MEU جـامـعـة الـشـرق الأوسـط
MIDDLE EAST UNIVERSITY

Hiding Secret Images within RGB Images Using an Enhanced LSB Method

اخفاء الصور السرية في الصور الملونه باستخدام طريقة LSBالمحسنة

By
Mohamed Zeyad Alhaj Qasem

Supervisor
Dr. Mudhafar Al-Jarrah

Submitted in Partial Fulfillment of the Requirements for the Master Degree in Computer Science

Department of Computer Science

Faculty of Information Technology

Middle East University

Amman – Jordan
2014

Authorization Statement

I, Mohamed Zeyad Alhaj Qasem, authorize Middle East University to supply hard and electronic copies of my thesis to libraries, establishments, bodies, and institutions concerned with research and scientific studies upon request, according to university regulations.

Name: Mohamed Zeyad Alhaj Qasem

Date: 26/5/2014

Signature:

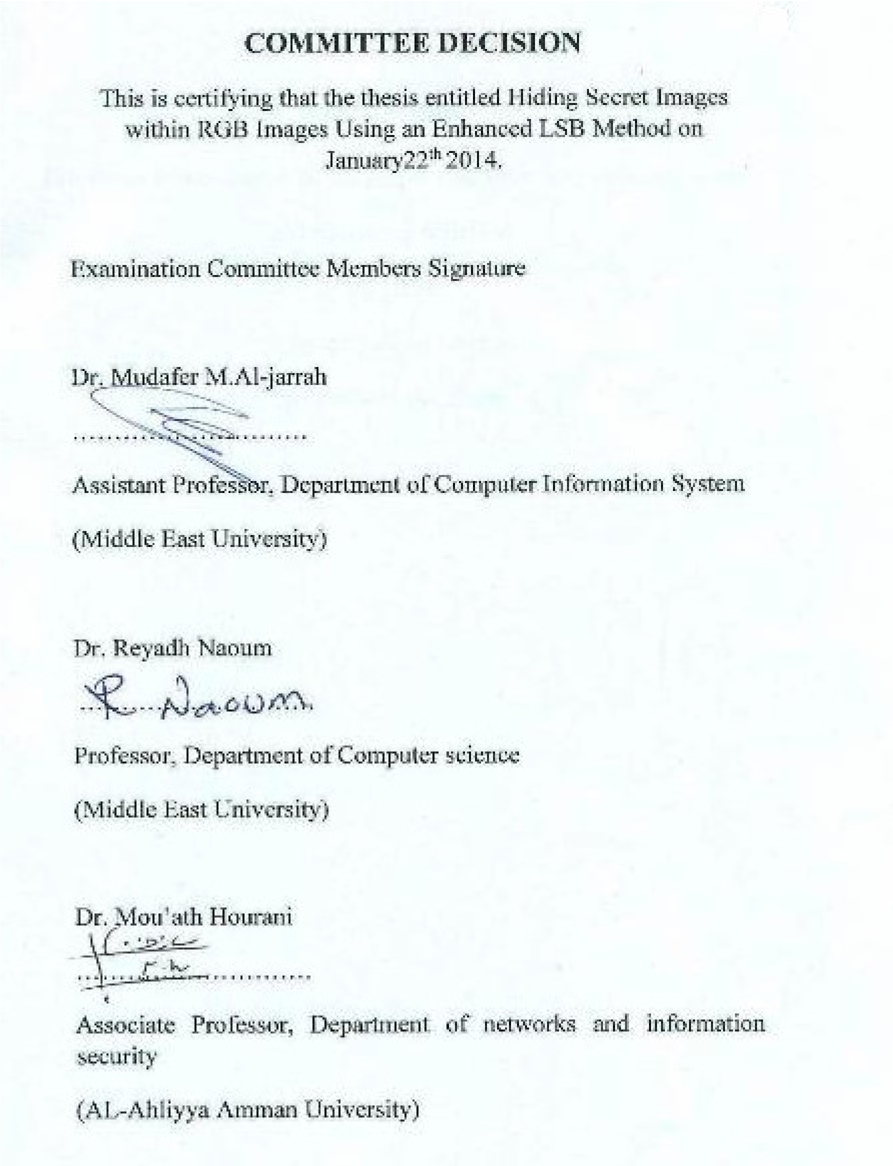
إقرار تفويض

أنا محمد زياد الحاج قاسم أفوض جامعة الشرق الأوسط بتزويد نسخ من رسالتي للمكتبات أو المؤسسات أو الهيئات المعنية بالأبحاث و الدراسات العلمية أو الأفراد عند طلبها.

الاسم:محمد زياد الحاج قاسم

التاريخ : ٢٦/٥/٢٠١٤

التوقيع: محمد الحاج



COMMITTEE DECISION

This is certifying that the thesis entitled Hiding Secret Images within RGB Images Using an Enhanced LSB Method on January22th 2014.

Examination Committee Members Signature

Dr. Mudafer M.Al-jarrah

……………………………

Assistant Professor, Department of Computer Information System

(Middle East University)

Dr. Reyadh Naoum

R. Naoum

Professor, Department of Computer science

(Middle East University)

Dr. Mou'ath Hourani

……………………………

Associate Professor, Department of networks and information security

(AL-Ahliyya Amman University)

Dedication

This thesis is dedicated to all the people who never stop believing in me

and who along with God

My father

The spirit of my mother

My brothers and sisters

Acknowledgements

I would like to thank my father for his continuous support during my study.

I also would like to thank my great supervisor Dr. Mudhafar Al-Jarrah for his support, encouragement, proofreading of thesis drafts, and for helping me throughout my studies, putting me in the right step of scientific research. I would like to thank the Information Technology Faculty members at the Middle East University. I would also like to thank all of my family members.

Abstract

The stenography method is one of the oldest methods used in protecting the secrecy of information through hiding it in a carrier that does not show its contents. Modern steganography techniques rely mostly on storing a secret message inside a cover image file of various file types such as bmp, tiff, png, and jpg, in grey and RGB formats. The research in this thesis deals with hiding compressed images of the jpg type inside un-compressed RGB images (tiff, bmp, png) with the objective of increasing the hiding capacity, in order to hide larger images, while maintaining un-detectability quality of the stego image.

The proposed work is based on the spatial method, extending the LSB method to store 4 bits (half-byte) in each color byte of the RGB channels, thereby crossing the limit of 3-bits that is considered as the limit of un-noticeable change to a color channel.

Two algorithms are presented. The first algorithm (Embed-All) stores the hidden image in the RGB channels of successive pixels, i.e. odd and even pixels. This algorithm gives a hiding capacity of 50% of the available pixel capacity, by hiding secret data in 12 bits per pixel (bpp).

The second algorithm (Embed-Odd) stores the hidden image in the RGB channels of the odd pixels, while changing RGB channels of even pixels by adding or subtracting the difference between the secret image half-bytes, and the LSB half-bytes of the odd pixels.

The purpose of this change is twofold, to neutralize the color change in the odd pixel, and to add noise to the even pixel in order to confuse the attacker.

The two algorithms were implemented in Matlab 2012b, and a series of experiments has been carried out to evaluate the proposed algorithms. Standard images such as Lena have been used as cover, and for secret images, the choice was for jpg images of various sizes, up to the maximum hiding capacity of the cover images. The reported result of hiding in half-bytes of RGB channels does not show any noticeable difference to human eye, even for the successive pixels method (Embed-All). The use of comparison statistics such as PSNR has shown acceptable distortion values even when hiding to the maximum capacity of an image.

The proposed work maintains quality of the extracted images, they are recovered unchanged. To test integrity of the recovered images, the PSNR metric was used which gave in all cases the value of infinity, i.e. zero change. Another set of tests were carried out to test correct recovery, including hiding audio and video media, to ensure that the recovered audio and video will work as before, for the same duration, and nested embedding, hiding image inside image for three levels, which resulted in correct recovery of the initially embedded image.

The thesis presents conclusions and suggestions for future research.

الملخص

طريقة الاختزال هي واحدة من أقدم الطرق المستخدمة في حماية سرية المعلومات من خلال إخفائه في الناقل الذي لا تظهر محتوياته. تقنيات إخفاء المعلومات الحديثة تعتمد في معظمها على تخزين رسالة سرية داخل ملف صورة الغلاف من أنواع الملفات المختلفة مثلBMP ، TIFF، PNG، JPG، GIFوبأشكال الرمادي و RGB العمل في هذه أطروحة يختبئ الصور المضغوط من نوع JPG في صور غير مضغوطه من نوع (BMP, PNG) بهدف زيادة سعة الاخفاء، من أجل إخفاء صور ذات حجم أكبر، مع الحفاظ على جودة الصورة الغطاء وخاصية عدم الكشف عن وجود صورة متضمنة .

ويستند العمل المقترح على توسيع طريقة التخزين في اخر 4 بت في كل بايت LSBمن من القنوات RGB في وبالتالي عبور حدود 3 بت التي تعتبر الحد الاعلى في استخدامها لاخفاء البيانات. في هده الاطروحه تعرض اتنين من الخوارزميات الخوارزميه الاولى تضمين البيانات في البكسل الفردي والزوجي حيث يخزن الصور المخفيه في قنوات RGB بحيث نخزن في كل بكسل 12 بت اي تعطينا هذه الخوارزميه قدرة اختباء تصل 50% من السعه المتاحه من اليكسل. اما الخوارزميه الثانيه تضمين البينات في القناة البكسل الاول والتغير على البكسل الثاني من حيث الاضافه والطرح بحيث نوازن من حيث الالوان ومن اجل زراعه تشويش حيث نستخدم في اخفاء المعلومات 6 بت بت في كل بكسل اي تعطينا نسبة تخزين 25% من سعة البكسل المتاحه.

تم تنفيد عمل الخوارزميات على برنامج Matlab وتم اجراء سلسله من التجارب في تقيم الخوارزميات المقترحه واستخدمنا صور قياسيه مثل لينا حيث استخدمت كغطاء والصور اسريه استخدمنا صور من نوع JPG من مختلف الاحجام و تصل الى القدره القصوى لاختباء. لاتظهر اي تاثير او اختلاف ملحوظ على العين البشريه عند استخدام نصف بايت في التخزين حتى بالنسبه لخوارزميه التضمين المتعاقب. وقد اظهرت الاحصاءات المقارنه مثل PSNR نسبه مقبوله حتى عند اخفاء السعه القصوى في الصوره.

في الخوارزميه المقترحه يحافظ على جودة الصورة المستردة حيث يتم استرداد الصوره المخفيه دون التغيير في محتواها ، ولاختبار الصور المستردة تم استخدام معيار التشابه PSNR ، حيث كانت قيمة المعيار لجميع الحالات المستردة هي اللانهاية اي لايوجد أي إختلاف بين الصورة الاصلية والمستردة .

ونفذت مجموعة أخرى من الاختبارات أجريت لاختبار سلامة الاسترجاع التام للخوارزميتين وشمل ذلك على :

أ. إخفاء ملفات متعددة الوسائط (صوت وفيديو) ، للتأكد من أن ملفات الصوت والفيديو المسترجعة ستعمل كما كانت من قبل، ولنفس فترة التشغيل.

ب. الاخفاء المتداخل ، بأخفاء صورة داخل صورة لثلاثة مستويات ، وأسترجاع الصورة االمخبئة ومقارنتها مع الصورة الاصلية بأستخدام معايير التباين والتشابه.

تنتهي الاطروحة بأستناجات وتوصيات لاعمال مستقبلية مستندة لنتائج البحث الحالي .

Table of Content

List of Figure

List of Table

List of Abbreviations

AE	Absolute Error
BPP	Bits Per Pixels
BMP	Microsoft Windows Bitmap
COVA	Cover Array
DF	Difference
ECB	Even Pixel Cover Byte
ELSB	Even Least Significant Bit
GIF	Graphics Interchange Format
HBA	Half-Byte Array
HB	Half-Byte
JPEG	Joint Photographic Experts Group
JSTEG	Java Steganography Project
KB	Kilo Byte
LSB	Least Significant Bit Insertion
MAE	Mean Absolute Error
ME	Mean Error
MSB	Most Significant Bit
MSE	Mean Square Error
OCB	Odd Cover-Byte
OLSB	Odd Least Significant Bit
PNG	Portable Network Graphic
Pi	Pixel

PSNR	Peak Signal to Noise Ratio
RGB	Red-Green-Blue
RM	Recovered Message
SHB	Secret Half-Byte
ST	Stego-Image
RLSB	Recover least Significant Bit
RMSB	Recover Most Significant Bit
TIFF	Tagged Image File Format

Chapter One

Introduction

Chapter One

Introduction

1.1. Background

Information or knowledge is power, as often said, and protecting the private information of individuals or groups, from access by adversaries or competitors has always been an area of interest and investigation. The Second World War and the cold war after have resulted in great research in cryptography and cryptanalysis. Strong encryption algorithms have been developed but time has shown that every encryption method can be broken (Cheddad et al., 2010). .

In recent years an alternative approach for protecting information has gained momentum and interest, it is the approach that attempts to hide the existence of information, thereby reducing the possibility that it will be attacked, this approach is steganography, an ancient method of hiding information that has been revived lately Shreelekshmi et al, 2010).

Basically, cryptography scrambles a message into a code to obscure its meaning, while Steganography hides the message completely (Cheddad, 2010). These two secret communication technologies can be used separately or together—for example, by first encrypting a message, then hiding it in another file for transmission (Parvez et al., 2008).

Steganography is a technology that is used to hide secret information in digital media, thus hiding the fact that secret communication is taking place (Jamil, 1999). By hiding secret information in less suspicious digital media, well-known channels, for example e-mail and social networking sites, are avoided, thereby reducing the risk of information being leaked in transit, should an attacker attempt to intercept the communication through a man-in-the-middle attack, he would have no reason to suspect that he has intercepted anything more that an innocent image (Artz, 2001).

Unlike a word-processed file where you're likely to notice letters missing here and there, it is possible to alter graphic and sound files slightly without losing their whole viability for the viewer and listener. With audio files, you can use bits of the file that contain non- audible sound to the human ear. With graphic images, you can get rid of redundant bits of color from the image and still produce a picture that looks intact to the human eye and is difficult to discern from the original (Cole et al., 2003).

Images are composed of picture elements, i.e. pixels. There are three major classes of images: 1) black and white – each pixel is composed of a single bit and is either a zero or a one, representing either white or black; 2) Grayscale – each pixel is composed of 8 bits (in rare cases, 16 bits) which defines the shade of grey of the pixel, from zero (black) to 255 (white); 3) Full color – also called 24-bit color as there are 3 primary colors (red, green, blue), each of which is defined by 8 bits. There are over 16 million possible colors in a 24 RGB image. There exist many other representations, but these three, by far, are the most common. For the steganography techniques presented here, we will either use grayscale or 24-bit color. Considering 8-bit grayscale, each pixel has 2^8 = 256 possible levels of grey, ranging from black to white. Each bit does not contribute the same amount of information. The Most Significant Bit (MSB) contributes ½ the information, while the Least Significant Bit (LSB) contributes 1/256th of the information. So, changing that LSB only affects 1/256th of the intensity and humans simply cannot perceive a difference (Marwaha, 2010).

1.2. Problem Definition

High capacity hiding of digital media, in particular digital images, inside other digital images, using the spatial approach (Wang, 2005), is the problem area of research in this thesis. A fundamental constraint that is taken into consideration in this research is that there should be no loss of data in the process of hiding and recovering of data during the hiding process. The research will deal mainly with hiding compressed digital images of the JPEG format (Chang et al., 2002), inside larger uncompressed Red-Green-Blue (RGB) images such as BMP, PNG or TIFF. Secret digital images such as photographs, industrial design diagrams, maps and scanned personal documents can be stored inside other images. Hiding a secret image within a cover image might result in the distortion and degradation of the cover image. The problem to be dealt with in this work is the balancing between storage capacity of image within image, and the perception (visibility) of possible cover image quality degradation resulting from secret image embedding within cover images.

Questions to be addressed

1- Can we embed a secret image inside another image without leaving a clear visible impact on it?

2- Can we store compressed images without having to un-compress them?

3- Can we increase the hiding capacity without a drop in image quality metrics ?

4- Can we recover hidden images without any loss of data?

5- How can we modify the LSB method for RGB images to achieve 1-4 above?

1.3. Objectives

The goal of this research is to hide large compressed digital images inside other digital images, allowing the increase of storage capacity without degradation to cover image quality, and without any loss of data during the hiding process. In other words; the main objective of this research is the enhancement of the LSB technique to get better results in balancing image capacity vs. quality while maintaining secret image integrity. The proposed changes to the LSB technique aim to increase storage capacity needed for storing images, by using a half-byte (half-octet) of each color channel of an RGB pixel, without the decrease in image quality.

1.4. Significance of the problem

Most previous works have dealt with hiding small text files (1 or 2 KB) or small images (less than 50 KB). On the other hand real documents such as technical drawings, scanned images, high resolution photographs, and maps tend to be much larger. To increase hiding capacity of image within image, some researchers have combined image compression with steganography, as in the work of (Jain and Ahirwal, 2010). The compression results obtained were loosy, which means that the secret image or document will not be fully recovered. The significance factors of this research is that it is addressing higher hiding capacity while maintaining both cover image quality and full secret image recovery.

1.5. Motivation

"**A picture is worth a thousand words**" refers to the notion that a complex idea can be conveyed with just a single still image. Images may contain a wealth of information for individuals; enterprises, industry and governments, and the falling of images into the wrong hands can lead to serious security problems (Selvi et al., 2012).

The increasing need and demand for the privacy and protection of image documents exchanged over the Internet or inside enterprises and organizations; have had a positive impact on research and development in the field of steganography. Criminals are using all the possible means to get access to private documents through illegal ways. To this end it is felt that there is a great potential for enhancing the hiding of secret images within innocent-looking cover images, with good tolerance for visual and steganalysis attacks (Reddy et al., 2011).

1.6. Methodology

The work will involve the following tasks:

- Studying of literature in steganography and steganalysis.
- Developing an LSB based steganography technique that aims to increase hiding capacity, reduce image degradation and improve resilience to stegananlysis.
- Implementing the proposed techniques using the Matlab 2012b environment.
- Experimenting with various standards RGB images (PNG, TIFF, and BMP) as cover images, to embed JPEG images of various sizes and resolutions.
- Experimenting with the hiding of other digital media such as audio and video.
- Evaluating the resulting stego-images visually and statistically.
- Discussing the results and drawing conclusions.

1.7 Thesis Organization

The thesis contains seven chapters:

Chapter 1 Provides an Introduction.

Chapter 2 Literature survey.

Chapter 3 Steganography Evaluation Criteria.

Chapter 4 Proposed Models.

Chapter 5 Experimental work and Discussion of Results.

Chapter 6 Conclusion and Future Work.

Chapter Two

Literature Survey

Chapter Two

2.1 Introduction

Steganography is derived from the Greek word "steganos" which means covered or secret , and graphy is writing or drawing; in other words we say it is the art of communicating a secret message, from one person to another in a way that no-one can say that a secret message exists. In simple word; Steganography is hidden writing, whether it consists of invisible ink on paper or copyright information hidden in an audio, image or video file. This is done by hiding the secret message within a non-sensitive one, and the recipient should believe that the non-sensitive message can see is all there is. Steganalysis, on the contrary, is recipient's task of detecting the presence of a secret message when senders employ Steganography.

Schaathum presented in his book (Schaathun, H. G, 2012) the following scenario to explain the concept of steganography"Steganography is the art of communicating a secret message, from Alice to Bob, in such a way that Alice's evil sister Eve cannot even tell that a secret message exists. This is (typically) done by hiding the secret message within a non-sensitive one, and Eve should believe that the non-sensitive message she can see is all there is. Steganalysis, on the contrary, is Eve's task of detecting the presence of a secret message when Alice and Bob employ steganography".

Cryptography is often used to protect information secrecy through making messages unreadable. However, unreadable messages may raise an opponent's suspicion and probably lead to his destruction of such a communication manner. Therefore, Steganography gets a role on the stage of information security. Steganography refers to the technique of hiding information in digital media in order to conceal the existence of the information. The media with and without hidden information are called stego media and cover media, respectively. Steganography can meet both legal and illegal interests. For example, civilians may use it for protecting privacy while terrorists may use it for spreading terroristic information (Narayana and Prasad, 2010).

Nowadays, Steganography is most often associated with the high-tech variety, where data is hidden within other data in an electronic file. For example, a Word document might be hidden inside an image file. This is usually done by replacing the least important or significant bits of data in the original file—bits that are hardly missed by the human eye or ear—with hidden data bits (Almohammad and Ghinea, 2010).

2.2. Steganography Classification

These days people all over the world are transferring images and audio over the Internet for genuine reasons of leisure or business, while at the same time many of transferred images and multimedia files are in fact carriers of hidden information (Zhang et al, 2010).

The hiding of information can be categorized into four categories:

1. Text Steganography.
2. Image Steganography.
3. Audio/Video Steganography.
4. Protocol Steganography.

Almost all digital file formats can be used for steganography, but the formats that are more suitable are those with a high degree of redundancy, redundancy can be defined as the bits of an object that provide accuracy far greater than necessary for the object's use and display. The redundant bits of an object are those bits that can be altered without the alteration being detected easily, image and audio files especially comply with this requirement, while research has also uncovered other file formats that can be used for information hiding (Bender et al., 1999). Figure (2-1) shows different uses to hide information.

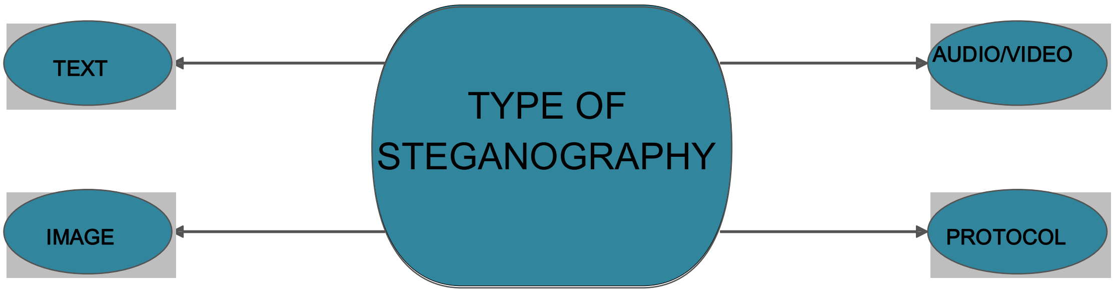


Figure (2-1) Categories of Different Types of Steganography

Hiding information in text:

Information can also be hidden in text files. The most popular method is to hide a secret message in every nth letter of every word of a text message. A variety of different techniques exist of hiding data in text files. Text steganography using digital files is not used very often since text files have a very small amount of redundant data. (Morkel et al., 2005)

Hiding information in images:

It is possible to use an image as a cover to hide various types of data such as text, photos and audio. A picture has so much redundancy in pixels, whose removal as in compression or use for hiding as in steganography will not have a noticeable difference to the human eye (Morkel et al., 2005).

Hiding Information in Audio Files:

Audio files can also be used for hiding secret data. One different technique unique to audio steganography is masking, which exploits the properties of the human ear to hide information unnoticeably. A faint, but audible, sound becomes inaudible in the presence of another louder audible sound (Morkel et al., 2005). This property creates a channel in which to hide information. The larger size of meaningful audio files makes them less popular to use than images (Cole and Chairperson-Grossman, 2004).

Hiding Information in Protocols:

The term protocol steganography refers to the technique of embedding information within messages and network control protocols used in network transmission. In the layers of the OSI network model there exist covert channels where steganography can be used (Chang and Tseng, 2004).

2.3. Principles of information security

Steganography, as a secret communication method, achieves most of these requirements since there is no method that can address all security concepts (Cole, 2003). Therefore, the key concepts of security that apply for steganography and equally take into consideration the main principles of information security requirements (discussed above) are as follows (Cole, 2003):

1. Confidentiality: Cryptography achieves the confidentiality by preventing unauthorized persons, who can see the information, from gaining access to this

information. With steganography, unauthorized people do not even know there is secret data there.

2. Survivability means that all data processing takes place between sender and receiver does not destroy the hidden information. Additionally, this received information must be extractable and readable.

3. No Detection: Steganography fails if someone can easily detect where you hid your information and find your message. Therefore, even if someone knows how the steganography method embeds the secret information, he or she cannot easily find out that you have embedded data in a given file.

4. Visibility: The stego file must be undetectable and there must be no visible changes to the stego file.

The main goal of steganography is achieving confidentiality by embedding data. Unlike cryptography which scrambles the content of the secret data, steganography hides the very existence of this data. Therefore, unauthorized people do not even know there is secret data there. From a confidentiality standpoint, steganography provides a higher level of information protection than cryptography (Cox et al., 2008).

2.4. Steganographic operations

A Steganography system is divided into two operations as stated below:

1- Embedding process: storing the secret message inside redundant parts of the cover media without leaving a noticeable impact (Cheddad et al, 2010).

2- Extraction process: recovering the hidden secret message from the stego file, without losing or distorting the hidden message (luo et al, 2010).

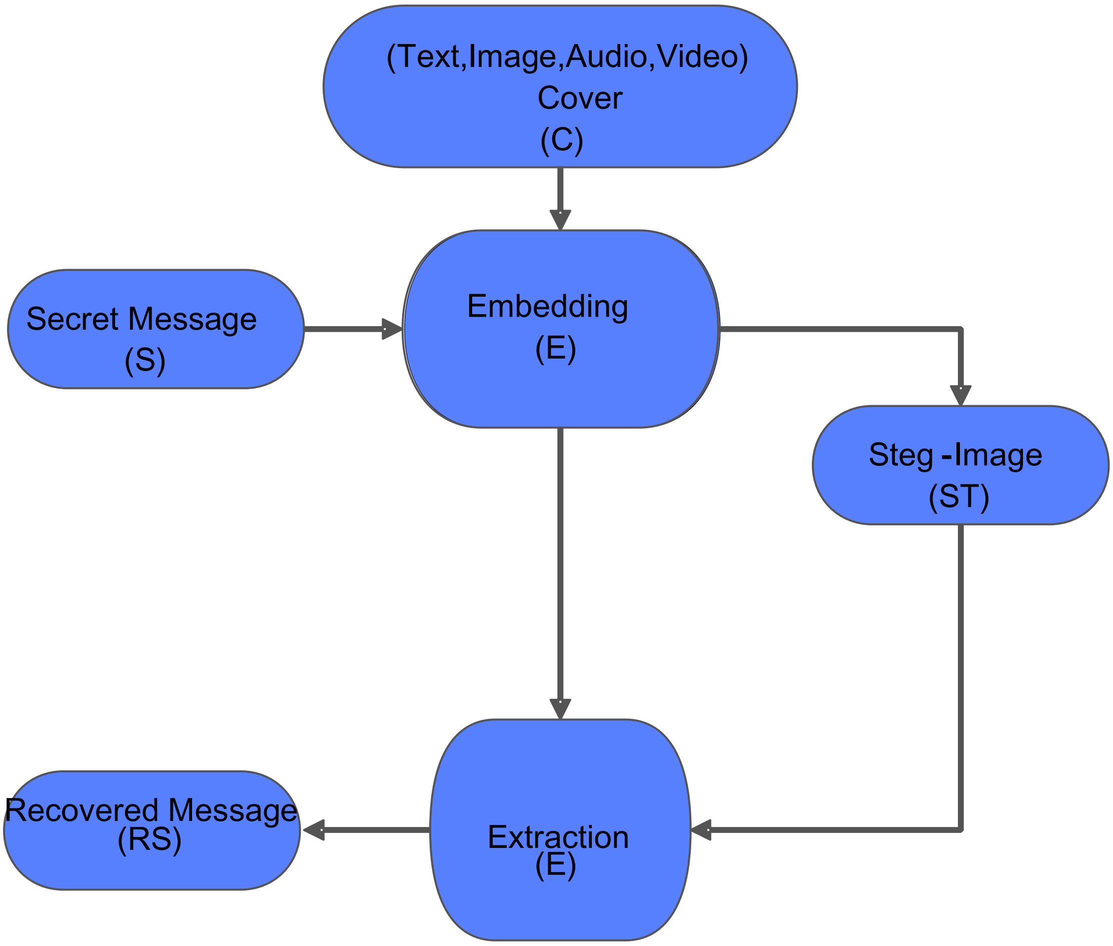


Figure (2-2) Overview System Steganographic

2.5. Steganalysis techniques

Steganalysis can be defined as the science and art of detecting and often decoding secret messages hidden within stego files (Artz, 2001). Nonetheless, steganography is considered broken if merely the presence of secret data within a stego file is detected, even if the secret message is not decoded. The increasing number of steganography techniques available has stimulated steganalysis research.

Thus, the significance of reliable detection techniques is increasing. In addition, it is suggested that such steganalysis techniques should be included in every virus-detection program in the future (Fridrich et al., 2001). Basically, most steganographic systems leave behind (in the stego files) some traces, so these traces make these files detectable even though these traces are indiscernible by humans. Generally, modifying some parts of a cover file changes the properties of this file in some way. Therefore, this can be a sign that there is a hidden message within this stego file (Provos and Honeyman, 2003). Therefore, a simple comparison between a stego file and its corresponding cover file may reveal the existence of a hidden message within this stego file. In order to avoid such a comparison, cover files used should not be publicly available or should be destroyed after usage, since the absence of cover files represents the weakest form of steganalysis (stego only attack) (Artz, 2001).

2.6. Cover Files used for Steganography

Basically, cover files represent the container of hidden data or secret messages. Additionally, some parts or characteristics of cover files will be modified, changed, or manipulated in order to hide these secret messages. However, these manipulations, which occur during the hiding procedure, should remain imperceptible to anyone not involved in the communication process. Therefore, the appearance or format of cover files must remain intact after hiding the secret data. As a result, it is not possible to use all types of files or data as cover files of steganography since different file types have different redundant area which can be replaced by the secret data (Katzenbeisser and petitcolas, 2000).

2.7. Related Work

Koppole (2009) presented a steganography schema which achives 100% hiding capacity, by using loosy compression with spatial steganography. The compression in voles changing the image from RGB to YIQ color model, then hiding the compressed image in 13 bits of pixels of an RGBA image. The limitation of this schema is the loos of data due to compression. Also the compression technique is limited in capacity compared to JPEG compression.

The field of steganography and steganalysis has been recently surveyed by Li et al. (2011) where commonly used strategies for improving steganographic security and enhancing steganalytic capability are summarized and possible research trends are discussed.

Al-Husainy (2012) proposed the idea to enhance the performance of the Classic-LSB image steganography technique. The message segmentation LSB image steganography technique was suggested here by splitting the long secret message into number of short segments. Then hide these short segments in different parts of the best matched LSB in the pixels of the stego-image.

Gangwar & Shrivastava (2013) proposed a method that LSB based image steganography using secret key which provides good security issue than general LSB based image steganography methods. It shows that the proposed method is an effective way to integrate hidden information reporting and it is very difficult for the unauthorized users to identify the changes in stego image. In this method, they used a secret key to hide hidden information into cover image the LSB of RGB bits. This process provides a new dimension for image steganography. It is very difficult to recover the hidden information for third party without knowing the secret key.

Luo & Huang (2010) pointed out that usually there exists some smooth regions in natural images, which would cause the LSB of the cover images not to be completely random or even to contain some texture information just like those in higher bit planes. To preserve the statistical and visual features in cover images, they have proposed a novel scheme which can embed the secret message into the sharper edge regions adaptively according to a threshold determined by the size of the secret message and the gradients of the content edges. The experimental results evaluated on 6000 natural images using different kinds of steganalytic algorithms show that both visual quality and security of the stego images are improved significantly compared to typical LSB-based approaches.

Bashardoost et al. (2013) implemented a method that the enhanced LSB method saves up to forty percent of capacity because of exploiting compression technique. Therefore, little number of pixels of the image will be probably modified and consequently the quality of the stego image will be improved. In addition, smaller amount of data will be distorted whereas the third party applies active visual attacks on the Stego image. The possibility of extracting the content of hidden data reduces significantly, when the private message becomes encrypted. Fix the weakness of Simple LSB system by providing some enhancements. Managed The Enhanced LSB method utilizes three fundamental improvements specifically Knight Tour embedding algorithm, encryption and LZW compression. The process starts with the encoding the confidential information by using encryption technique. Both of the sender and receiver have a secret key which is used in encryption and decryption phases. Afterward, the LZW compression technique reduces the size of encrypted data to improve the payload capacity.

Singh et al (2007) created across platform that can effectively hide a message inside a digital image file. As there are many application of image steganography like it allows for two parties to communicate secretly. They investigated whether taking the image as the cover into account increases the security of the message by creating cross- platform self evaluating tool. Also described the benefits from the approach like the security of message increases and distortion rate has reduced.

Laskar & Hemachandran (2012) employed a method for applications that require high-volume embedding with robustness against certain statistical attacks. This method is an attempt to identify the requirements of a good data hiding algorithm and it is not intended to replace steganography or cryptography but rather to supplement it. If a message is encrypted and hidden with a LSB steganographic method the embedding capacity increases and thus it is possible to hide large volume of data and the method satisfies the requirements such as capacity, security and robustness which are intended for data hiding. The resulting stego-image can be transmitted without revealing that secret information is being exchanged. If an attacker was to defeat the steganographic technique to detect the message inside the stego-object, the attacker would still require the cryptographic decoding key to decipher the encrypted message. The main aim is to develop a system with extra security features where a meaningful piece of text message can be hidden by combining two basic data hiding techniques.

Chapter Three

Steganography Evaluation Criteria

Chapter Three

3.1. Introduction

Stegangraphy styles and techniques can be evaluated using multiple factors for evaluation. (Lashkari et al., 2011) proposed six-factor steganography evaluation scheme which included the following criteria: Capacity**,** Perception (Visibility), Robustness, Detectability, Dependency and Domain. Detectability is the subject of on-going Steganalysis research, which uses statistical methods such as Histogram Analysis. One criterion that has received considerable interest in steganalysis is the PSNR (Peak Signal Noise to Ratio), which is a measure distortion resulting from embedding (Cole et al., 2003).

There are many techniques and algorithms used in Steganography such as F5, LSB, Java Steganography (JSteg), etc. We will focus on the LSB algorithm. LSB stands for (Least Significant Bit) which is a replacement process in which bits from the secret document replaces least significant bits from pixels of the carrier image. It is a simple approach for embedding a message into a cover document (Cheddad et al., 2010).

3.2. Evaluation of Image Steganography

In this section, we discuss the evaluation criteria for the image hiding algorithms: Invisibility, Undetectability, Robustness, and Security attacks against image manipulation in the form of the statistical possibility of discovery.

3.2.1 Golden Square

1. Robustness:

The ability to extract hidden information after common image processing operations: linear and nonlinear filters, loss compression, contrast adjustment, recoloring, resampling, scaling, rotation, noise adding, cropping, printing / copying / scanning, D/A and A/D conversion, pixel permutation in small neighborhood, color quantization (as in palette images), skipping rows / columns, adding rows / columns, frame swapping, frame averaging (Sarkar et al., 2010).

2. Un-detectability:

Impossibility to prove the presence of a hidden message this concept is inherently tied to the statistical model of the carrier image. The ability to detect the presence does not automatically imply the ability to read the hidden message. Un-detectability should not be mistaken for invisibility – a concept related to human perception (Stamm et al., 2010).

3. Invisibility:

This image evaluation criteria is related to perceptual transparency, and it is based on properties of the human visual system (Nag et al., 2010).The evaluation is carried out through human inspection, where evaluation are asked to compare an original image with a stego image in order o test whther a distortion can be detected by the human eye.

4. Security:

The embedded information cannot be removed beyond reliable detection by targeted attacks based on a full knowledge of the embedding algorithm and the detector (Except a secret key), and the knowledge of at least one carrier with hidden message (Pevny and Fridrich, 2010).

3.3. Active Attack

This involves destroying the hidden message and is more prevalent in such technologies as digital watermarking where the main purpose is to remove the mark or render it useless. Active attacks are also useful in situations where steganography is suspected to be in use but discovering the hidden message is unimportant. It works in that all objects are modified in such a manner that the object still appears to be the same but any hidden bits of information will be void. A good example is with images whereby a certain digital effect can be applied to the image without any human noticeable change but will change the bits of the embedded secret message and render it unrecoverable (Sallee, 2005).

3.4. Passive Attack

A passive attack involves detecting the use of steganography and is a prelude to actually deciphering the hidden message (Cheddad et al., 2010).

Methods of steganalysis include:

- Viewing the file
- Listening to the file
- Performing comparisons on a file (if you have the original)
- Statistical Attack – this involves detecting changes in the patterns of pixels of LSB.

3.5. Type of Steganalysis Attacks

1-Visual attacks: Visual analysis is defined as the process of detecting hidden messages in stego files through inspection by naked eye or by assistance of a computer. Therefore, the visual attack represents one of the easiest steganalysis methods (Wang, 2004). Visual attacks examine the entire stego file (i.e. image) or only the LSB of this file in order to detect any alteration or irregularity. Thus, steganography methods that leave some kind of trail or signature would be vulnerable to such attacks. these signatures could be: (i)adjacent pixels in an image have very different colors, (ii)the number of an image colors has drastically been increased or decreased,(iii)the image size has been changed, and the image quality has been modified (Bailey et al., 2004).

2-Statistical attacks: Statistical analysis relies on examining the contents of files. Moreover, this kind of attacks is more powerful than the visual attack since it reveals even tiny modifications which have occurred in the statistical properties of files (Artz, 2001). The statistics of a file may reveal that it has been modified in some way but this doesn’t specify which technique was used for modification. This represents one of the difficulties of the statistical analysis (Watters et al., 2005).

3.6. Evaluation of Steganographic Capacity (Payload)

Since the main application of information hiding and steganography is the secret communication, it is important to determine how many bits a steganographic system can embed imperceptibly in comparison to the other methods. Therefore, evaluating the capacity of a steganography technique means to find the maximum number of bits that can be replaced in a cover image without compromising UN-detect ability.

It is mentioned before, that there is a tradeoff between the steganographic hiding capacity and imperceptibility. Nevertheless, steganography techniques that embed larger size messages in cover files and introduce more distortion to stego files are considered as worthless systems. On the other hand, increasing the steganographic capacity and maintaining an acceptable level of stego image quality is considered a good contribution. Additionally, improving the stego image quality while maintaining the steganographic capacity is also considered a significant Contribution (Wu and Hwang, 2007).

3.7. Objective Quality Evaluation

In the literature, the peak signal-to-noise ratio metric (PSNR) has shown the best advantage almost overall objective image quality metrics under different image distortion environments and strict testing conditions (Wang et al., 2002). Indeed, PSNR and the MSE metrics are the most common measures used to evaluate the quality of image coding and compression (Costa and Veiga, 2005).

3.8. Quality Evaluation Metrics

PSNR and MSE are the most common and widely-used full-reference (FR) metrics for objective image quality evaluation. Furthermore, PSNR is used in many image processing applications and considered as a reference model to evaluate the efficiency of other objective image quality evaluation methods (Wang et al., 2002).

The PSNR (Peak Signal-to-Noise Ratio), is a tool for measuring the distortion between the original and the recovered signals, “recovered” refers to either operation, decompression, reconstruction or any other engineering manipulation. It is evaluated in logarithmic decibel scale (KHMOU Youssef, 2005).

The MSE metric formula in RGB images is:-

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

Where m and n are the image pixel resolution (width and height)

The PSNR is calculated as:-

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

$$= 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

Other metrics that can be useful in stego evaluation are:

- AE (absolute error count) it is simply the number of pixels that have been changed, regardless of the amount of change (Gui and Yang, 2012). This metric is useful in the comparison of an embedded image and the extracted image. If the AE value is greater than zero, this means that extracted image has been changed during the steganography process.

- MAE (mean absolute error) this metric measures the average of absolute value of change between original and stego image pixels:

$$MAE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}|I(i,j) - K(i,j)|$$

It serves the purpose of determining the absolute amount of change is pixels (Zhang and Ping 2003).

- ME (mean error).This metric is a new metric proposed in this thesis it is the accumulated difference between all pairs of pixels original and stego image, without taking the absolute value as in MAE this metric is measure as :

$$ME = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(i,j) - K(i,j)$$

The idea behind this new metric, is that a negative color change in one channel should cancel (neutralize) the effect of a similar but positive change in the same channel .This metric is useful in evaluating stego images using the new proposed Change-Even Algorithm.

Chapter Four

The Proposed Model

Chapter Four

4.1. Overview

The proposed research deals with hiding data in the spatial domain. In particular it is dealing with hiding of secret images inside RGB image pixels, using two methods:

- Storing secret data in odd and even pixels (successive pixels method).
- Storing secret data in odd pixels and adjusting the even pixels to balance changes in the odd pixels.

4.2. The Proposed Model

The proposed model aims to increase the hiding capacity of cover images. Image data will be hidden in color channel of the RGB pixels, in odd or odd and even pixels. The sizes of secret and cover images will be taken into consideration when selecting the pixels for embedding. To balance the change in a color channel of a pixel, the color of the equivalent channel in the adjacent pixel will be adjusted by the amount of change to the embedding pixel, thereby to visually neutralize the effect of the change due to embedding. This method will be compared with the alternative of embedding data in successive pixels. The unit of change will be a half-byte, the low half-byte (LSB) of each byte of a color channel. The upper half-byte (MSB) will not be changed in the embedding process.

4.3. Principle of Operation of the Proposed Algorithm Embed-Odd

The purpose of this algorithms is to hide in odd pixels of the cover image and adjust (odd or subtract) from the bytes of the even pixels the differences between secret half-bytes and LSB half-bytes of the odd pixels.

Several operations to be implemented, as in steps below (an example in Fig. illustrates how the operation are done):

1- Get next secret half-byte (SHB) of a byte from the secret image.

2- Get the lower half-byte (LSB) of the next color channel of the current pixel (PI).

3- Calculate the difference (DF) between SHB and LSB.

4- Replace cover LSB with SHB.

5- Adjust the same color byte of the next pixel (PI+1) with the difference DF.

6- After changing the lower half-bytes of the three channels of current pixel, select the next embedding pixel (PI+2).

The rate of bits per pixels (payload) = 6 bpp. Hiding various types of digital media of 200 KB will need 800 KB of cover pixels' bytes.

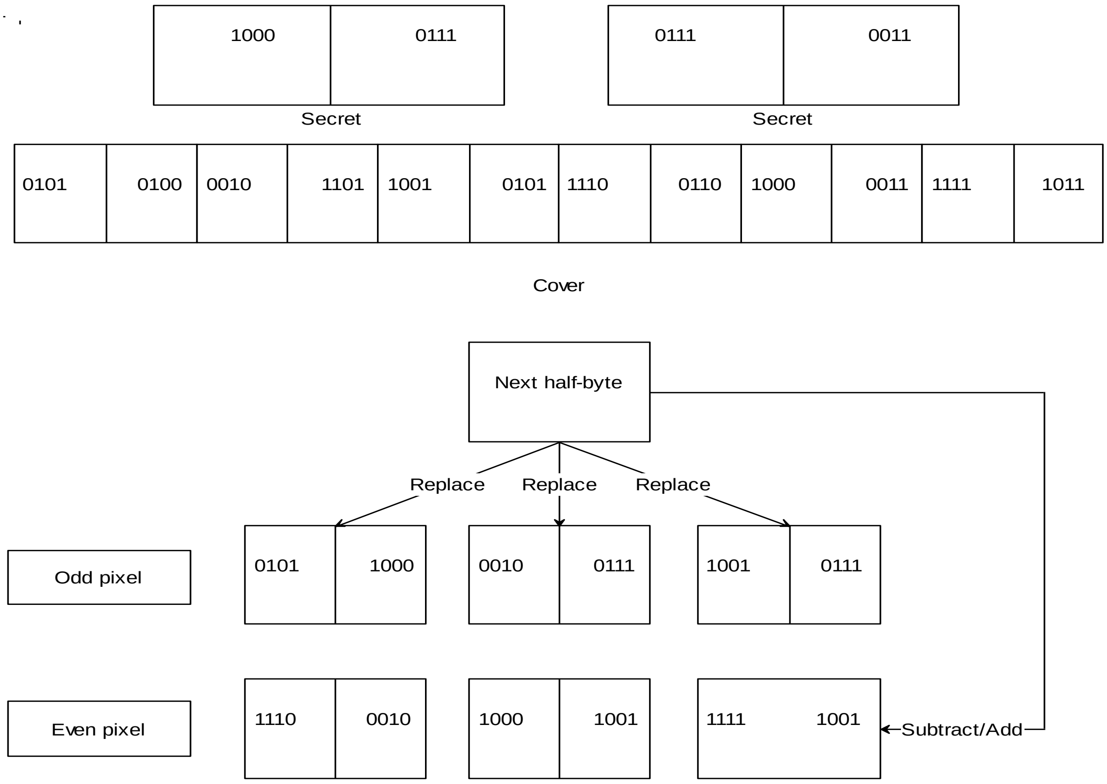


Figure (5-1) an Example Illustrating the Embedding Change Process

4.3.1 Embed-Change Algorithm

Embed-Change Embedding Algorithm

Input: Secret Message, Cover-image.

Output: Stego-image.

Processing: to embed the secret image in half-bytes of RGB channels, in odd pixels, and to add or subtract the difference between odd pixels and secret to even pixels. Hiding capacity is 25% of the available bytes of the selected pixels, which is equal it to 6 bits-per-pixels (bpp).

Algorithm steps:

```
Open secret image file (jpg, bmp, tiff, mp3, mp4, pdf format) in binary format
Store secret image file in secret array
Convert secret image to half-bytes in half-byte array (HBA)
HBCount = size of secret half-bytes
Open cover file (BMP or TIF format)
Read cover image into cover array (COVA)
Get height and width of cover array (Height and Length)
Note: The following steps Process Secret half-byte array, store secret half-bytes in odd pixels, add / subtract differences from even pixels.
For i = 1 to Height
For j= 1 to Width -1 Step 2
For k= 1 to 3
Get next secret half-byte (SHB) (if end of array break)
Get next odd cover-byte (OCB = COVA (I, j, k))
Get LSB of OCB and store in OLSB
```

```
Replace LSB of OCB with SHB (the half-byte of cover is replaced with half-byte of secret)
Store OCB back in Cover (COVA (I, j, k) = OCB)
Get even pixel cover byte (ECB = COVA (I, j+1, k))
Get LSB of even cover byte (ELSB)
If SHB > OLSB
 Diff = SHB – OLSB
 Subtract Diff from even LSb (ELSB = ELSB – Diff)

 %Note If ELSB - Diff < 0 store 0 in ELSB

Else Diff = OLSB – SHB
Add Diff to even lsb (ELSB = ELSB + Diff)

%Note If ELSB+ Diff > 255 store 255 in ELSB

End if

Replace LSB of even byte (EB) with ELSB
Store EB back in COVA (COVA (I, J+1, k) = EB

 End For End For End For
```

Store the changed Cover array into stegano file, which has the same name as cover file, added to it the prefix "enc".

4.3.2 Extract-Change Algorithm

Extract-change algorithm Embedding Algorithm

Purpose: To extract hidden data from odd pixels of the stego image and save the recovered data to Recovered-Secret file.

Input: Stego file, in BMP, PNG or TIFF formats.

Output: Recovered-Secret file, in the format of the original secret file.

Data Structure

Stego [Width, Height, and Color]: to store the stego 3-dimensional image array

Stego-Vector [Half-Bytes-Count]: to store bytes of stego file as a vector

R-Bytes [Bytes-Count] : To store recovered bytes

RLSB: 8-bit recovered LSB half byte

RMSB: 8-bit recovered MSB half byte

Bytes-Count: Double, size of secret file in bytes

Half_Bytes-Count: Double, Bytes-Count x 2

Processing Steps:

```
Open Stego file
Read Stego files into Stego array
Get Secret file name and Secret file size in bytes (Bytes-Count)
Open Recovered-Secret file for output
Half-Bytes-Count = Bytes-Count * 2
R-Index = 0
```

```
        Vectorize Stego array into Stego-Bytes array, skipping even pixels
        For i = 1 to Half-Bytes-Count, Step 2
        Get LSB of Stego-Bytes (i) and store in RLSB
        Get LSB of Stego-Bytes (i+1) and store in RMSB
        R-Index = R-Index+1
        Concatenate RLSB and RMSB into R-Byte (R-Index)
End For
Save R-Bytes vector array into Recovered -Secret file
Close files
End Algorithm
```

4.4. Principle of operation of the proposed Embed-All algorithm

The purpose of this algorithm is to hide in odd and even pixels, by replacing LSB half _bytes of cover image pixel with half-bytes the secret image.

Several operations to be implemented, as in the flow diagram below:

1- Get next secret half-byte (SHB) of a byte from the secret image.

2- Get the lower half-byte (LSB) of the next color channel of the current pixel (PI).

3- Replace LSB with SHB.

4- After changing the lower half-bytes of the three channels of the current pixel, select the next pixel, which is PI+1 (storage in odd and even pixels)

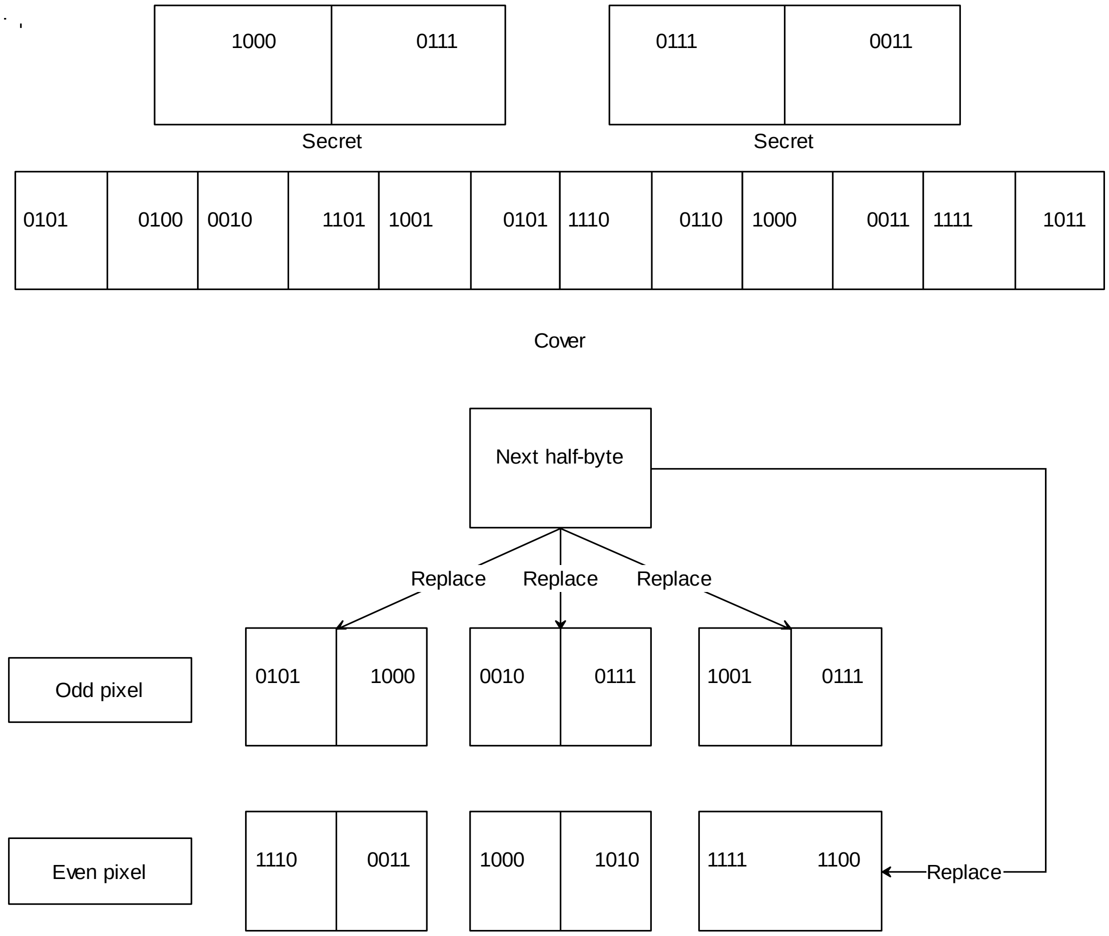


Figure (5-2) an Example Illustrating the Embedding All Process

4.4.1 Embed-All Algorithm

Embed-All Algorithm

Input: Secret Message, Cover-image.

Output: Stego-image.

Processing: to embed the secret image in half-bytes of RGB channels, in odd and even pixels. Hiding Capacity is 50% of the available bytes of the selected pixels, which is equivalent to 12 bits-per-pixel (bpp).

Algorithm steps:

```
    Open secret image file (jpg, bmp, tiff, mp3, mp4, pdf format) in binary format
    Store secret image file in secret array
    Convert secret image to half-bytes in half-byte array (HBA)
    HBCount = size of secret half-bytes array
    Open cover file (BMP or TIF format)
    Read cover image into cover array (COVA)
    Get height and length of cover array (Height and Length)
    Note: Process Secret half-byte arrays, store secret half-bytes in odd pixels and even pixels.
    For I = 1 to Height
    For J= 1 to Width -1, Step 1
    For K= 1 to 3
    Get next secret half-byte (SHB) (if end of array break)
    Get next cover-byte (CB = COVA (I, J, K))
    Get LsB of CB and store in LSB
    Replace LSB of CB with SHB (the half-byte of cover is replaced with half-byte of secret)
    Store CB back in Cover (COVA (I, J, K) = CB)
End for End For End For
```

Store the changed cover array into stegao file, which has the same name as cover file, added to it the prefix "enc".

End of Embed-All Algorithm

4.4.2 Extract-All Algorithm

Purpose: To extract hidden data from odd and even pixels of the stego image and save the extracted data to the Recovered-Secret file.

Input: Stego file, in BMP, PNG or TIFF formats.

Output: Recovered-Secret file, in the format of the original secret file.

Data Structure

Stego [Width, Height, and Color]: to store the stego 3-dimensional image array

Stego-Vector [Half-Bytes-Count]: to store bytes of stego file as a vector

R-Bytes [Bytes-Count] : To store recovered bytes

RLSB: 8-bit recovered LSB half byte

RMSB: 8-bit recovered MSB half byte

Bytes-Count: Double, size of secret file in bytes

Half-Bytes-Count: Double, Bytes-Count x 2

Processing Steps:

```
    Open Stego file
    Read Stego file into Stego array
    Get Secret file name, and Secret file size in bytes (Bytes_Count)
    Open Recovered_Secret file for output
    Half-Bytes-Count = Bytes-Count * 2
    R-Index = 0
    Vectorize Stego array into Stego-Bytes array
    For i = 1 to Half-Bytes-Count, Step 2
```

```
        Get LSB of Stego-Bytes (I) and store in RLSB
        Get LSB of Stego-Bytes (i+1) and store in RMSB
        R-Index = R_Index+1
        Concatenate RLSB and RMSB into R-Byte (R-Index)
End For
Save R-Bytes vector array into Recovered-Secret file
Close files
End Algorithm
```

Chapter Five

Experimental Work and

Discussion of Results

Chapter Five

Experimental Work and Discussion of Results

5.1. Overview

This chapter includes discussion of results of using the proposed algorithms for hiding and extracting data, as well as comparison with previous work. In the implementation of the algorithm, the matlab2012b package was used. The proposed algorithms were applied in hiding several JPG images, of varying sizes and content. Additional media embedding was carried out (audio and video) for testing the integrity of the recovery (extraction) process, as well as multi level nested embedding.

5.2. Image Embedding / Extraction Results

Figure(5-1) Lena + Hubble vs. Sego Image (Embed-Odd)

Figures (5-1) shows the cover-image lena.bmp(768KB), and the stego-image after embedding the secret image Hubble.jpg (79.2KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels. The resulting PSNR = 41.9726.

Figure(5-2)Lena + Peacock.jpg vs. Sego Image(Embed- Odd)

Figures (5-2) shows the cover-image lena.bmp(768KB),and the Stego-image after embedding the secret image Peacock.jpg(94.4KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels The resulting PSNR= 40.9588

Figure(5-3)Lena.bmp + Starry-Night vs. Sego Image(Embed- Odd)

Figures (5-3) shows the cover-image lena.bmp(768KB),and the Stego-image after embedding the secret image Starry-Night.jpg(105KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels The resulting PSNR = 40.7453.

Figure(5-4)Lena + Blubird vs. Sego Image(Embed- Odd)

Figures (5-4) shows the cover-image lena.bmp(768KB),and the Stego-image after embedding the secret image Blubird.jpg(133KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels The resulting PSNR= 39.6795.

Figure(5-5)Lena + Fox1 vs. Sego Image(Embed- Odd)

Figures (5-5) shows the cover-image lena.bmp(768KB),and the Stego-image after embedding the secret image Fox1.jpg(190KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels.The resulting PSNR= 38.8275.

Figure(5-6) Lena.bmp + Hubble vs. Sego Image (Embed- Odd)

Figures (5-6) shows the cover-image lena.bmp(768KB),and the Stego-image after embedding the secret image Hubble.jpg(79.2KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels. The resulting PSNR= 43.4968.

Figure(5-7)Lena + Peacock vs. Sego Image(Embed-All)

Figures (5-7) shows the cover-image lena.bmp (768KB), and the stego-image after embedding the secret image Peacock.jpg (99.4KB), which replaces the LSB half-bytes of each color channel in all pixels (odd and even) with a half-byte from the secret image. The resulting PSNR= 42.4310.

Figure(5-8)Lena + Starry-Night vs. Sego Image(Embed-All)

Figures (5-8) shows the cover-image lena.bmp(768KB),and the Stego-image after embedding the secret image Starry-Night.jpg(105KB), which replaces the LSB half-bytes of each color channel in all pixels (odd and even) with a half-byte from the secret image. The resulting PSNR= 42.1982.

Figure(5-9)Lena + Blubird vs. Sego Image(Embed-All)

Figures (5-9) shows the cover-image lena.bmp(768KB),and the Stego-image after embedding the secret image Blubird.jpg(133KB), which replaces the LSB half-bytes of each color channel in all pixels (odd and even) with a half-byte from the secret image. The resulting PSNR= 41.2134.

Figure(5-10)Lena + Fox1 vs. Sego Image(Embed-All)

Figures (5-10) shows the cover-image lena.bmp(768KB),and the Stego-image after embedding the secret image Fox1.jpg(190KB), which replaces the LSB half-bytes of each color channel in all pixels (odd and even) with a half-byte from the secret image. The resulting PSNR= 39.8674.

Figure (5-11)Tulips + Hubble vs. Sego Image(Embed- Odd)

Figures (5-11) shows the cover-image tulips.bmp (1.12MB), and the Stego-image after embedding the secret image Hubble.jpg(79.2KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels. The resulting PSNR= 43.6550.

Figure(5-12)Tulips + Peacock vs. Sego Image(Embed- Odd)

Figures (5-12) shows the cover-image tulips.bmp(1.12MB),and the Stego-image after embedding the secret image Peacock.jpg(99.4KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels. The resulting PSNR= 42.6494.

Figure(5-13)Tulips + Starry-Night vs. Sego Image(Embed- Odd)

Figures (5-13) shows the cover-image tulips.bmp(1.12MB),and the Stego-image after embedding the secret image Starry-Night.jpg(105KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels. The resulting PSNR= 42.4234.

Figure(5-14)Tulips + Blubird VS. Stego Image(Embed- Odd)

Figures (5-14) shows the cover-image tulips.bmp(1.12MB),and the Stego-image after embedding the secret image Blubird.jpg(133KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels. The resulting PSNR= 41.3645.

Figure(5-15)Tulips + Fox1 vs. Stego Image(Embed- Odd)

Figures (5-15) shows the cover-image tulips.bmp(1.12MB),and the Stego-image after embedding the secret image Fox1.jpg(190KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels.The resulting PSNR= 40.7941.

Figure(5-16)Tulips + Fox2 vs. Stego Image(Embed-Odd)

Figures (5-16) shows the cover-image tulips.bmp(1.12MB),and the Stego-image after embedding the secret image Fox2.jpg(230KB), using the Embed-Odd alghortihm which replaces the LSB half-bytes of each color channel with a half-byte from the secret image and adjusts even pixels.The resulting PSNR= 39.0299.

Figure(5-17)Tulps + Hubble vs. Stego Image(Embed-All)

Figures (5-17) shows the cover-image tulips.bmp(1.12MB),and the Stego-image after embedding the secret image Hubble.jpg(79.2KB), which replaces the LSB half-bytes of each color channel in all pixels (odd and even) with a half-byte from the secret image. The resulting PSNR= 45.4136.

Figure(5-18)Tulips.bmp with Peacock.jpg VS. Sego Image(Embed-All)

Figures (5-18) shows the cover-image tulips.bmp(1.12MB),and the Stego-image after embedding the secret image Peacock.jpg(99.4KB), which replaces the LSB half-bytes of each color channel in all pixels (odd and even) with a half-byte from the secret image. The resulting PSNR= 44.3771.

Figure(5-19)Tulips + Starry-Night vs. Stego Image(Embed-All)

Figures (5-19) shows the cover-image tulips.bmp(1.12MB),and the Stego-image after embedding the secret image Starry-Night.jpg(105KB), which replaces the LSB half-bytes of each color channel in all pixels (odd and even) with a half-byte from the secret image. The resulting PSNR= 44.1766

Figure(5-20)Lena + Blubird vs. Stego Image(Embed-All)

Figures (5-20) shows the cover-image tulips.bmp(1.12MB),and the Stego-image after embedding the secret image Blubird.jpg(133KB), which replaces the LSB half-bytes of each color channel in all pixels (odd and even) with a half-byte from the secret image. The resulting PSNR= 43.1474.

Figure(5-21)Lena + Fox1 vs. Stego Image(Embed-All)

Figures (5-21) shows the cover-image tulips.bmp(1.12MB),and the Stego-image after embedding the secret image Fox1.jpg(190KB),changing lower half-octet of each color channel,with change to even pixels(Successive RGB)

Figure(5-22)Tulips + Fox2 vs. Stego Image(Embed-All)

Figures (5-22) shows the cover-image tulips.bmp(1.12MB),and the Stego-image after embedding the secret image Fox2.jpg(390KB), which replaces the LSB half-bytes of each color channel in all pixels (odd and even) with a half-byte from the secret image. The resulting PSNR= 40.7476.

 Starry-night	 **Peacock**	 **Hubble**
 Fox2	 **Fox1**	 **Blue bird**

Figure (5-23) Secret massage

5.3. Results in Terms of PSNR and Hiding Capacity

Secret Image	Size KB	Embed Odd			Embed All		
		PSNR	MSE	HC	PSNR	MSE	HC
Hubble	79.2KB	41.9726	0.0446982	196608	43.4968	0.0341964	393216
Peacock	99.4KB	40.9588	0.0593702	196608	42.4310	0.0437076	393216
Starry – Night	105KB	40.7453	0.0593702	196608	42.1982	0.0461148	393216
Bluebird	133KB	39.6795	0.0761121	196608	41.2134	0.057472	393216
Fox1	181KB	38.8275	0.10021	196608	39.8674	0.0788726	393216

Table (5-1) Image Quality of lena in Terms of PSNR and MSE and HC.

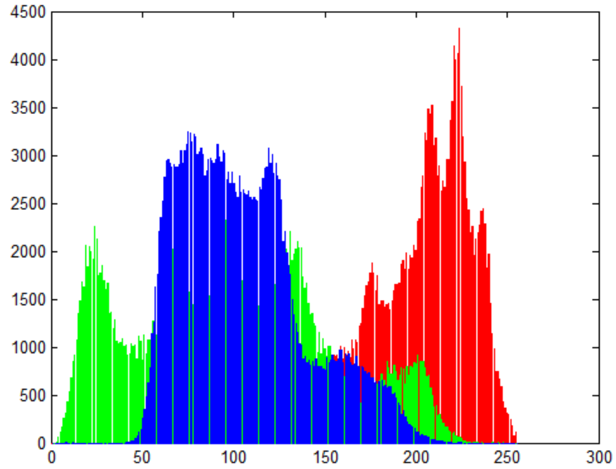
Secret Image	Secret Image Size KB	Embed-Odd			Embed-All		
		PSNR	MSE	HC	PSNR	MSE	HC
Hubble	79.2KB	43.6550	0.029813	294912	45.4136	0.0219939	589824
Peacock	99.4KB	42.6494	0.0377167	294912	44.3771	0.0279221	589824
Starry – Night	105KB	42.4234	0.0397156	294912	44.176	0.0292419	589824
Bluebird	133KB	41.3645	0.0506796	294912	43.1474	0.0371354	589824
Fox1	181KB	40.7941	0.0658929	294912	42.4868	0.0498685	589824
Fox2	381KB	39.0299	-	-	40.7476	0.106257	589824

Table (5-2) Image Quality of Tulips in Terms of PSNR, MSE and HC

5.4. Histogram Comparison

This test shows a comparison between the original image and the stego image, using the histogram as a visual comparison tool. The degradation in image quality can be visually noticed by applying the histogram analysis. In statistics, a histogram is a graphical display of tabulated frequencies, shown as lines. It shows what proportion of cases fall into each of several categories: it is a form of data binning. Figure 5-24 shows the histogram of the orginal lena image and Figure 5-25 shows the histogram of the stego image after embbeding Hubble.jpg. Similarly the histograms of lena using other secret images are shown in Figures 5-26 to 5-28, and the histograms of Tulips.bmp as a cover are shown in Figures 5-30 to 5-33.

It is evident that histogram analysis will show the effect of steganography, assuming that the attacker will have available both original and stego image.

	
Original color image	Histogram image

Figure(5-24) Histogram of Cover Lena.bmp

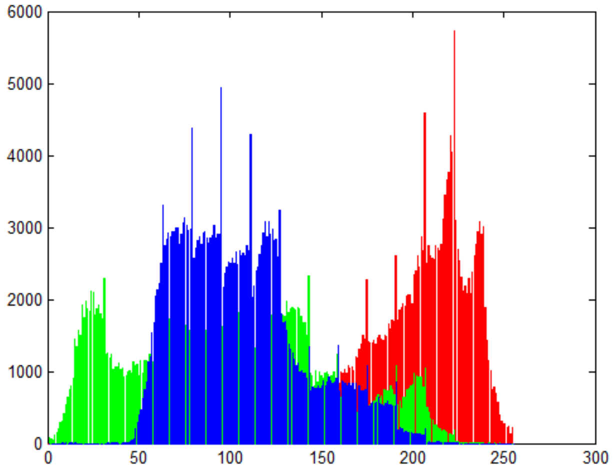
	
Stego color image	Histogram image Stego lena (Hubble) Embed-Odd

Figure (5-25) Histogram of Cover Lena with Hubble Embed-Odd

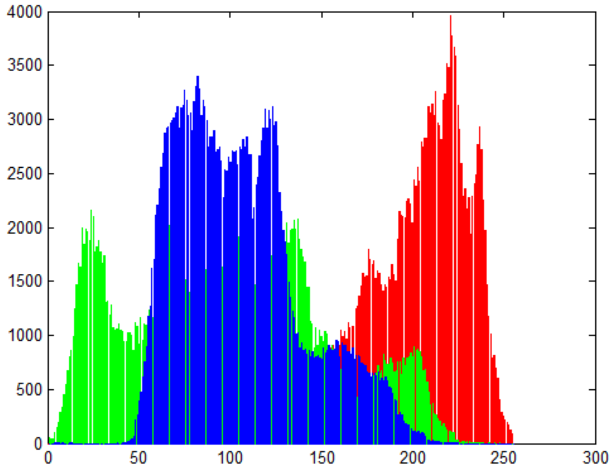
	
Stego color image	Histogram image Stego lena (Fox1) Embed-Odd

Figure (5-26) Histogram of Cover lena with Fox1 Embed-Odd.

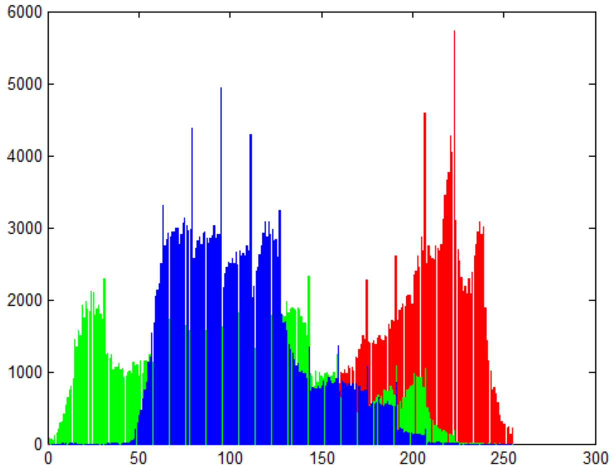
	
Stego color image	Histogram image Stego lena (Hubble) Embed-All

Figure (5-27) Histogram of Cover lena with Hubble Embed-All.

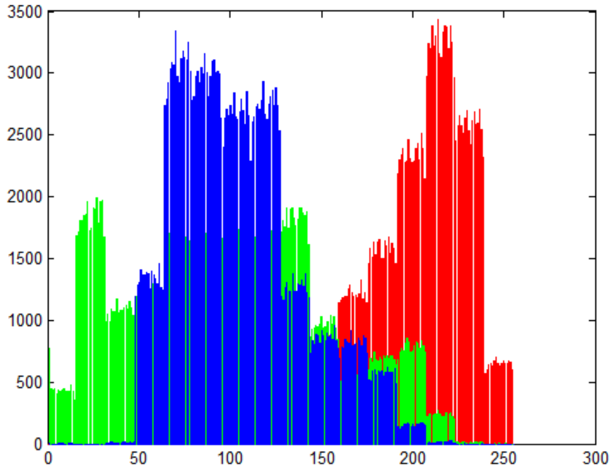
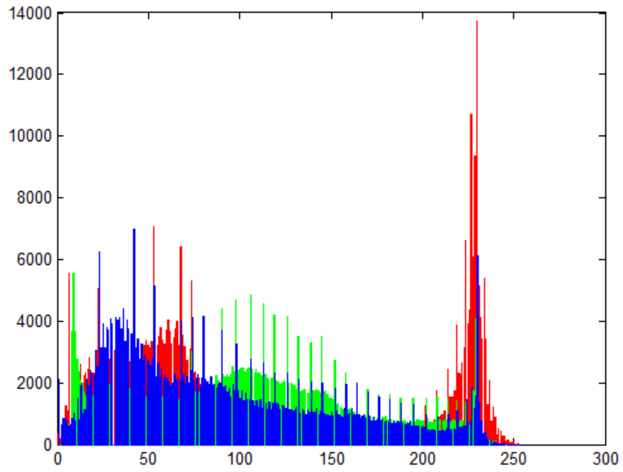
	
Stego color image	Histogram image Stego lena (Fox2) Embed-All

Figure (5-28) Histogram of Cover lena with Fox2 Embed-AlL

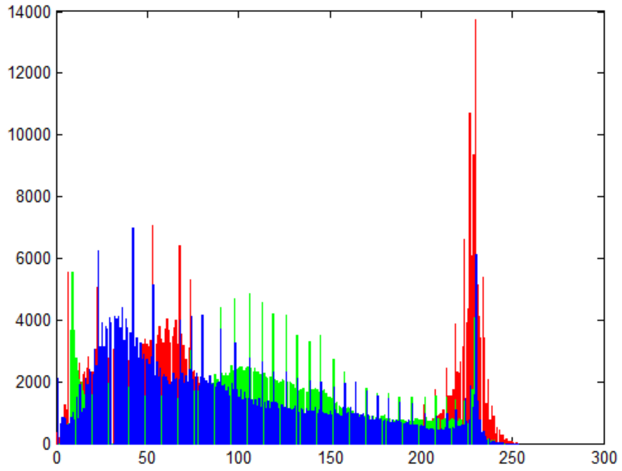
Original color image



Histogram image

Figure (5-29) Histogram of Cover Tulips.bmp

Stego color image



Histogram image Stego Tulips (Hubble) Embed-Odd

Figure (5-30) Histogram of Cover Tulips with Hubble Embed-Odd

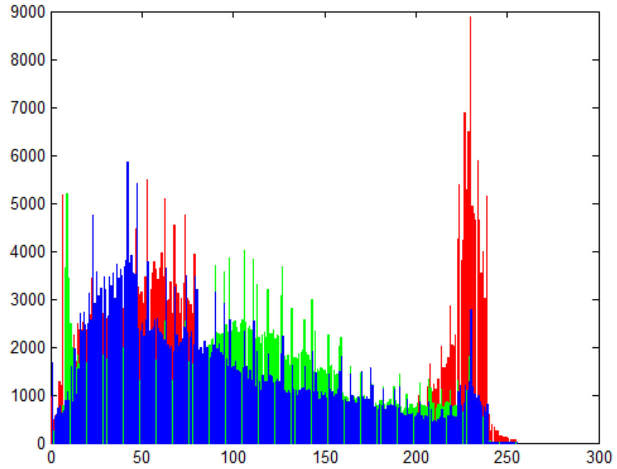
	
Stego color image	Histogram image Stego Tulips (Fox1) Embed-Odd

Figure (5-31) Histogram of Cover Tulips with Fox1 Embed-Odd

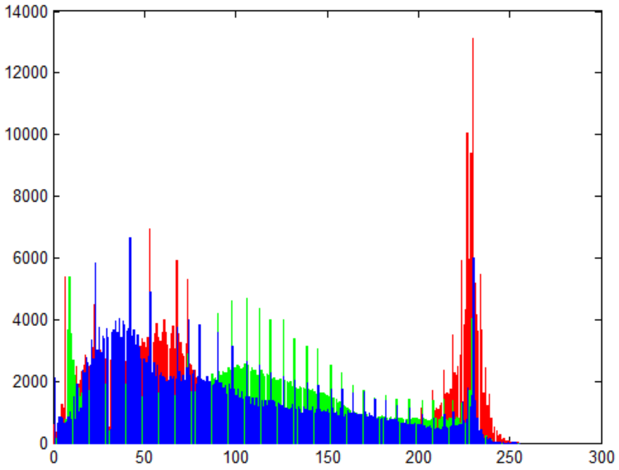
	
Stego color image	Histogram image Stego Tulips (Hubble) Embed- All

Figure (5-32) Histogram of Cover Tulips with Hubble Embed All

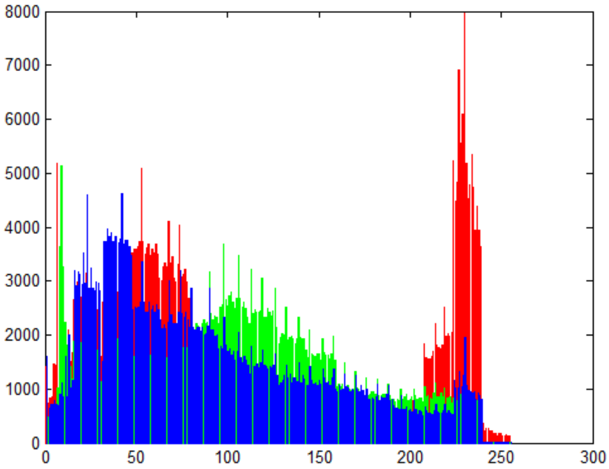
	
Stego color image	Histogram image Stego Tulips (Fox2) Embed- All

Figure (5-33) Histogram of Cover Lena with Fox2 Embed-All

After studying the above tables, figures and performing calculation based on PSNR values, and maximum hiding capacity, and after viewing the above figures we can conclude that the average hiding capacity of the proposed technique shows more satisfied experimental outcomes, retains good visual clarity of stego images, In the histogram analysis the histogram Where storage is used on all channels without affecting the image of the cover.

5.5. Image Quality Comparison with Low Hiding Capacity Work

In this section a comparison is made between the proposed algorithm and an algorithm that has a low hiding capacity. For this purpose Ghosal's algorithm is considered which uses 2 bits per pixel for hiding secret data.

The results are shown in Table 5-3 which compares image quality in terms of PSNR between the proposed algorithms and that of Ghosal, using the same set of cover images of 1024x1024 BMP images, and a small secret JPG image of size 31.2 KB. It can be seen that the PSNR values of the two algorithms and Ghosal's algorithm are very close, despite the fact the proposed algorithms use 12 and 6 bits per pixel for hiding data, while Ghosal's algorithm uses 2 bits per pixel. This suggest that we can increase size of the secret image to take advantage of the higher hiding capacity of the proposed algorithms and still maintain acceptable PSNR values.

Images	Image size	Ghosal's Algorithm	Proposed algorithm Embed-Odd	Proposed algorithm Embed-All
		PSNR		
Animal1	1024*1024	56.50	52.5465	53.8704
Animal2	1024*1024	56.68	52.4533	53.8164
Animal3	1024*1024	56.92	52.8868	54.5336
Sport1	1024*1024	53.91	52.1704	53.8315
Sport2	1024*1024	46.14	51.2715	53.6769
Cartoons1	1024*1024	48.06	50.3386	50.3385

Table (5-3) Image Quality Comparison in Terms of PSNR.

Figures (5-34), (5-35), (5-36), (5-37), (5-38), and (5-39) compares the PSNR of the proposed algorithms with Ghosal's, using a histogram, which shows the similarity of the PSNR values.

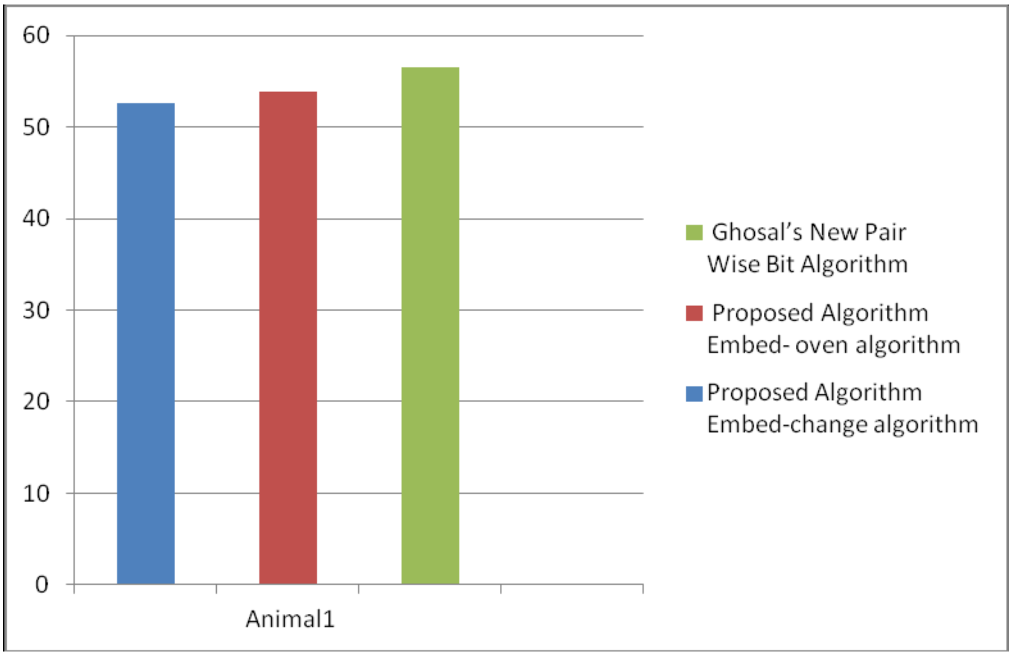


Figure (5-34) shows the PSNR test for Animal1 image

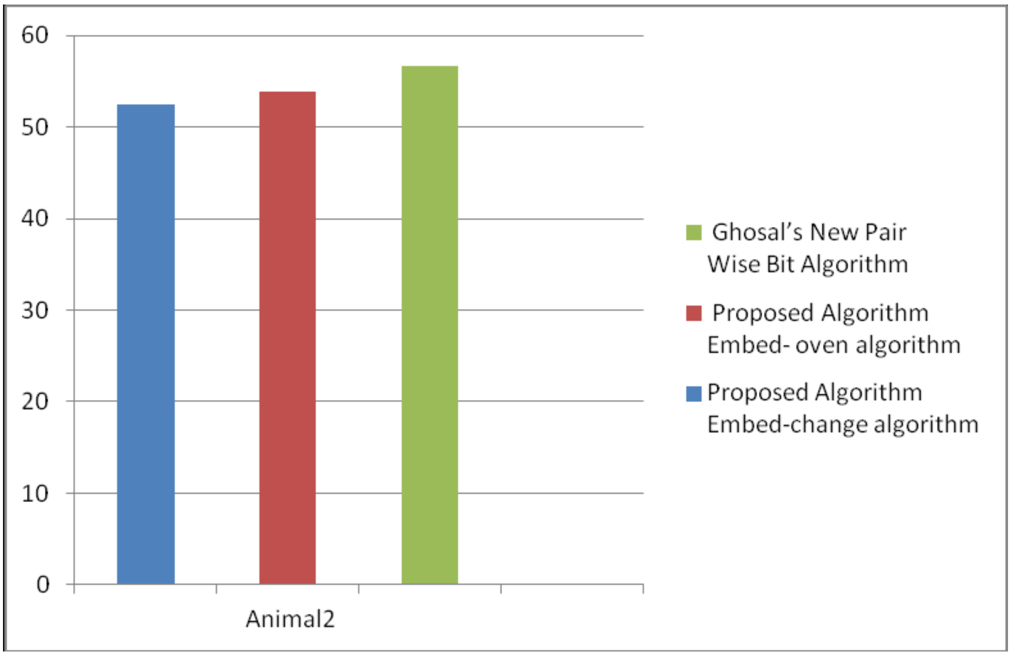


Figure (5-35) shows the PSNR test for Animal2 image

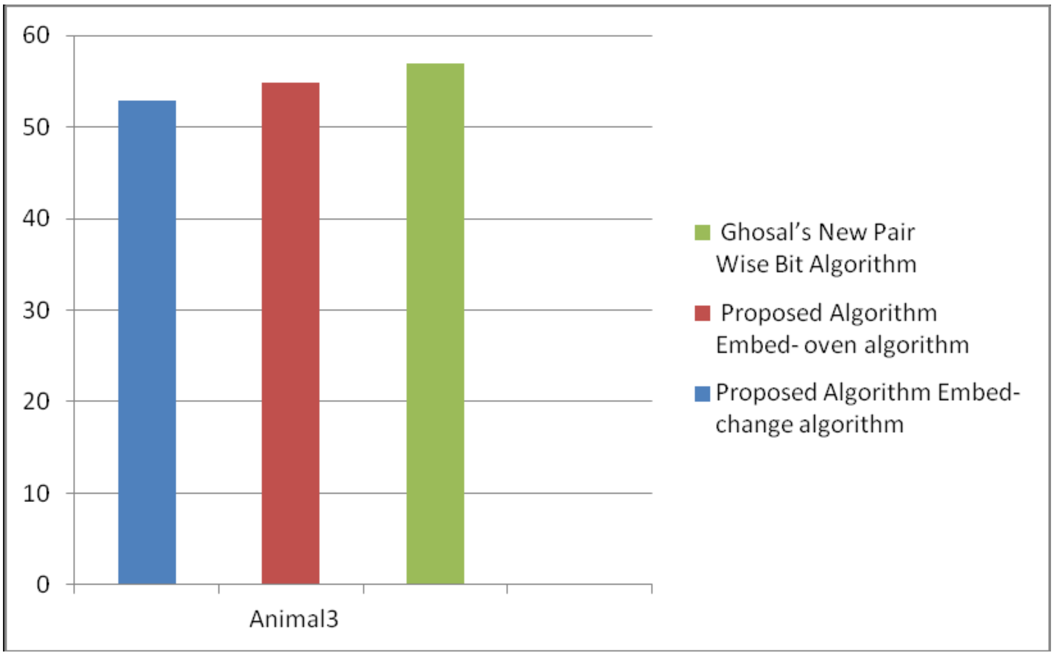


Figure (5-36) shows the PSNR test for Animal3 image

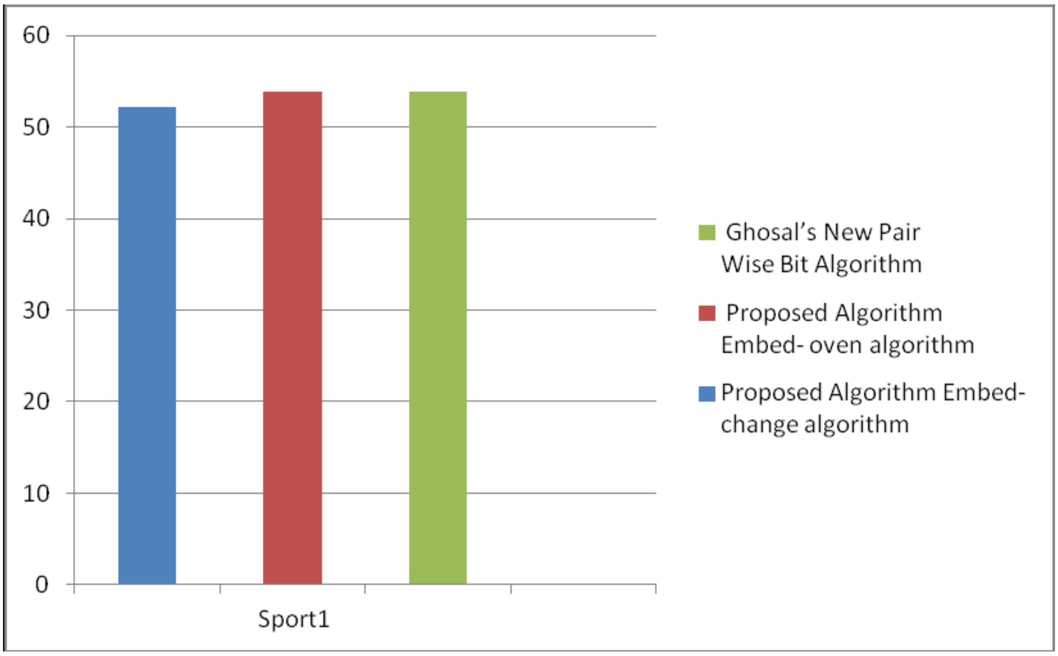


Figure (5-37) shows the PSNR test for Sport1 image

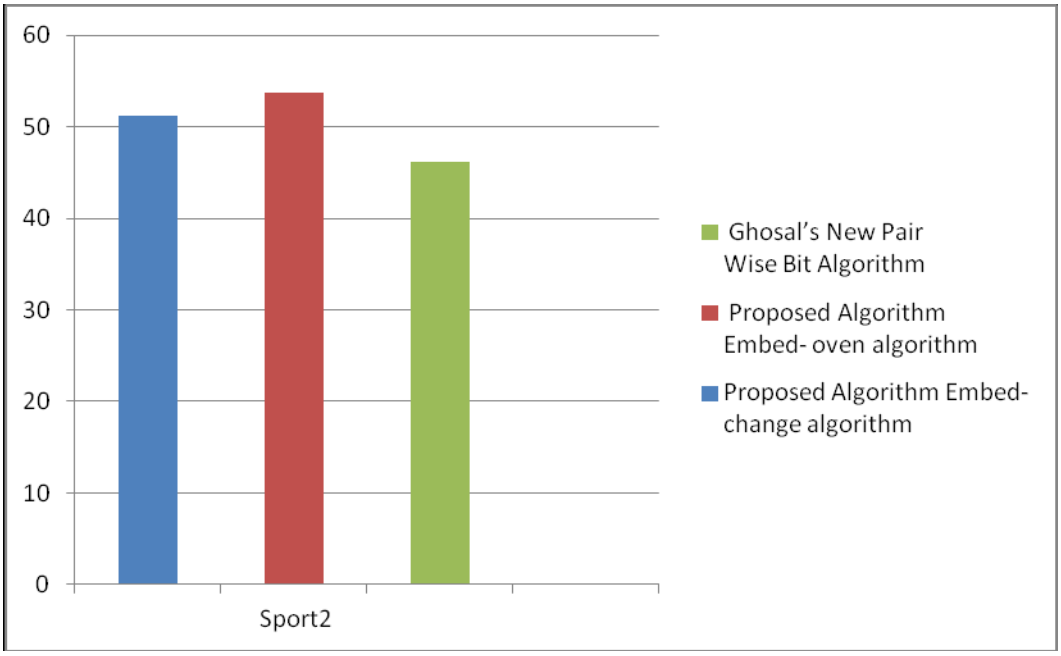


Figure (5-38) shows the PSNR test for Sport2 image

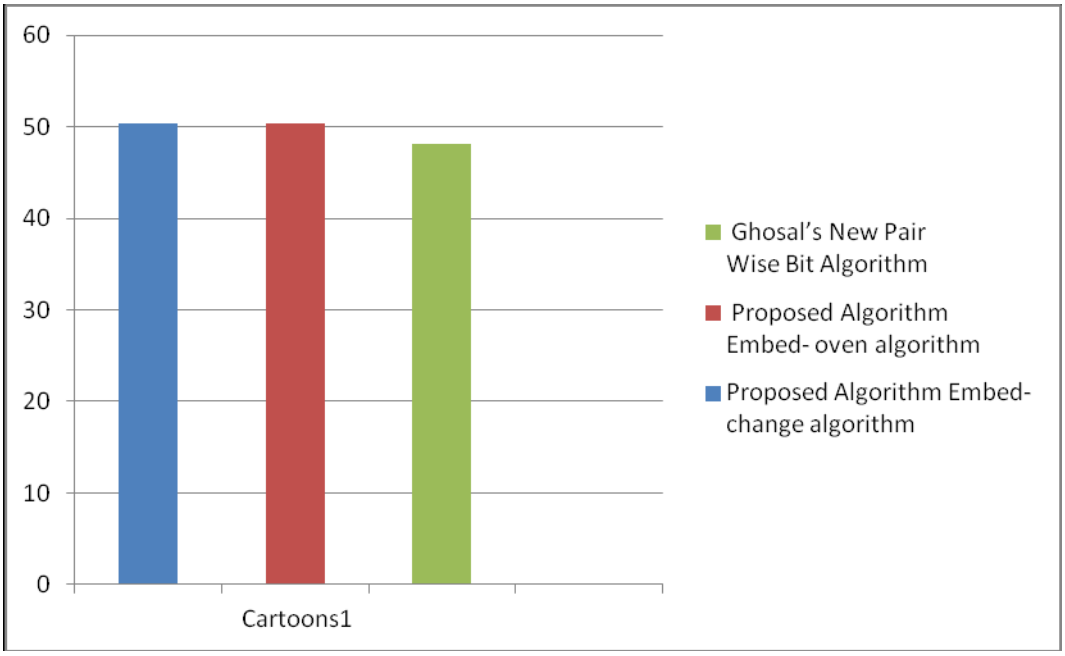


Figure (5-39) shows the PSNR test for Cartoons1 image

5.6 Comparison with a High Hiding Capacity Algorithm

In the work of Koppola (2009), the hiding capacity payload is 100%, which means that it is possible to hide a secret image that is equal in size to the cover image. This is achieved through lossy compression combined with LSB embedding of 13 bits per pixel in an RGBA image. In our work the compression is considered a step to be performed before the stegoanography process, using the JPG compression which results in very high compression (Chang et al., 2002).

To compare the two approaches, we have taken two BMP images, lena and flower, from the test images used by Koppola. The BMP images were first compressed by converting them to JPG. Then the JPG image is processed as a secret image and embedded in the original un-compressed BMP image.

Table (5-4) shows the PSNR comparison results of Koppola’s algorithm with the combined JPG compression and Embed-All algorithm, using lena and flower as both secret and cover images.

Cover Images	Secret Image	Image resolution	Image size (Bytes)	Koppola PSNR	Proposed Embed-All PSNR
Lena.bmp	Lena.bmp	512x512	786,432	34.5	45.3
Flower.bmp	Flower.bmp	512x480	737,280	36.0	46.3

Table (5-4) PSNR Comparison between Embed-All and Koppola Algorithms

The higher PSNR obtained in our work compared to Koppola is due to the fact that we have used the JPG compression which gives much higher compression than Koppola's algorithm, and therefore the resulting compressed image was much smaller, and when hidden in the un-compressed image the distortion is much less than in the case of Koppola's work.

In addition, the hiding capacity payload of the combined JPG and Embed-All algorithm is much higher than the 100% achieved in the Koppola work, due to the much higher compression ratio of JPG compared to the compression method of Koppola’s algorithm. Figure 5-40 shows a visual comparison between the cover image flower.bmp and the stego image which carries a compressed copy of the cover image, and similarly Figure 5-41 shows lena.bmp as a cover against the stego image which carries a compressed image of the cover.

Figure(5-40) flower.bmp cover image vs. stego image compressed flower.jpg

Figure(5-41) lena.bmp cover image vs. stego image compressed lena.jpg

5.7. Payload (Hiding Capacity)

The table (5-5) shows the data payload which can be embedded inside different loads of the images by using the proposed algorithms and Ghosal's new pair-wise bit algorithm with its estimated load. This table shows that the data payload using the proposed algorithms is higher than Gosal's algorithm due to the reason that the proposed Embed-Odd algorithm mbeds 6 bits per pixel, and the Embed-All algorithm embeds 12 bits in each pixel, while Gosal's algorithm embeds 2 bits in each pixel.

Image	Animal1
Image Size	1024*1024
Ghosal's Algorithm (Hidden Capacity Bytes)	262,144
Proposed algorithm Embed-Odd Algorithm (Hidden Capacity Bytes)	786,432
Proposed algorithm Embed-All Algorithm (Hidden Capacity Bytes)	1,572,864

Table (5-5) Comparison of Hiding capacity Using a 1024x1024 24-Bit BMP Image

It is evident from the results in table that the hiding capacities of the proposed algorithms are six times that of Ghosal for the Embed-All algorithm and 3 times for Embed-Odd algorithm.

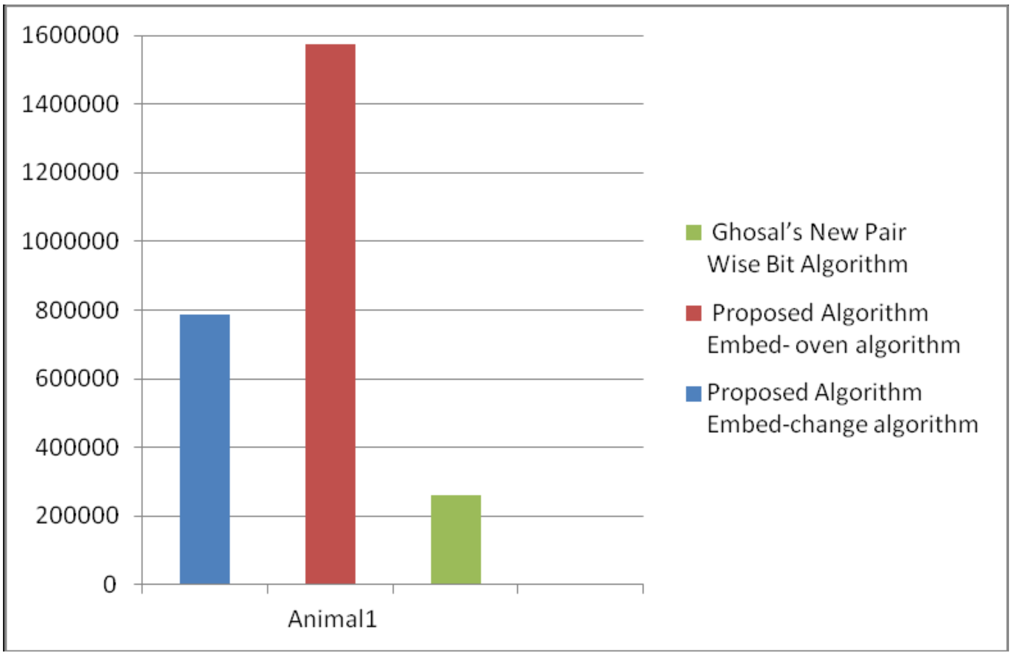


Figure (6-42) shows the payload inside Animal1 image

Figure (5-43-A, B) Cover / Stego images comparison, after hiding 31.2 KB inside (Animal 1) using proposed Embed-Odd

Figure(5-43-A)Cover-image for (Animal 1) with size 1024X1024	Figure(5-43-B)Stego-image for (Animal 1) with size 1024X1024

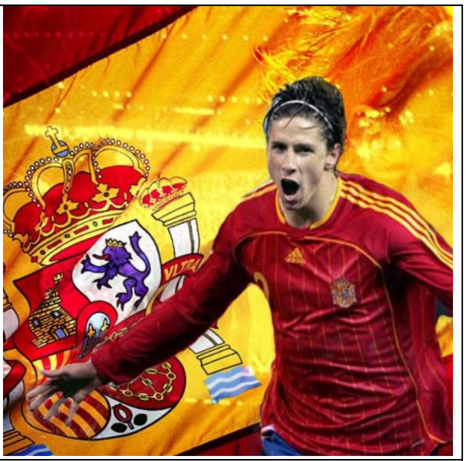
Figures (5-44) shows Cover-image, Stego-image after 31.5KB inside (Animal 2). Picture by proposed Embed- Odd

Figure(5-44-A)Cover-image for (Animal 2) with size 1024X1024	Figure(5-44-B) Stego-image for (Animal 2) with size 1024X1024

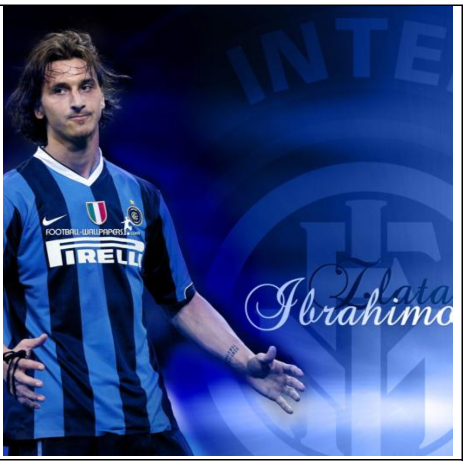
Figures (5-45) shows Cover-image, Stego-image after 31.5KB inside (Animal 3). Picture by proposed Embed-Odd

Figure(5-45-A)Cover-image for (Animal 3) with size 1024X1024	Figure(5-45-B)Stego-image for (Animal 3) with size 1024X1024

Figures (5-46) shows Cover-image, Stego-image after 31.5KB inside (Sport1). Picture by proposed Embed- Odd

	
Figure(5-46-A)Cover-image for (Sport 1) with size 1024X1024	Figure(5-46-B)Stego-image for (Sport1 1) with size 1024X1024

Figures (5-47) shows Cover-image, Stego-image after 31.5KB inside (Sport 2). Picture by proposed Embed-All

	
Figure(5-47-A) Cover-image for (Sport 2) with size 1024X1024	Figure(5-47-B)Stego-image for (Sport 2) with size 1024X1024

Figures (5-48) shows Cover-image, Stego-image after 31.5KB inside (Cartoons1). Picture by proposed Embed-All

Figure(5-48-A) Cover-image for (Cartoons1) with size 1024X1024	Figure(5-48-B)Stego-image for (Cartoons 1) with size 1024X1024

5.8 Additional Testing of the Proposed Algorithms

The purpose of these additional tests is to test the algorithms in the maximum hiding capacity .Also, to test nested embedding for further confirmation that a recovered secret message is not altered in the process. Audio and video secret messages are to show the intergity of recovered files, and that no damage or alteration has been done to those files.

5.8.1 Test No. 1: Embedding Large PDF Images File

The Purpose To demonstrate the embedding and extraction of a large PDF image file, using the successive algorithm.

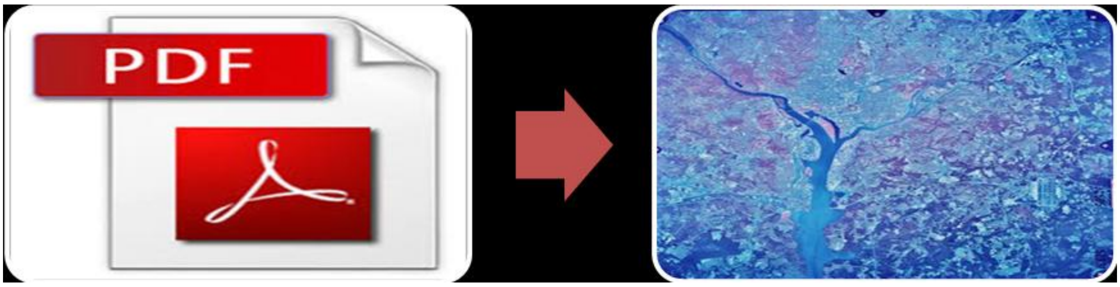
Secret Document :

اطلس التاريخ العربي الاسلامي (د. شوقي أبوخليل، ط5 ، دار الفكر ، دمشق ، سوريا، 2005)

Secret File:6.9 MB, PDF file, 60 image pages (of 337 pages)

Cover Image:

Infra-Red Aerial View of Washington DC (SIPI image database, University of Southern California)BMP Image, 14 MB, PSNR= 36.6016.



Figure(5-49)Embedding A Pdf File

5.8.2 Test No. 2: Embedding a Video File

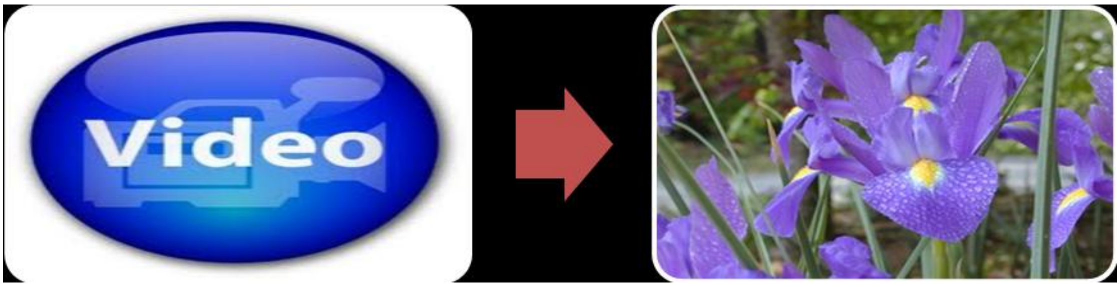
The Purpose demonstrate the integrity of embedding and recovery of a video file.

- Secret Video:

  25dividedby5 (source from the internet) asx format, 4.05 MB, 2:15 minutes.

- Cover Image:

  Flower (from Gonzales image database) BMP image, 9.4 MB, PSNR= 37.116



Figure(5-50)Embedding A Video File

5.8.3 Test No. 3: Nested Embedding

The Purpose To demonstrate integrity of recovered images by applying nested (Onion-Layers) embedding.

- Secret Cover:

Fur Elisa (Beethoven, no. 25 solo piano in A minor, 1810) MP3 file, 1.5 MB, Time 2:30 minutes.

- Cover Layer-1:

Therese (dedicatee of Fur Elisa, by unknown artist)BMP 3 MB, PSNR = 36.6463.

- Cover Layer-2:

Beethoven (by Joseph Karl Stieler, 1820) BMP 6 MB, PSNR = 36.5714.

- Cover Layer-3:

Monalisa (Leonardo Da Vinci, 1503-1506) BMP image, 12 MB, PSNR= 36.2130.

Figure(5-51)Embedding Nested Fil

Chapter Six

Conclusion and Future

Work

Chapter Six

6.1. Conclusions

Based on the experimental work carried out using the proposed Embed-Odd and Embed-All algorithms, several conclusions can be made.

Replacing four bit LSB (half-byte) of each color channel of every pixel, using the Embed-All algorithm, is visually un-noticeable, as shown in the presented standard image comparison between cover and stego images.

In addition, the resulting hiding capacity (payload) of the Embed-All algorithm is 50% of the available storage space in pixels, as the number of embedded bits per pixel is 12 (out of 24 bits in a pixel). This gives a high storage capacity to store digital images and multi-media file, especially when such files are compressed before embedding. The Embed-Odd algorithm has a hiding capacity of 25%, as it embeds 6 bits per pixel.

The obtained results of image quality comparison between cover and stego images using the similarity comparison metric of PSNR has shown acceptable results, well over the threshold of 30 dB which is considered as the minimum value for un-distorted images.

For the Embed-Odd algorithm, the PSNR values were close to the Embed-All algorithm, however although the Embed-Odd has lower payload, but it has the advantage of adding noise to even pixels, to confuse an attacker, as well as reducing image distortion through color adjustment in even pixels.

The work has demonstrated the practical possibility of reading compressed images and multi-media files without the need for un-compressing them, thereby allowing large image embedding through compression first, then embedding.

6.2. Future work

The following ideas can be suggested as an extension of the work in this thesis:

1- Investigating the use of 16-bit color channel (48 bit RGB) for hiding larger data.

2- Investigating the use of the successive pixels method (Embed-All algorithm) with alternating number of bits per color, for example:

  - In odd pixels: 4 bit red, 2 bit green, 4 bit blue.
  - In even pixels: 2 bit red, 4 bit green, 2 bit blue.

3- Investigating the hiding of RGB images inside other media files such as audio, video.

4- Implementing the proposed algorithms as smart phone APP for protecting documents on mobile phone.

References

Ahmed Laskar, S., & Hemachandran, K. (2012). High Capacity data hiding using LSB Steganography and Encryption. *International Journal of Database Management Systems*, 4(6), (pp.57-68).

Al-Husainy, M. A. F. (2012). Message Segmentation to Enhance the Security of LSB Image Steganography. Transit, 3(3).

Almohammad, A., & Ghinea, G. (2010, June). Image steganography and chrominance components. In Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on (pp. 996-1001).

Bashardoost, M., Sulong, G. B., & Gerami, P. (2013). Enhanced LSB Image Steganography Method By Using Knight Tour Algorithm, Vigenere Encryption and LZW Compression, ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784

Bender, W., Gruhl, D., & Morimoto, N. (1999). U.S. Patent No. 5,893,067. Washington, DC: U.S. Patent and Trademark Office

Chang, C. C., Chen, T. S., & Chung, L. Z. (2002). A steganographic method based upon JPEG and quantization table modification. *Information Sciences*, *141*(1), 123-138.

Chang, C. C., & Tseng, H. W. (2004). A steganographic method for digital images using side match. Pattern Recognition Letters, 25(12), 1431-1437.

Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. Signal Processing, 90(3), 727-752.

Cole, E., & Krutz, R. D. (2003). Hiding in plain sight: Steganography and the art of covert communication. John Wiley & Sons, Inc...

Cole, E., & Chairperson-Grossman, F. (2004). Stego-marking packets to control information leakage on TCP/IP based networks.

Fabien, A., Ross,J. & Markus, G. (1999) Information Hiding- A Survey, Proceeding of the IEEE, special issue on protection of multimedia content, 87 (7), 1062-1078.

Ghosal, S. K. (2011) A New Pair Wise Bit Based Data Hiding Approach on 24 Bit Color Image using Steganographic.

Gui, X., Li, X., & Yang, B. (2012, September). Improved payload location for LSB matching steganography. In Image Processing (ICIP), 2012 19th IEEE International Conference on (pp. 1125-1128).

Gutub, A. A. A. (2010). Pixel indicator technique for RGB image steganography. Journal of Emerging Technologies in Web Intelligence, 2(1), 56-64.

Gui, X., Li, X., & Yang, B. (2012, September). Improved payload location for LSB matching steganography. In Image Processing (ICIP), 2012 19th IEEE International Conference on (pp. 1125-1128).

Hemalatha, S., Acharya, U. D., Renuka, A., & Kamath, P. R. (2012). A secure and high capacity image steganography technique. *Signal & Image Processing: An International Journal (SIPIJ) Vol, 4,* 83-89.

Jamil, T. (1999). Steganography: the art of hiding information in plain sight.Potentials, IEEE, 18(1), 10-12.

Jain, V. (2012). Public-Key Steganography Based On Modified LSB Method.Journal of Global Research in Computer Science, 3(4), 26-29.

Jain, Y. K., & Ahirwal, R. R. (2010). A novel image steganography method with adaptive number of least significant bits modification based on private stego keys. International Journal of Computer Science and Security, 4(1), 40-49.

Khandekar, S. A., & Dixit, M. M. (2012). Steganography for Text Messages Using Image. J. Elect. Commun. Eng, 2, 01-04.

Luo, W., Huang, F., & Huang, J. (2010). Edge adaptive image steganography based on LSB matching revisited. Information Forensics and Security, IEEE Transactions on, 5(2), 201-214.

Marwaha, P. (2010, July). Visual cryptographic steganography in images. InComputing Communication and Networking Technologies (ICCCNT), 2010 International Conference on (pp. 1-6).

Morkel, T., Eloff, J. H., & Olivier, M. S. (2005, June). An overview of image steganography. In Information Systems Security Association (ISSA), (pp. 1-11).

Nag, A., Biswas, S., Sarkar, D., & Sarkar, P. P. (2011). A novel technique for image steganography based on DWT and Huffman encoding. *International Journal of Computer Science and Security, (IJCSS)*, *4*(6), 497-610.

Narayana, S., & Prasad, G. (2010). Two new approaches for secured image Steganography using cryptographic Techniques and type conversions. Signal & Image Processing: An International Journal (SIPIJ) Vol, 1.

Jamil, T. (1999). Steganography: the art of hiding information in plain sight.Potentials, IEEE, 18(1), 10-12.

Lashkari, A. H., Manaf, A. A., Masrom, M., & Daud, S. M. (2011). A Survey on Image Steganography Algorithms and Evaluation. In Digital Information Processing and Communications (pp. 406-418). Springer Berlin Heidelberg.

Li, B., He, J., Huang, J., & Shi, Y. Q. (2011). A survey on image steganography and steganalysis. Journal of Information Hiding and Multimedia Signal Processing, 2(2), 142-172.

Luo, W., Huang, F., & Huang, J. (2010). Edge adaptive image steganography based on LSB matching revisited. Information Forensics and Security, IEEE Transactions on, 5(2), 201-214.

Parvez, M. T., & Gutub, A. A. (2008, December). RGB intensity based variable-bits image steganography. In Asia-Pacific Services Computing Conference, 2008. APSCC'08. IEEE (pp. 1322-1327).

Petitcolas, F. A., Anderson, R. J., & Kuhn, M. G. (1998, January). Attacks on copyright marking systems. In Information Hiding (pp. 218-238). Springer Berlin Heidelberg.

Pevny, T., Bas, P., & Fridrich, J. (2010). Steganalysis by subtractive pixel adjacency matrix. Information Forensics and Security, IEEE Transactions on, 5(2), 215-224.

Poornima, R., & Iswarya, R. J.(2013). An overview of digital image steganography. International Journal of Computer Science & Engineering, (IJCSES) Vol.4.

Reddy, V. L., Subramanyam, A., & Reddy, P. C. (2011). Implementation of LSB Steganography and its Evaluation for Various File Formats. Int J Advanced Networking and Applications, 2(05), 868-872.

Sallee, P. (2005). Model-based methods for steganography and steganalysis.International Journal of Image and graphics, 5(01), 167-189

Sarkar, A., Madhow, U., & Manjunath, B. S. (2010). Matrix embedding with pseudorandom coefficient selection and error correction for robust and secure steganography. Information Forensics and Security, IEEE Transactions on,5(2), 225-239.

Schaathun, H. G. (2012). Machine learning in image steganalysis. Wiley-IEEE Press.

Selvi, G. K., Mariadhasan, L., & Shunmuganathan, K. L. (2012, March). Steganography using edge adaptive image. In *Computing, electronics and electrical technologies (ICCEET), 2012 international conference on* (pp. 1023-1027).

Stamm, M. C., Tjoa, S. K., Lin, W. S., & Liu, K. R. (2010, September). Undetectable image tampering through JPEG compression anti-forensics. InImage Processing (ICIP), 2010 17th IEEE International Conference on (pp. 2109-2112).

Shreelekshmi, R., Wilscy, M., & Wilscy, M. (2010). Preprocessing Cover Images for More Secure LSB Steganography. International Journal of Computer Theory and Engineering, 2(4), 546-551.

Singh, K. M., Singh, L. S., Singh, A. B., & Devi, K. S. (2007, March). Hiding secret message in edges of the image. In Information and Communication Technology, 2007. ICICT'07. International Conference on (pp. 238-241).

Sudha, K. L. (2012). Text Steganography using LSB insertion method along with Chaos Theory. arXiv preprint arXiv:1205.1859.

Wahab, A. W., Briffa, J. A., Schaathun, H. G., & Ho, A. T. (2009, May). Conditional probability based steganalysis for JPEG steganography. In *2009 International Conference on Signal Processing Systems* (pp. 205-209).

Wang, S. J. (2005). Steganography of capacity required using modulo operator for embedding secret image. *Applied Mathematics and Computation*, *164*(1), 99-116.

Zhang, T., & Ping, X. (2003). A new approach to reliable detection of LSB steganography in natural images. *Signal Processing*, *83*(10), 2085-2093.

Zhang, T., Li, W., Zhang, Y., Zheng, E., & Ping, X. (2010). Steganalysis of LSB matching based on statistical modeling of pixel difference distributions.Information Sciences, 180(23), 4685-4694.