.

The key stream in level one steganography is being used to pseudo-randomly select the pixels and encrypts message bytes. Table 3.1 is a summary about hiding:

Table 3.1 is a summary about hiding

For all bytes in the message stream
Read a byte from the key stream , store in the current key.
Read a byte from the message stream.
XOR these bytes, store result in current byte.
Move key pixels to the right, move down if necessary.
Get the color of the pixel.
Set the R, G or B values to current Byte.

To extract a hidden message from a bitmap, open the bitmap file and specify the password or key you used when hiding the message. Then choose a file to store the extracted message in (or leave the field blank, if you only want to view hidden Unicode text), and click the *Extract* button. The application steps through the pattern specified by the key and extract the bytes from the pixels. At last, it stores the extracted stream in the file and tries to display the message. Don't bother about the character chaos, if your message is not a Unicode text. The data in the file will be all right. This works with every kind of data, you can even hide a small bitmap inside a larger bitmap. Table 3.2 is a summary about extracting:

Table 3.2 is a summary about extracting

For all expected bytes of the message
Read a byte from the key stream.
Calculate the position of the next pixel.
Get the color of the pixel.
Write the values of R, G or B into the message stream.

3.3.1.1 Steps Embedding Process in level one using Modified LSB (secure LSB-L1)

1. Get streams containing the message.
2. Get streams containing the key (just another sequence of bytes).
3. The process starts with writing the length of the entire message into the first pixel. We'll need this value before extracting the message later on.
4. XOR Message Byte with Key byte.
5. Check! Make sure we're not going to run out of pixels before the entire message is hidden. Between steps 2 and 3 the program executes some checks. It walks through the pixels without hiding stuff, just to make sure we're not going to run out of pixels before the entire message is hidden.
6. Reads a byte from the key stream to calculate the next position (start with the second pixel).
7. Jump k to the next pixel.
8. Get the pixel and put the message-byte into one color component.
9. IF all bits embedded?
    - **A.** Go step 10 end and show stego-image.
    - **B.** Else Return Step 6 read bytes from the key stream.
10. Display result stego-image.

Figure 3.2 below Shown and explain Steps Embedding Process in level one.

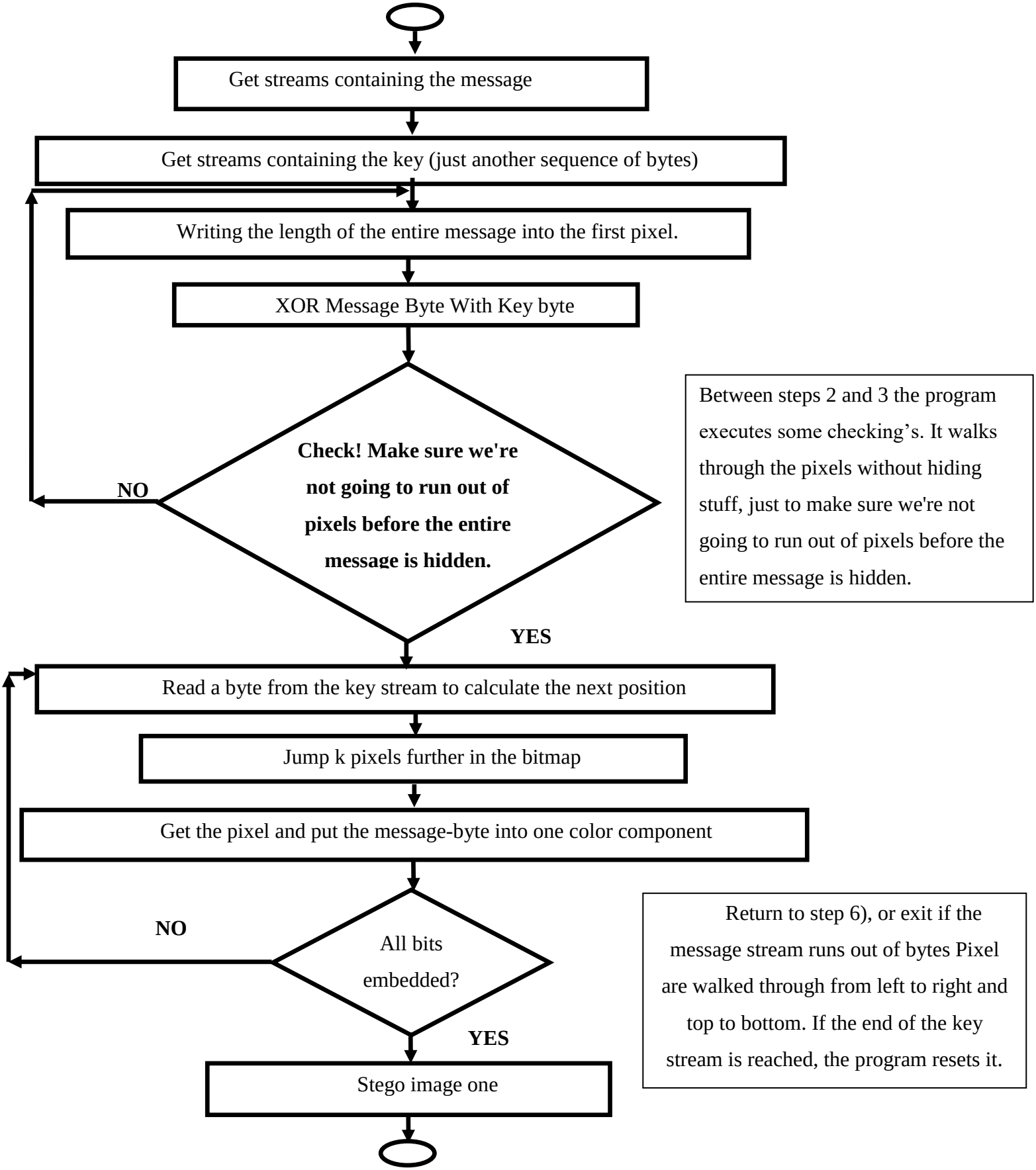

Figure 3.2: Steps Embedding Process in level one in details.

3.3.1.2 Steps extracting Process in level one using Modified LSB (secure LSB-L1)

1. Read stego image (intermediate image).
2. If the method runs in extraction mode, it reads the message's length and the color components instead of setting it. Here is how to get the length from the first pixel:
   //UnTrimColorString fill the String with '0'-,chars‘ to match the specified length
3. Read a byte from the key stream.
4. Calculate the position of the next pixel.
5. Get color of the modified pixel.
6. Extract the hidden message-byte from the color.
7. Rotate color components.
8. Proceed to next byte.
9. IF all bytes extracted?
   - **A.** Yes! Go step 10 display Result.
   - **B.** Else Return Step 3.
10. Display Result Text Message.

Figure 3.3: below Shown and explain Steps extracting Process in level one

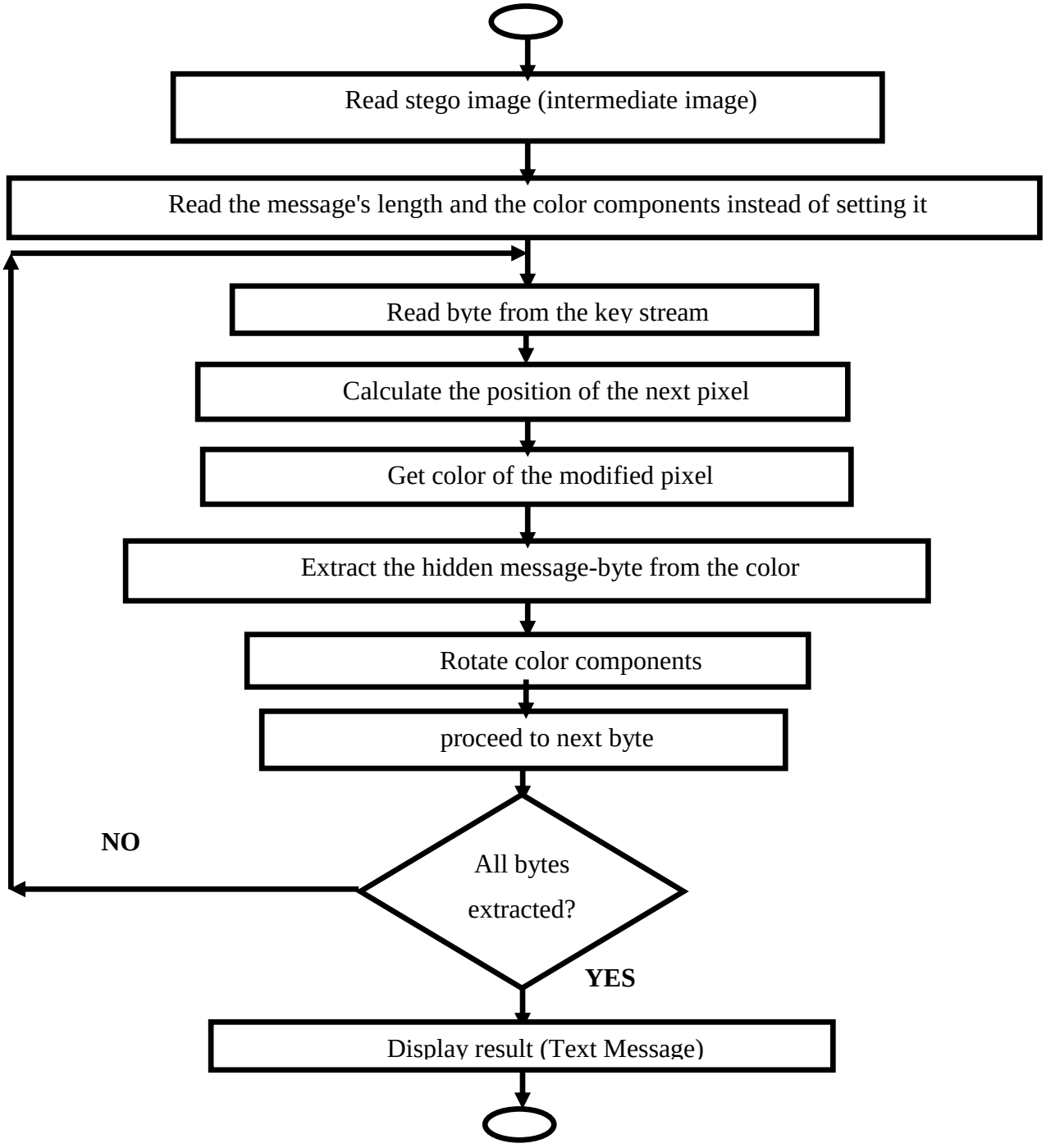

Figure 3.3: extracting Process in level one

3.3.2 Spread the information over the images

Hide one message in many bitmaps. It is quite similar to writing text across a couple of pages. It means spreading the pixels over multiple images. Figure 3.4 below Shown and more explain:

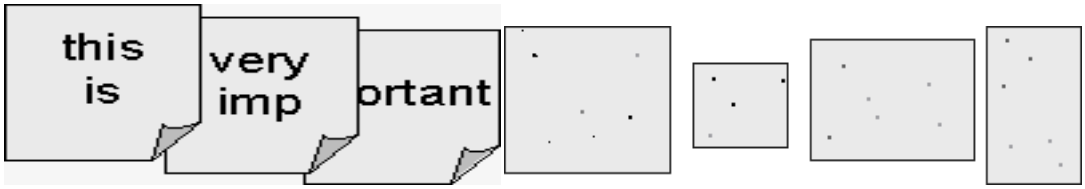

Figure 3.4: Spread the information over the images

You can send each image in a separate E-mail, post them in different mailboxes, or store them on different discs. The GUI allows selecting carrier bitmaps the same way as selecting key files. The selection is stored as an array of Carrier Images.

Larger images can hide more bytes (more pixels) than smaller images.

3.3.2.1 Process of carrier unit

Now, I start with the first carrier bitmap, loop over the message, hide a number of bytes, and switch to the second carrier bitmap, and so on.

Current position in the carrier bitmap Start with 1, because (0,0) contains the message length.

In the end, we must save the new images. Each image can be saved using a format (BMP).

3.3.3 Level Tow process

It can accept any type of image to hide information file, but finally, it gives the only "BMP" image as an output that has hidden file inside it. We use LSB bit for writing our security information inside BMP pictures. So, if we just use last layer (8th layer) of information, we should change the last bit of pixels, in other hands, we have 3 bits in each pixel so we have 3*high*width bits' memory to write our information. But before writing our data, we must write the name of the data (file), the size of the name of data & size of data. We can do this by assigning some first bits of memory (8th layer).

(00101101 00011101 11011100)

(10100110 11000101 00001100)

(11010010 10101100 01100011)

Using each 3 pixels of picture to save a byte of data

The algorithm used for Encryption and Decryption in this application provides using several layers lieu of using only LSB layers of the image. Writing data starts from the last layer (8th or LSB layer); because significant of this layer is least and every upper layer has doubled significant from its down layer. So every step we go to upper layer image quality decreases and image retouching transpires.

The encrypt module is used to hide information into the image; no one can see that information or file. This module requires any type of image and message and gives the only one image file in the destination.

The decrypt module is used to get the hidden information in an image file. It takes the image file as an output and gives two files at destination folder, one is the same image file and another is the message file that is hidden it that.

Before encrypting file inside the image, we must save the name and the size of the file in a definite place of the image. We could save file name before file information in LSB layer and save file size and file name size in most right-down pixels of the image. Writing this information is needed to retrieve the file from an encrypted image in decryption state.

The graphical representation of this system is as follows:

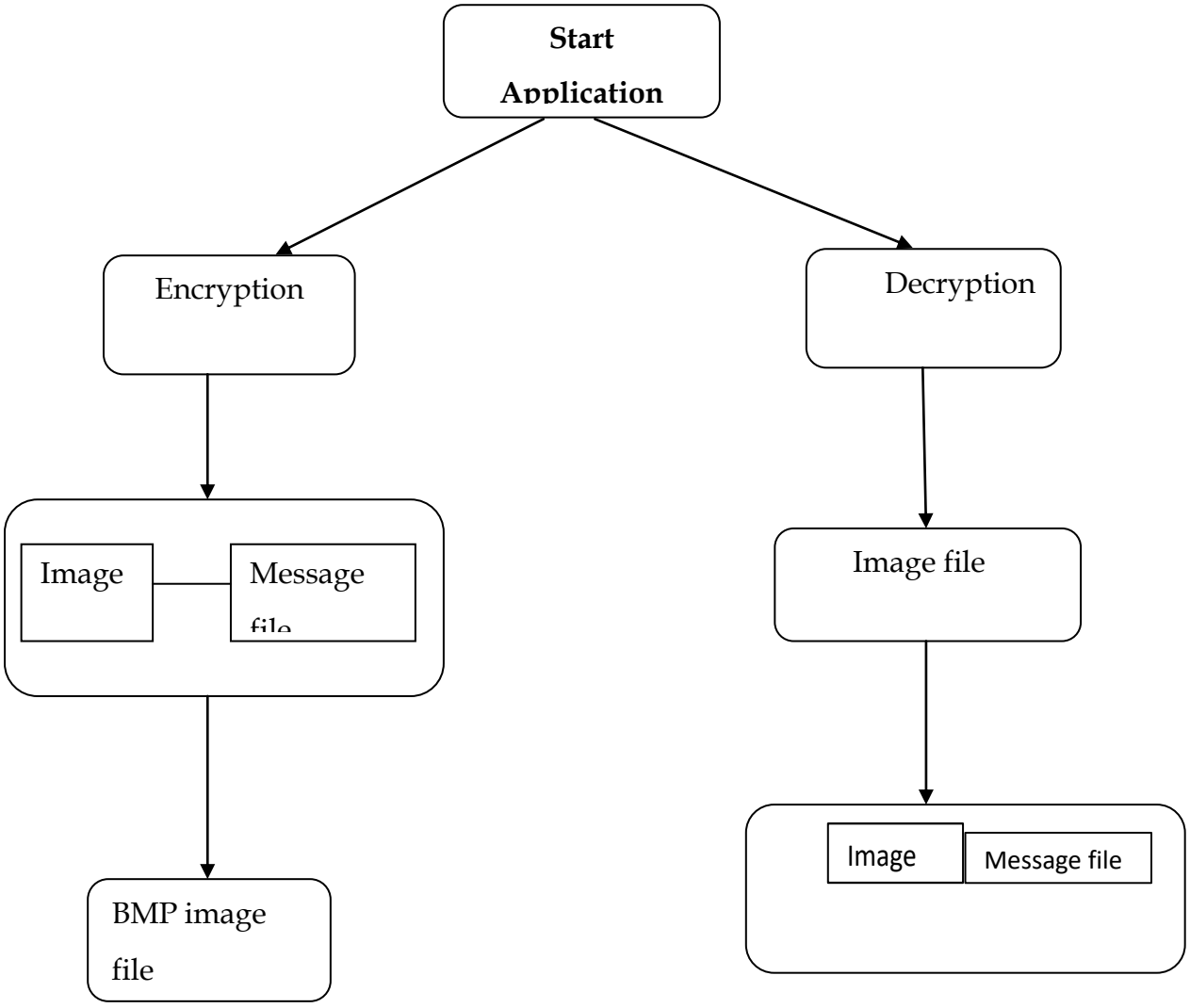

Figure 3.5: graphical representation the system.

3.3.3.1 Steps Embedding Process in level tow using Modified LSB (secure LSB-L2)

Before embedding file inside image writing this information because is needed to retrieve a file from an encrypted image in decryption state

1. Use any 3 pixels to input secret data.
2. Must write the name of the data (file).
3. Write Size of the name of data.
4. Write Size of data.
5. Embedding secret stego-media.
6. IF all bits embedded?
    - **A.** Yes, go Step 7.
    - **B.** Else Return Step 5.
7. Display Stego image Tow.

Figure 3.6: below Shown and explain Steps embedding process information about the file:

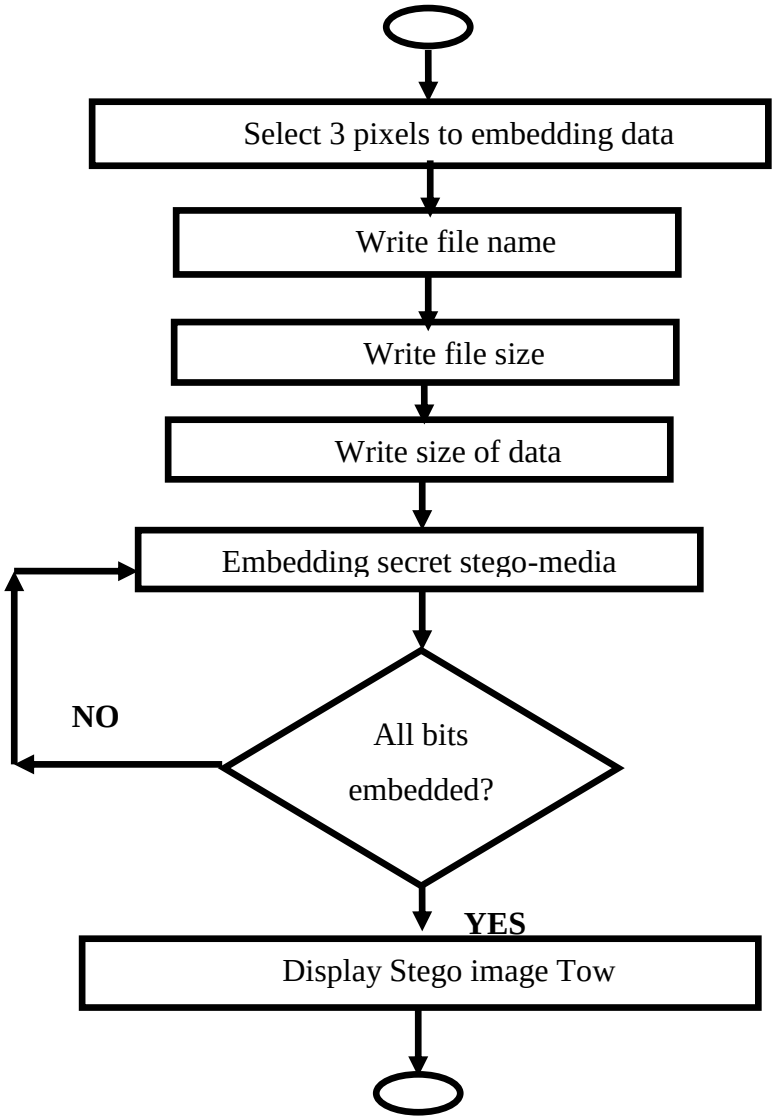

Figure 3.6: Steps embedding process information

3.3.3.2 Steps extracting Process in level Tow using Modified LSB (secure LSB-L2)

1. Read file name.
2. Read file size.
3. Read the size of data.
4. Read the data at Modified LSB.
5. IF all bits extracted?

A. Yes! Display Compression File and Go Step 6.

B. Return to step 4.

6. Save the file with name and size and data.

Figure 3.7: below Shown and explain Steps extracting Process in level Tow using (secure LSB-L2):

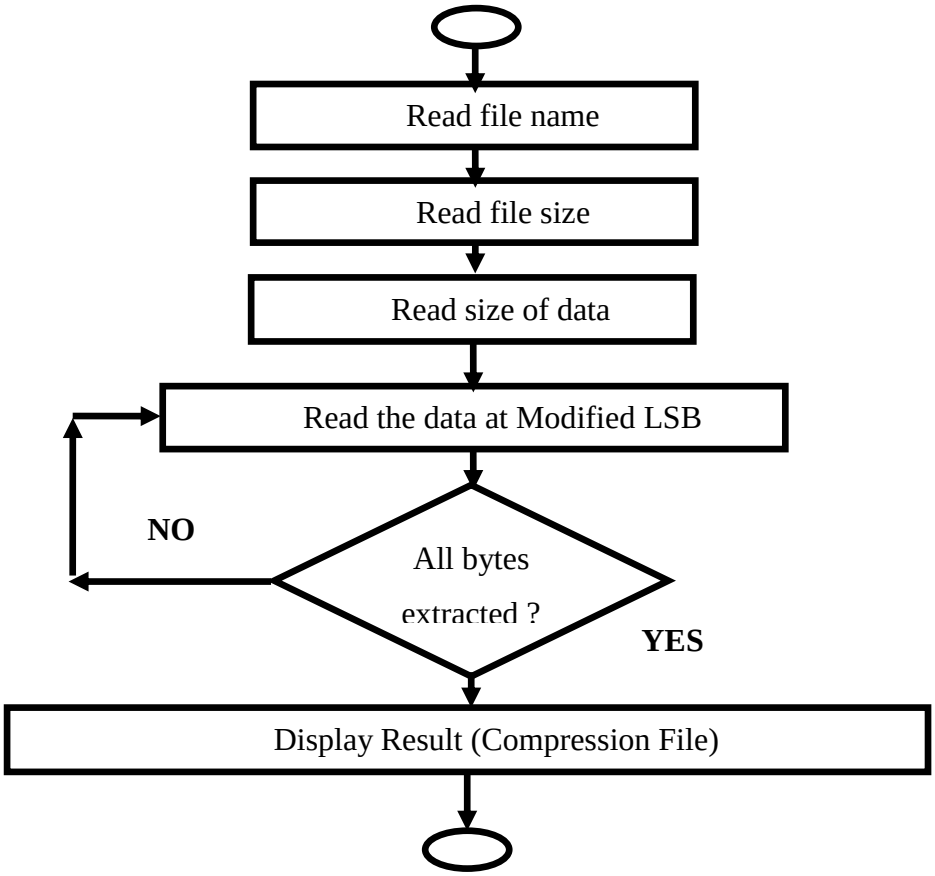

Figure 3.7: Steps extracting Process in level Tow using (secure LSB-L2)

3.3.4 Compression process Using Huffman algorithm

1. Initializing 511 of Huffman nodes by its ASCII values.
2. Calculates each ASCII frequency in the input buffer.
3. Sorts ASCII characters depending on frequency.
4. Constructs Huffman tree.
5. IF Constructs is finished?

A. Yes, Go Step 6 Compression ASCII code.

B. ELSE Return Step 4 Constructs Huffman tree.

6. Compression ASCII code.
7. Save Huffman tree nodes with its frequencies (encoding).

Figure 3.8: below Shown and explain Steps Huffman Compression algorithm:

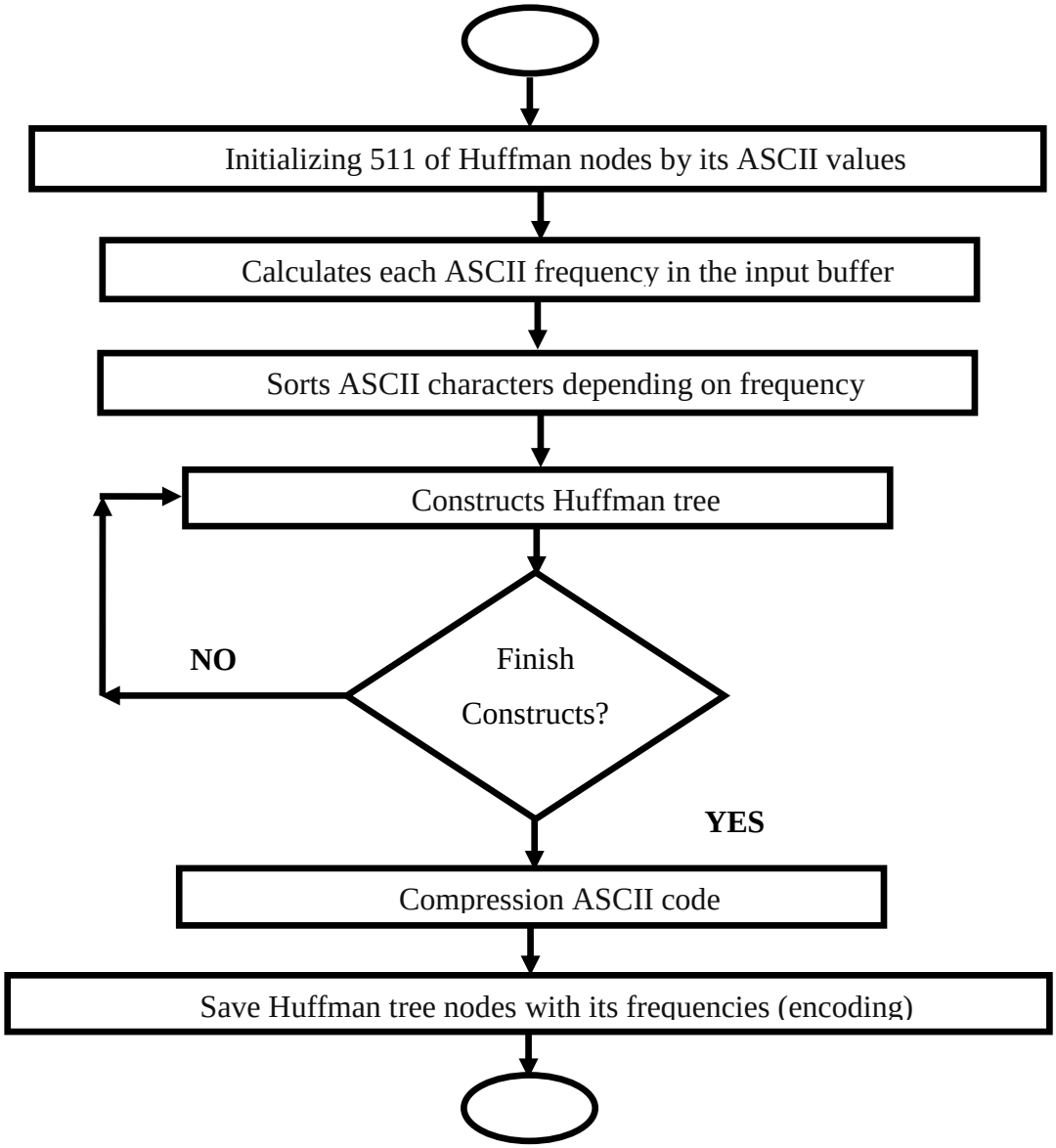

Figure 3.8: Huffman Compression algorithm process Steps.

3.3.5 Decompression Process Using Huffman algorithm

1. Read Construct Huffman tree.

2. Replace each code with its ASCII.
3. All replaced each code with its ASCII is done?
    - **A.** Got Step 4 Decompression ASCII code and save data.
    - **B.** Return Step 2 Replace each code with its ASCII.
4. Decompression ASCII code and save data.

Figure 3.9: below Shown and explain Steps Huffman Decompression algorithm:

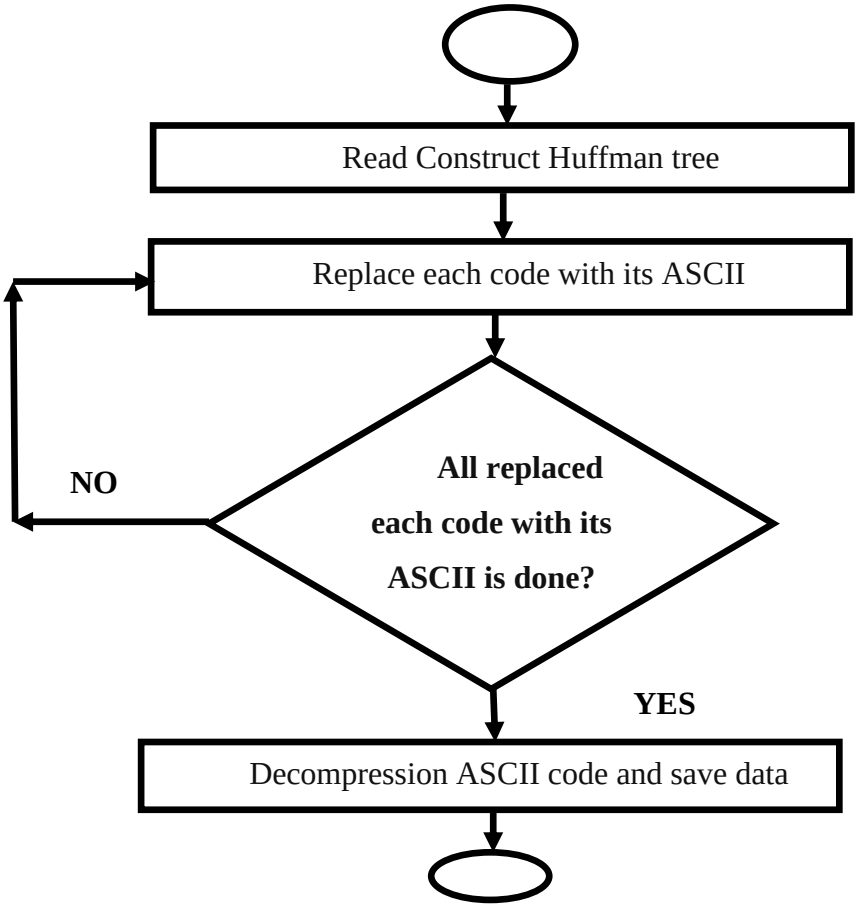

Figure 3.9: Huffman Decompression algorithm process Steps.

3.3.6 Compression process Using LZW algorithm

1- Input first byte stores in the string.
2- Input next byte and stored in char.
3- If string + char in the table in atable.
   A. Yes, go to step 7.
   B. Else, Go Step 4.
4- Output the code for String.
5- Add an entry in the table for string ÷ char.
6- String = char.
7- String = String ÷ char.
8- More bits to Input.
   A. Input next byte stored in char.
   B. Else Go Step 9.
9- The output code for string and end.

Figure 3.10: below Shown and explain Steps LZW Compression algorithm:

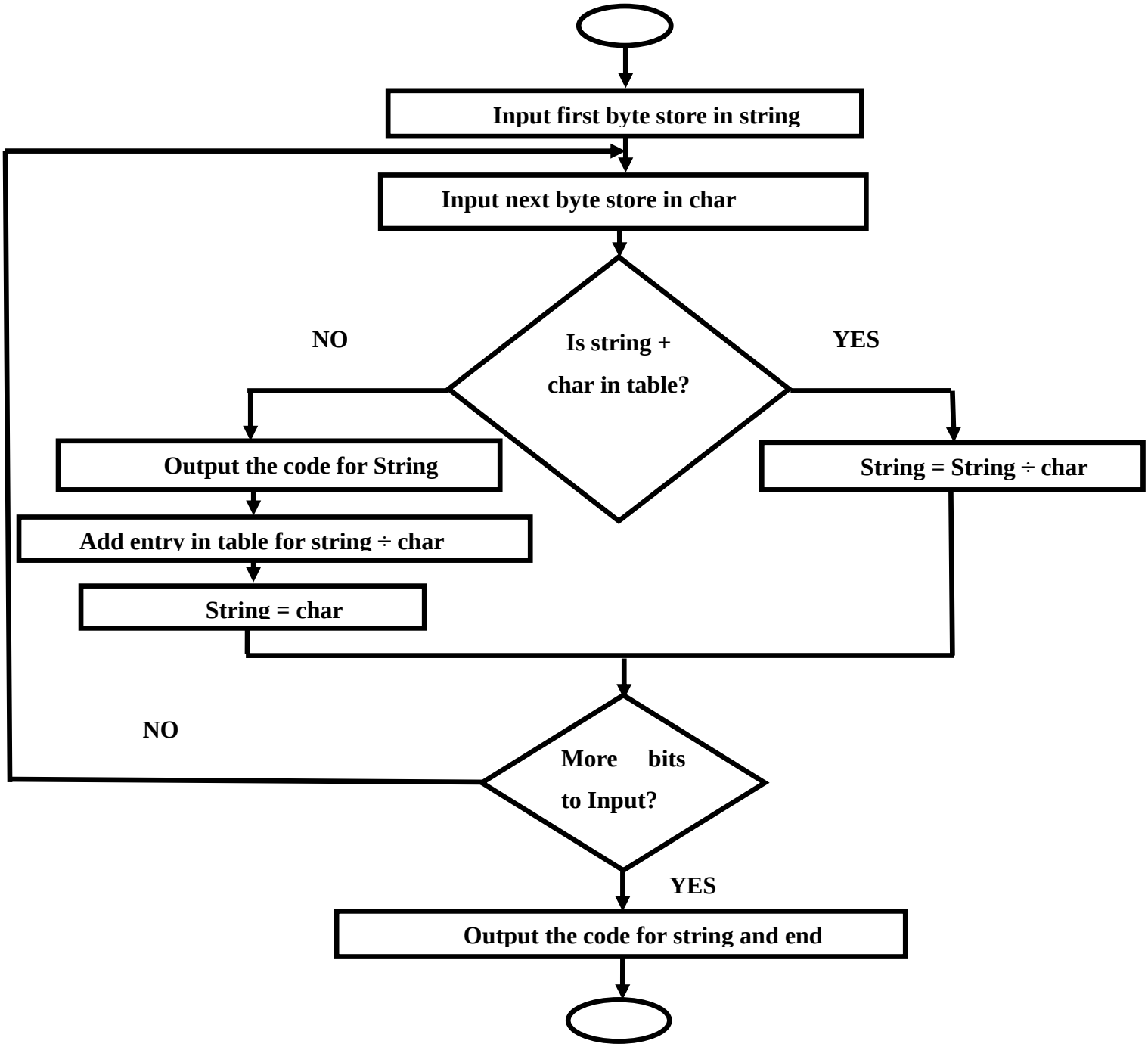

Figure: 3.10: LZW Compression algorithm process Steps

3.3.7 Decompression process Using LZW algorithm

1- Input first code stored in OLD CODE.
2- Output translation of OLD CODE.
3- Input next code stored in NEW CODE.
4- Is NEWCODE in the table.

A. Yes! Go step 7.

B. Else No Go step 5.

5- String = translation of OLD CODE.

6- String = String ÷ Char.

7- String = translation of NEW CODE.

8- Output string.

9- Char = the first character in the string.

10- Add an entry in the table for OLDCODE + CHAR.

11- OLDCODE = NEWCODE.

12- More Code to input?

A. Yes! Go Step 3.

B. No! Go Step 13.

13- Output bytes and end.

Figure 3.11: below Shown and explain Steps LZW Decompression algorithm:

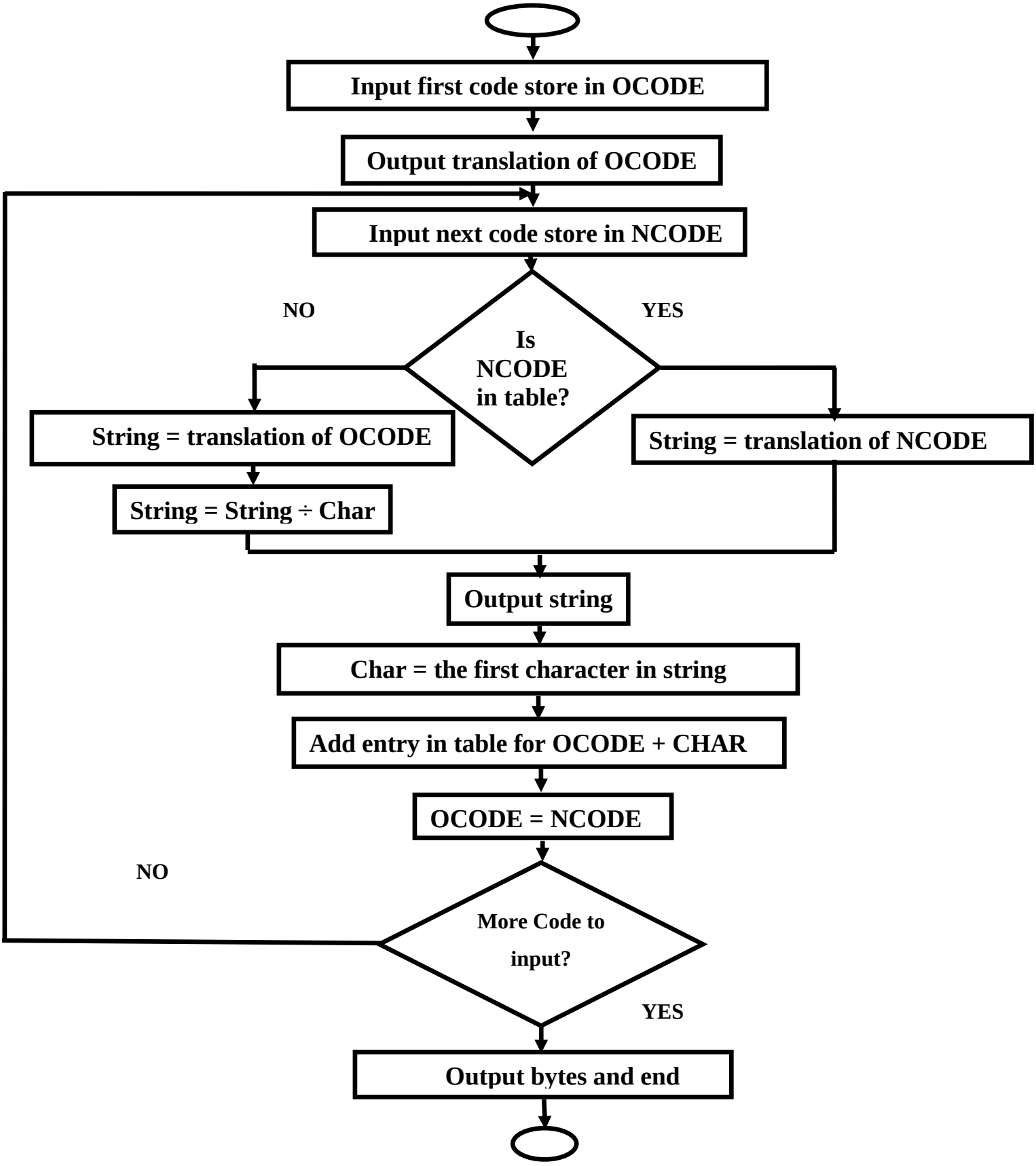

Figure: 3.11: LZW Decompression algorithm process Steps.

CHAPTER 4

RESULT AND DISCUSSION

CHAPTER 4

Result and Discussion

4-1 Result

Comparative analysis of multilevel image steganography (secure LSB-L1) and (secure LSB-L2) based image steganography) has been done on the basis of parameters like Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Normalized Cross-Correlation (NCC), Average Difference (AD), Structural Content (SC), Maximum Difference (MD) and Normalized Absolute Error (NAE) embeds data size.

4-2 Quality Image Measurement

4-2-1 Mean Squared Error (MSE):

Is the average squared difference between a reference image and a modified image (stego-image). It is computed pixel-by-pixel by adding up the squared differences of all the pixels and dividing by the total pixel count.

4-2-2 Peak signal to noise ratio (PSNR):

It is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is usually expressed in terms of dB for a wide range of signals The PSNR is most commonly used as a measure of the quality of reconstruction for lossy compression. The cover image, in this case, is the original data, and the information logo is the error introduced by watermarking. A higher PSNR would normally indicate that the reconstruction is of higher quality. A small value of PSNR indicates poor quality.

4-2-3 Normalized cross correlation (NCC):

It is a measure of similarity of two series as a function of the displacement of one relative to the other. Has been commonly used as a metric to evaluate the degree of similarity (or dissimilarity) between two compared images. The main advantage of the normalized cross-correlation over the ordinary cross correlation is that it is less sensitive to linear changes in the amplitude of illumination in the two compared images. Furthermore, the Normalized Cross Correlation is confined in the range between –1 and 1.

4-2-4 Average Difference (AD):

Calculate Average Difference between original Image and distance Image

4-2-5 Structural Content (SC):

Structural Content is the measure of information content in the image and is defined as the ratio of the structural information content of the original to the recovered information logo. A large value of Structural Content (SC) thus means that the image is of poor quality.

4-2-6 Maximum Difference (MD):

Calculate Max Difference Error between original Image and distance Image.

4-2-7 Normalized Absolute Error (NAE):

A large value of Normalized Absolute Error (NAE) means that image is poor quality.

The following table 4-1 shown and explain Quality Image Measurement Calculation

Table 4.1 shown and explain Quality Image Measurement Calculation.

Mean Square Error	$MSE = \frac{1}{MN} \sum_{j=1}^{M} \sum_{k=1}^{N} \left(x_{j,k} - x'_{j,k} \right)^2$
Peak Signal to Noise Ratio	$PSNR = 10 \log \frac{(2^n - 1)^2}{MSE} = 10 \log \frac{255^2}{MSE}$
Normalized Cross-Correlation	$NK = \sum_{j=1}^{M} \sum_{k=1}^{N} x_{j,k} \cdot x'_{j,k} \Big/ \sum_{j=1}^{M} \sum_{k=1}^{N} x_{j,k}^2$
Average Difference	$AD = \sum_{j=1}^{M} \sum_{k=1}^{N} \left(x_{j,k} - x'_{j,k} \right) / MN$
Structural Content	$SC = \sum_{j=1}^{M} \sum_{k=1}^{N} {x_{j,k}}^2 \Big/ \sum_{j=1}^{M} \sum_{k=1}^{N} {x'_{j,k}}^2$
Maximum Difference	$MD = Max \left(\left| x_{j,k} - x'_{j,k} \right| \right)$
Laplacian Mean Square Error	$LMSE = \sum_{j=1}^{M} \sum_{k=1}^{N} \left[O\left(x_{j,k}\right) - O\left(x'_{j,k}\right) \right] \Big/ \sum_{j=1}^{M} \sum_{k=1}^{N} \left[O\left(x_{j,k}\right) \right]^2$ $O\left(x_{j,k}\right) = x_{j+1,k} + x_{j-1,k} + x_{j,k+1} + x_{j,k-1} - 4x_{j,k}$
Normalized Absolute Error	$NAE = \sum_{j=1}^{M} \sum_{k=1}^{N} \left| x_{j,k} - x'_{j,k} \right| \Big/ \sum_{j=1}^{M} \sum_{k=1}^{N} \left| x_{j,k} \right|$

There is a different message size have been used to embed them in different image size in the upper level of image steganography, the first message (first secret message) will be Use is shown in figure 4.1

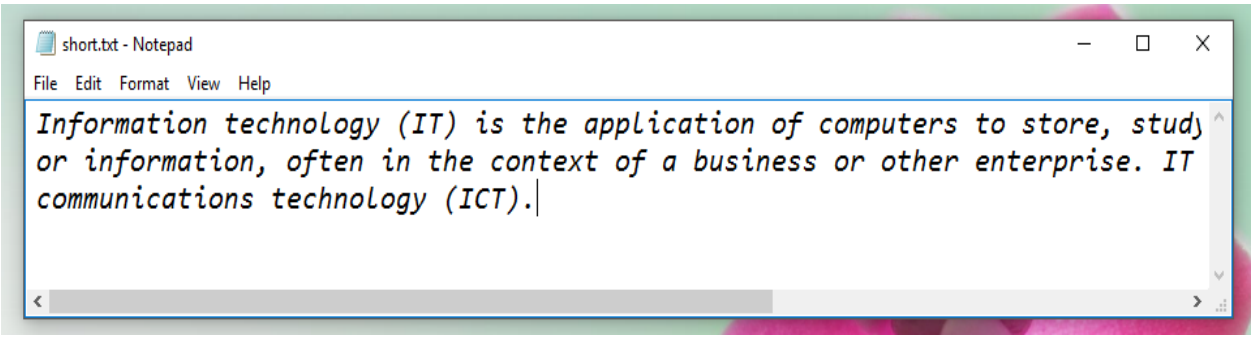

Figure 4.1: the first secret message (message1)

The size of the first secret message is 270 bytes and the size will be increased in the Next secret message, the second secret message is shown in figure 4.2.

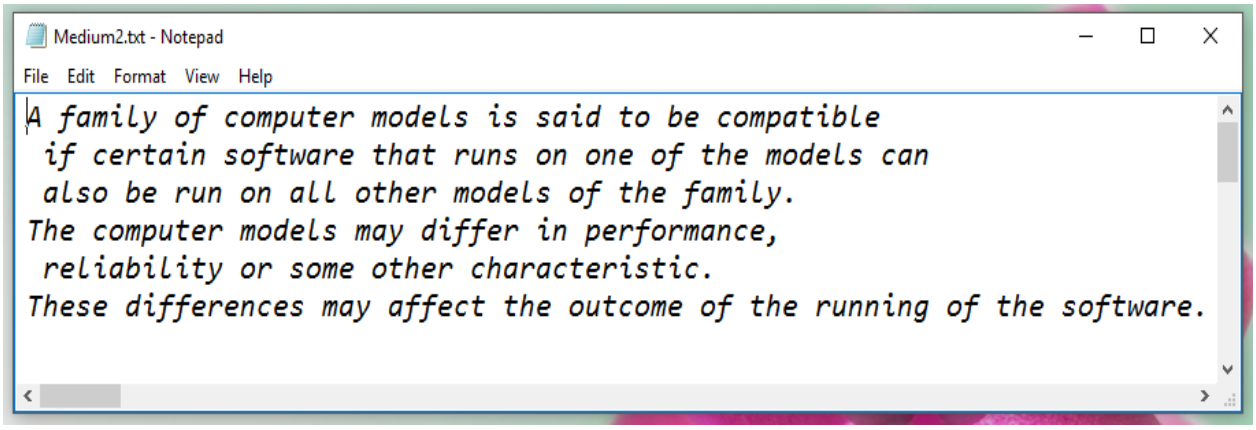

Figure 4.2: the second secret message (message2)

The size of the second secret message is 4,650 bytes and the size will be increased in the next secret message, the third secret message is shown in figure 4.3.

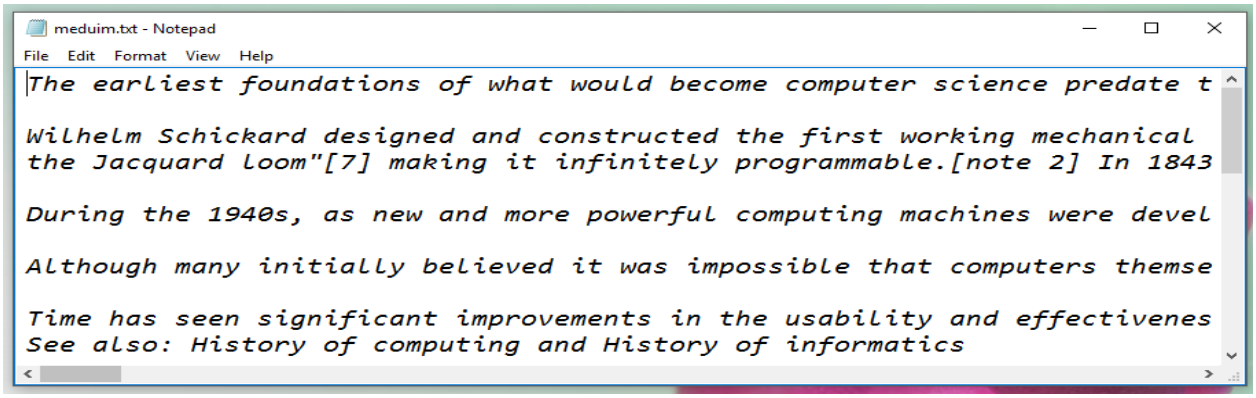

Figure 4.3: the third secret message (message3)

The size of the third secret message is 8,232 bytes (almost the maximum capacity).

After the upper level (secure LSB-L1) is applied to the above secret messages the output is more than one image. Figure 4.4: Shown and explain level one applied method

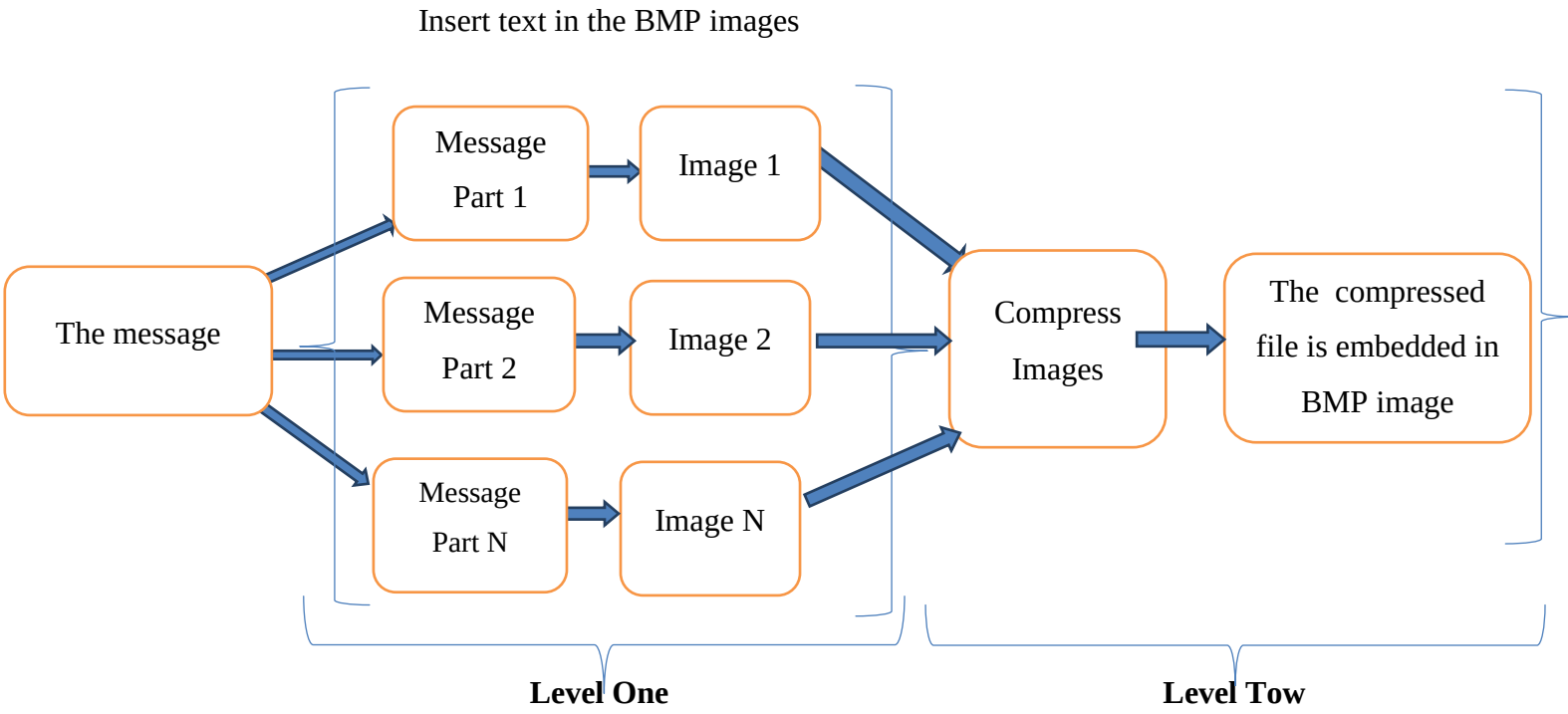

Figure 4.4 shows the proposed algorithm to hide information

In level one (secure LSB-L1) three images with the different sizes have been used, each image concealing one of the secret messages. The first cover image is the black-box image and is concealing (message1) as a secret data; the size of the stego image is 1.40 MB. Figure 4.5 shows the Black-box stego images.

The second cover image is a Red - box image and is concealing (message2) as secret data; the size of the stego image is 2.31 MB Figure 4.6 shows the Red-box _stego image.

Figure 4.7 shows a White-box_stego image with (message3) as embedded secret data and the size of which is 3.81 MB.

Original Image **Stago-images**

Figure 4.5: shows the tow Black-boxes stego images from level one with secret message1

Original Image **Stego-images**

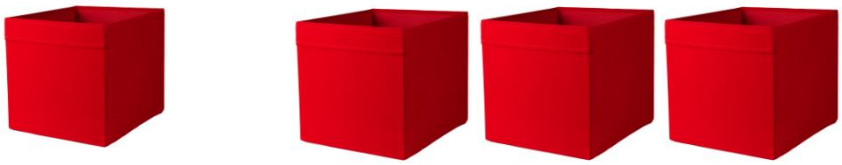

Figure 4.6 shows the three Red-boxes stego images from level one with secret message2

Original Image **Stego-images**

Figure 4.7 shows the three White-boxes stego images from level one with secret message3

In level Tow (secure LSB-L2) we also have three images of different sizes and dimensions used.

The first image is the monaliza image Figure 4.8 with dimension 3000 × 4482 used as a cover image the first secret data to be embedded in this cover image is the compressed black-box stego-images which were the output of Winrar compression, Second is the compressed black-box stego-images which were the output of Huffman compression algorithm and third secret to be embedded in the compressed black-box stego-images which were the output of the LZW compression algorithm,The new stego-images show in figure 4.9, figure 4.10 and figure 4.11 as examples

Figure 4.8: monaliza original image

Figure 4.9: monaliza stego1image Embedded data: compressed stego-image by Winrar.

Figure 4.10: monaliza stego2 image Embedded data: compressed stego-image By Huffman Algorithm

Figure 4.11: monalizastego3image Embedded data: compressed stego-image By LZW Algorithm

Table 4.2 shows the experiment results of the monaliza _stego images and contains the Quality Imageparametervalues of stego images above. Figure 4.12 is a Diagram Showing its Quality Image parameter values

Secret message	Size of Secret message	Size of secret Key	Level one Carry image Size embedded image	Size of stego image Total of stego-images	Compression algorithm	Level tow embedded Image Level Tow –Size embedded image	Mean Square Error	Peak Signal to Noise Ratio	Normalized Cross-Correlation	Average Difference	Structural Content	Maximum Difference	Normalized Absolute Error
Message1	270 bytes	55 bytes	Black Box 700×700 1.40 MB	1.86 MB 1 Image	Winrar	MonaLisa 3000×4482 38.4 MB	0.010	67.91	1	0.00023	1	58	0.0002
					Huffman		0.050	61.15	1	-0.004	1	59	0.0008
					LZW		0.018	65.53	1	0.003	1	56	0.0003
Message2	2,016 bytes	55 bytes	Black Box 700×700 1.40 MB	1.86 MB 2 Images	Winrar	MonaLisa 3000×4482 38.4 MB	0.022	64.76	1	0.001	1	55	0.0003
					Huffman		0.036	62.58	1	0.001	1	59	0.0006
					LZW		0.037	62.48	1	0.006	1	56	0.0006
Message3	7,445 bytes	55 bytes	Black Box 700×700 1.40 MB	1.86 MB 7 Images	Winrar	MonaLisa 3000×4482 38.4 MB	0.078	59.29	1	-0.001	1	55	0.0012
					Huffman		0.128	57.09	1	-0.001	1	59	0.0020
					LZW		0.130	57.00	1	0.013	1	56	0.0021

Table 4.2

Experimental results-1

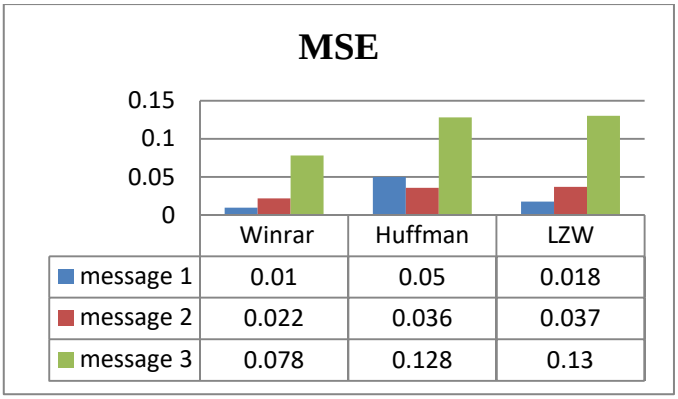

Figure 4.12 MSE value for MonaLisa image

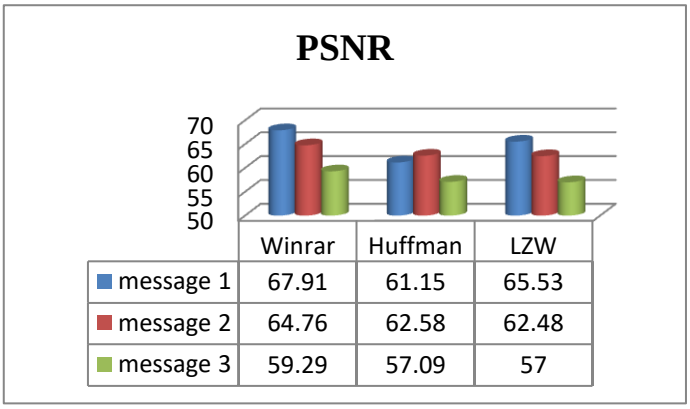

Figure 4.13 PSNR value for MonaLisa image

The Second image is the cyber-security image Figure 4.14 with dimension 4000 × 2664 used as a cover image the first secret data to be embedded in this cover image is the compressed red-box stego-images which were the output of Winrar compression, Second is the compressed red-box stego-images which were the output of Huffman compression algorithm and third secret to be embedded in the compressed red-box stego-images which were the output of the LZW compression algorithm, The new stego-images show in figure 4.15, figure 4.16 and figure 4.17 as example

Figure 4.14: cyber-security Original image

Figure 4.15: cyber-security stego1-image embedded data: compressed stego-image by Winrar

Figure 4.16: cyber-security stego2- image. Embedded data: compressed stego-image by Huffman Algorithm

Figure 4.17: cyber-security stego3 Embedded data: compressed stego-image by LZW Algorithm

Table 4.3 shows the experiment results of the cyber-security –stego images and contains the Quality Imageparametersvalues of stego images above. Figure 4.18 is a Diagram Showing its Quality Image parameters values

Secret message	Size of Secret message	Size of secret Key	Level one Carry image Size embedded image	Size of stego image Total of stego-images	Compression algorithm	Level tow embedded Image Level Tow – Size embedded image	Mean Square Error	Peak Signal to Noise Ratio	Normalized Cross-Correlation	Average Difference	Structural Content	Maximum Difference	Normalized Absolute Error
Message 1	548 bytes	55 bytes	Red Box 900×900 2.31 MB	3.08 MB 1Image	Winrar	Cybersecurity 4000×2664 30.4 MB	0.118	57.45	1	0.006	1	205	0.0061
					Huffman		0.268	53.86	1	0.010	1	206	0.0014
					LZW		0.170	55.82	1	0.035	1	206	0.0009
Message 2	4,650 bytes	55 bytes	Red Box 900×900 2.31 MB	3.08 MB 3Images	Winrar	Cybersecurity 4000×2664 30.4 MB	0.349	52.71	1	0.031	1	201	0.0019
					Huffman		0.387	52.26	1	0.032	1	206	0.0021
					LZW		0.395	52.17	1	0.093	1	203	0.0021
Message 3	10,882 bytes	55 bytes	Red Box 900×900 2.31 MB	3.08 MB 6 Images	Winrar	Cybersecurity 4000×2664 30.4 MB	0.663	49.91	1	0.047	1	202	0.0029
					Huffman		1.354	46.81	1	0.089	1	206	0.0049
					LZW		1.439	46.55	1	0.271	1	203	0.0050

Table 4.3
experimental result-2

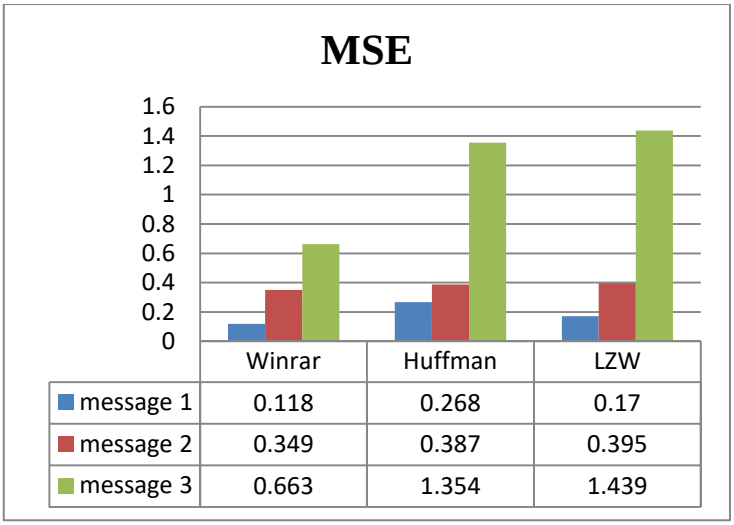

Figure 4.18 MSE value for Cybersecurity image.

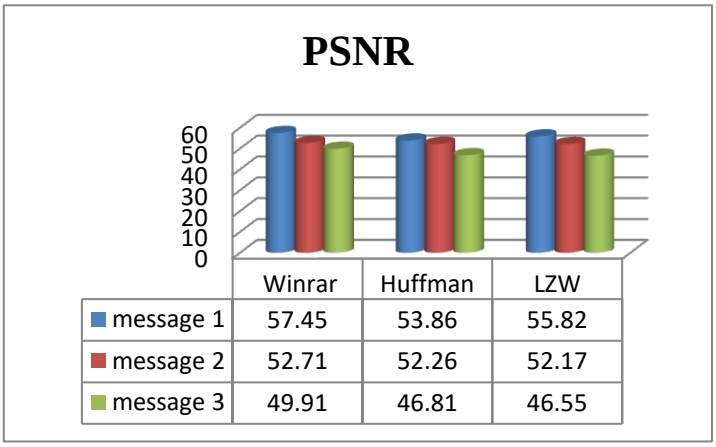

Figure 4.19 PSNR value for Cybersecurity image.

The third image is the horse image Figure 4.20 with dimension 3500 × 2187 used as a cover image the first secret data to be embedded in this cover image is the compressed white -box stego-images which were the output of Winrar compression, Second is the compressed white -box stego-images which were the output of Huffman compression algorithm and third secret to be embedded in the compressed white -box stego-images which were the output of the LZW compression algorithm , The new stego-images show in figure 4.21, figure 4.22 and figure 4.23 as examples

Figure 4.20: horse original image

Figure 4.21 horse stego1 image Embedded data: compressed stego-image by Winrar

Figure 4.22 horse stego2 image. Embedded data: compressed stego-Huffman

Figure 4.23 horse stego3 image. Embedded data: compressed stego- image by image by LZW

Table 4.4 shows the experimental results of the horse–stego images and contains the Quality Image parameter values of stego images above. Figure 4.24 and 4.25 are a Diagram Showing its Quality Image parameter values.

Secret message	Size of Secret message	Size of secret Key	Level one Carry image	Size of stego image	Compression algorithm	Level tow embedded Image	Mean square Error	Peak Signal to Noise Ratio	Normalized Cross-Correlation	Average Difference	Structural Content	Maximum Difference	Normalized Absolute Error
			Size embedded image	Total of stego-images		Level Tow Size embedded image							
Message1	818 bytes	55 bytes	White box 1333×1000 3.81 MB	5.08 MB 1 Image	Winrar	Horse 3500×2187 21.8 MB	0.141	56.52	1	-0.001	0.991	1	0.0031
					Huffman		0.721	49.50	1	-0.156	1	3	0.0125
					LZW		0.304	53.31	1	-0.024	1	1	0.0064
Message2	8,232 bytes	55 bytes	White box 1333×1000 3.81 MB	5.08 MB 3 Images	Winrar	Horse 3500×2187 21.8 MB	0.429	51.90	1	-0.175	0.991	1	0.0089
					Huffman		0.759	49.33	1	-0.180	0.990	3	0.0127
					LZW		1.160	47.49	1	-0.143	1	3	0.0169
Message3	11,487bytes	55 bytes	White box 1333×1000 3.81 MB	5.08 MB 4 Images	Winrar	Horse 3500×2187 21.8 MB	0.419	51.91	1	-0.176	0.990	1	0.0089
					Huffman		1.741	45.72	1	-0.491	0.991	3	0.0222
					LZW		1.651	45.95	1	-0.363	1	3	0.0216

Table 4.4

Experimental results-3

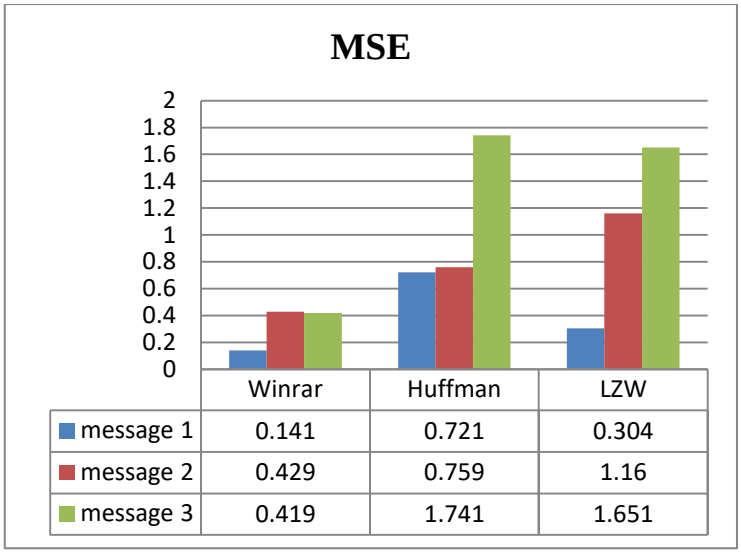

Figure 4.24 MSE value for horse image

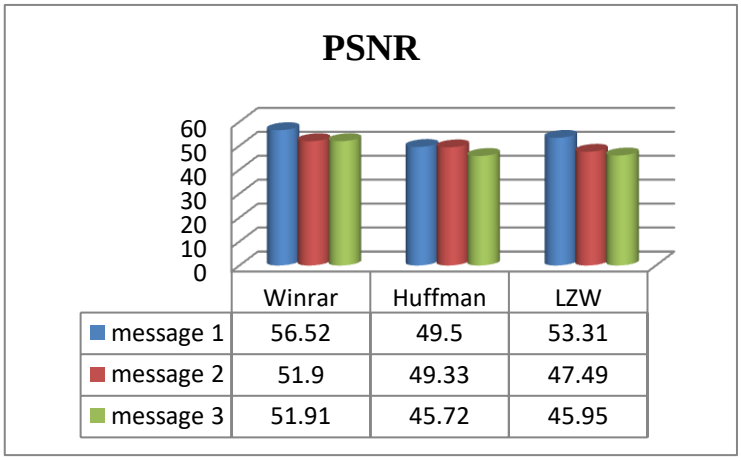

Figure 4.25 PSNR value for horse image

CHAPTER 5

CONCLUSION AND FUTURE WORKS

CHAPTER 5

Conclusion and Future Works

5.1 Conclusion

The proposed model adds a level of security through the main theme of steganography: “hiding information in plain sight”. The cover object usually does not invite suspicion, since it looks similar to the original object to the general observer.

The main objective is applying and improves the way to hide the information division the text on more BMP images.

In this thesis, a new concept for performing hidden secret data, called Multilevel Steganography for image steganography, was presented.

The proposed method is two levels of image steganography, In the level one uses modified least significant bit (secure LSB-L1) image steganography to hide the secret information into more than one image carrier of the text (at least in 2 images). And that improving hide information by being distributed in more than one image carrier. The last step in this level, adding a key string to secure the information.

The Level tow employs the algorithm called (secure LSB-L2). In this level (secure LSB-L2) provides using several layers lieu of using only LSB layer of the image. Writing data starts from the last layer (8th or LSB layer); because significant of this layer is least and every upper layer has doubled significant from its down layer. So every step we go to upper layer image quality decreases and image retouching transpires.

Multilevel Steganography has potential benefits, as it may enhance the confidentiality of the secret information by using two level image steganography in one the system and add more complexity to the steganography process through applying it in two levels.

Measuring the performance of proposed algorithm has been applied using many experiments and calculate many values of each experiment, the first value is Peak signal to noise ratio (PSNR), this ratio is used as a quality measurement between two images. If PSNR ratio is high then images are best of quality, the second measurement value is Mean Squared Error is the average squared difference between a reference image and a modified image (stego-image). And other calculates values are Normalized Cross-Correlation, Average Difference, Structural Content, Maximum Difference and Normalized Absolute Error.

There are many experiments have been conducted through the different size of secret messages (secret message one, two and three) utilized as a secret data in level one. And compress in one file, then concealed in one BMP image the output is compressed file or (intermediate object) and it's used as a secret data in level two.

5.2 Recommendations

- The proposed method can be used in military applications for secure communications.
- Try to check the result of proposed algorithm using the grayscale image on both levels to compare the performance results.
- Apply another compression technique.
- Apply compression to a text file.

5.3 Future Work

1- Adding Advance encryption algorithm to in the upper level to encrypt the secure text message to increase the security to proposed method.

2- Adding one more level (level Three)

3- Increase the System functionality to hide all other data types like audio, video not only text data and images.

4- Trying to enhance the performance of algorithms in both levels to increase the system capacity.

References

Abdul-Jabbar, Jamal Hamed, (2006), “Watermark Robustness Algorithms Using Error Correcting Codes on Compressed Image ", Ph.D. Thesis, Informatics Institute for Postgraduate (IIP), Baghdad University, Baghdad, Iraq.

Al-Najjar, Atef Jawad(2008). "The decoy: multi-level digital multimedia steganography model." WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering. No. 12. World Scientific and Engineering Academy and Society,.

Al-Dieimy, I.I.U, (2002), “Information Hiding In an Open Environment ", Computer Science & Information System (CSIS), University of Technology Malaysia, Malaysia.

Al-Hamami, Maha, (2006), " Information Hiding Attack in Image”, Informatics Institute for Postgraduate Studies (IIIPS), Baghdad, Iraq.

Al_Mayyahee, Noha, (2005), “New Robust Information Hiding Technique”, Informatics Institute for Postgraduate Studies (IIPS), Baghdad, Iraq.

Andre, A.F, S.A, & Reza, (2001), “ Zero-knowledge Watermarking Detection and Proof of Ownership " , Information Hiding 4th International Workshop (IHIW),Proceeding,Vol.2137, Lecture Notes in Computer Science, University ofTechnology, Springer, p.p273-288.

Aos Alaa Zaidan, Bilal Bahaa Zaidan, A.W.Naji, Fazida Othman, Alaa Taka, (2009),“ Securing Cover-File of Hidden Data Using Statistical Technique and AES Encryption Algorithm”, International Conference on Information management and engineering (ICIME09), Indexed by (EI Compendex, INSPEC, Thomson ISI, IEEE XploreTM, IEEE Computer Society (CSDL) digital libraries.), KL, Malaysia.

Aos, A.Z.Ansaef (2009) Securing cover-file of hidden data using statistical technique and aes encryption algorithm. Masters thesis, University of Malaya.

Asylum, T, (April 5, 2001), “Detection and Proof of Ownership”, International Conference of Information Hiding 7th (ICIH), University of Technology Malaysia,Malaysia.

Avedissian, L.Z, (2005), “Image in Image Steganography System”, Ph. D. Thesis,Informatics Institute for Postgraduate Studies (IIIPS), University of Technology,Baghdad, Iraq.

Bilal Bahaa Zaidan, Aos Alaa Zaidan, Fazidah Othman, (2008), “ Enhancement Of The Amount Of Hidden Data And The Quality Of Image ”, MyEduSec08. www.udm.edu.my/MyEduSec/2008, k.terengganu, Malaysia.

Brigit, P.T, (1996), " Information Hiding Terminology Information Hiding ” , First

International Workshop (FIW), proceeding, vol. 1174, of Lecture Notes in Computer Science, University of Technology ,Springer, P.P 347-350.

C. J. S. B, (2002),” Modulation and Information Hiding in Images”, Vol. 1174, of Lecture Notes in Computer Science, University of Technology, Springer, p.p 207-226.

Clelland, C.T.R, V.P & Bancroft, (1999), “ Hiding Messages in DNAMicroDots”Proceedings of IEEE International Symposium on Industrial Electronics (ISIE) ,University of Indenosia , Indenosia, Vol. 1, p.p 315-327.

Davern, P.S, M.G, (2002), “Steganography It History and Its Application to Computer Based Data Files”, School of Computer Application (SCA), Dublin City University. Working Paper. Studies (WPS), Baghdad, Iraq.

Dorothy, E.R, D.K, (2000), “Cryptography and Data Security”, IEEE International Symposium on Canada Electronics (ISKE), University of Canada, Canada, Vol.6, p.p 119-122

Eltyeb E. Abedelgabar (2009) Comparison of LSB Steganography in BMP and JPEG Images

Eltyeb E. Abedelgabar Sridevi, R., A. Damodaram, and S. V. L. Narasimhan. "EFFICIENT METHOD OF AUDIO STEGANOGRAPHY BY MODIFIED LSB ALGORITHM AND STRONG ENCRYPTION KEY WITH ENHANCED SECURITY." Journal of Theoretical & Applied Information Technology

Er. Meenakshi Garg (2014) Research paperon Text Data Compression Algorithm using Hybrid Approach

Figueiredo, D.T, (May2001), “ Hide In Picture (HIP) ” , International Conference on Advanced Management Science (ICAMS01), Indexed by (EI Compendex, INSPEC, Thomson ISI (ISTP), IEEE XploreTM, IEEE Computer Society (CSDL) digital libraries), Singapore.

Gurmeet Kaur, and Aarti Kochhar(2012). "A steganography implementation based on LSB & DCT." International Journal of Science and Emerging.

Ingemar, L.M.J, A.B, J.C&Mattew, (2007), ” Digital Watermarking Morgam Kaufmann ” , International Conference on Future Computer and Communication (ICFCC07), Indexed by (EI Compendex, INSPEC, Thomson ISI, IEEE XploreTM), India.

Ingemar, J.COX, M.L.M, Jefferey A.Bloom, (2002), “ Digital watermarking for Hidding Morgam Kaufmann “ , International Conference on Information management and engineering (ICIME02), Indexed by (EI Compendex, INSPEC, Thomson ISI, IEEE XploreTM, IEEE Computer Society (CSDL) digital libraries.), India.

Inoue, S.M, K.J, (2006), “A Proposal on Information Hiding Methods Using Xml”, The 17th Annual meeting of the association for Natural Language Processing (AMANLP), Canada, p.p 135-138.

Jafer, A.M, (2006), “Image Steganography Using Wavelet Transform techniques”, M.Sc. Thesis, University of Baghdad, Baghdad, Iraq.

Jajodia, S., (2000), “Advances in Information Security ", The origins of cryptology: The Arab contributions, Cryptologia journal, George Mason University, VO.16, p.p 97–126. Jajodia, S., (2005), “Steganalysis: The Investigation in Information Security”, http://www.jjtc.com/stegdoc/, the origins of cryptology: The Arab contributions, Cryptologia journal, George Mason University, VO.6, p.p 15–26.

Johnson, F. N, (June 2. 2005), “Steganography in IPV6”, Information System and Software Engineering (ISSE), George Mason University". http://www.jjtc.com/stegdoc/.

Johnson, N. F. S. D, Z. (2001), “Information Hiding: Steganography and Watermarking-Attacks and Countermeasures”, Center for Secure Information Systems (CSIS), Boston/Dordrecht/London, George Mason University.

JONAS (2015)Design Study of a Computer System Employing Memory Compression by JONAS ANDERSSON and NIKLAS DOVERBO June 2015.

Katzenbisser, S. P., P. A, (2005) “ Information Hiding Techniques for Steganography and Digital water marking ”, Proceedings of the Eighth Symposium on programming Languages and Software Tools SPLST'05, available from: Norwood: Artech House.

(Kristine Arthur-Durett December 2014)The Weakness of WinRAR Encrypted Archives to Compression Side-channel Attacks by Kristine Arthur-Durett December 2014

Lempel Ziv Welch Data Compression using Associative Processing as an Enabling Technology for Real-Time Applications. By Manish Narang, A Thesis Submitted In Partial Fulfillment of the Requirements for the Degree of MASTER OF SCIENCE In Computer Engineering March 1998.

Lee, S.J.S.H.J, (2001), “A Survey of Watermarking Techniques Applied to multimedia”, Proceedings of IEEE International Symposium on Industrial Electronics (ISIE), Vol. 1, p.p 272-277.

Lin, E.T.D, E.J, (2005), “A Review of Data Hiding in Digital Images”, Proceedings of IEEE International Symposium on Industrial Electronics (ISIE), VO.22, p.p 62-68.

Manish Narang (1998) Lempel Ziv Welch Data Compression using Associative Processing as an Enabling Technology for Real-Time Applications. By Manish Narang, A Thesis

Submitted In Partial Fulfillment of the Requirements for the Degree of MASTER OF SCIENCE In Computer Engineering March 1998

Martin, K, (2007), “Digital Watermarking Frequently Asked Questions (FAQ)”, Information Hiding Technologies in Security: Services, Security and Management, Master’s Thesis, Lappeenranta University Of Technology, Available from: http://www.tbrc.fi/pubfilet.

Mega, T, (2004),”ExecutableFileTypes”, International Conference on Information Security and engineering (ICISE04), Indexed by (EI Compendex, INSPEC, Thomson ISI, IEEE XploreTM, IEEE Computer Society (CSDL) digital libraries.),Japan, http://www.megatokyo.com/osfaq2/index.php/executable%20file%20types.

Mohammad Tanvir Parvez, and Adnan Gutub(2009). "RGB intensity based variable bits image steganography." Asia-Pacific Services Computing Conference, 2008.APSCC'08. IEEE. IEEE, 2009.

Musa, A.K, (2006), “Watermark Application in Color Images Using Wavelet Transforms ", Ph.D. Thesis, Informatics Institute for Postgraduate Studies (IIPS). Baghdad, IRAQ.

(NILKESH PATRA And SILA -2007) DATA REDUCTION BY HUFFMAN CODING AND ENCRYPTION BY INSERTION OF SHUFFLED CYCLIC REDUNDANCY CODE ByNILKESH PATRA And SILA SIBA SANKAR 2007.

Noel, S.S, H, (2005), “Multimedia Authenticity with Watermarking “, Spie's 14th Annual International Symposium on Aerospace/ Defense Sensing (SAISA), Simulation and Controls. Orland, Florida.

Nspw, A.B.C, CO, (2006), “Paradigam In Steganography”, Center for Hiding Assurance, Computer System (CHACS), Naval Research Laboratory, Washington, DC 20375, us Government work, Research Supported by the Office of Naval Research. Republic of Ireland.

Ross, A.J.N.A.S, (2005), “The Steganography File System”, Information Hiding: Second International Workshop (SIW), Proceeding, Vol. 1525 of Lecture Notes in Computer Science, Springer, p.p 73-82.

Samir Kumar (2012) Bitmap Steganography: An Introduction Beau Grantham 20070413

Bandy opadhyay, Samir Kumar, and Barnali Gupta Banik. "Multi-Level Steganographic Algorithm for Audio Steganography using LSB Modification and Parity Encoding Technique." International Journal of Emerging Trends & Technology In Computer Science (IJETTCS) 1.2 (2012).

S, J.N.F.J, (2004), “Steganalysis: The Investigation of Hidden Information “, Center for Information Systems (CIS), George Mason University, MS: 4A4, Fairfax, Virginian 22030-4444.

Seleborg,S.,(2004),"Advanced Encryption Standard (AES)", ACM Proceedings Of The 2004 Annual Research Conference Of The South African Institute Of Computer Scientists And Information Technologists On Enablement Through Technology(CSITE), South African, http://www.axantum.com/AxCrypt/etc/About-AES.pdf.

Sellar, D., (2003), "An Introduction to Steganography", Spie's 14th Annual International Symposium on Aerospace/ Defense Sensing (SAISA),http://ww.cs.uct.ac.za/Simulation and Controls. Orland, Florida.

Singla, Deepak, and Rupali Syal (2013). "Data Security Using LSB & DCT Steganography In Images." Editorial Board: 359

Smith, B., Cpre531 (10 Dec. 2006) "Steganagroply in vedio Streaming", Final Project, Center for Secure Information Sysytems (CSIS), Boston/Dordrecht/London, George Mason University.

Souvik Bhattacharyya(2, February 2011) Hiding through Multi Level Steganography and SSCE Department of Computer Science and Engineering, University Institute of Technology The University of Burdwan, Burdwan, India

Standard, I.T.A.E., (2007), "Introduction to Advanced Encryption Standard (AES) ", http://www.itu.dk/courses/DSK/E2003/DOCS/aes_introduction.pdf. Article.

Stive, (2008), "Microsoft Portable Executable and Common Object File Format Specification",MicrosoftCooperation(MC),Revision1.8,http://www.microsoft.com/whdc .

(TAO TAO B.E 1998)COMPRESSED PATTERN MATCHING FOR TEXT AND IMAGES By TAO TAO B.E. Huazhong University of Science and Technology, 1994 M.E. Huazhong University of Science and Technology, 1998 M.S. University of Central Florida, 2000.

Umbangh, S.E. (2004), "Computer Vision and Image Processing", London, Prentices Hall

Welzel, D., & Hausen, H. L. (1993). A five steps method for metric-based software evaluation: effective software metrication with respect to the quality standard. *Journal of ACM, 39*(2), 273 – 276.

Walia, Dr. Ekta, Payal Jain, and Navdeep Navdeep (2010). "An analysis of LSB & DCT based steganography." Global Journal of Computer Science and Technology 10.1 (2010).

Zaidoon KH A. Zaidan, A.A.Zaidan, B.B& Alanazi.H.O., (2010). Overview: main fundamentals for steganography. *Journal of Computing*, 2(3), pp.40-43.