جامعة الـشرق الأوسط MIDDLE EAST UNIVERSITY

Amman - Jordan

Security Enhancement of Image Steganography Using Embedded Integrity Features

التحسين الأمني لإسلوب الإخفاء في الصور بإستخدام خصائص السلامة

المتضمنة

Prepared By

Zinah Talaat Rashid AL-Windawi

Supervisor

Dr. Mudhafar Al-Jarrah

Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master in Computer Science

Department of Computer Science

Faculty of Information Technology

Middle East University

May, 2017

Authorization Statement

I, Zinah Talaat Rashid AL-Windawi, authorize the Middle East University to provide hard copies or soft copies of my thesis to libraries, institutions or individuals upon their request.

Name: Zinah Talaat Rashid AL-Windawi

Date: 30/ 5/ 2017

Signature:



إقرار تفويض

انا زينة طلعت رشيد الونداوي افوض جامعة الشرق الاوسط بتزويد نسخ من رسالتي ورقياً او الكترونياً للمكتبات او المنظمات او الهيئات و المؤسسات المعنية بالابحاث و الدراسات العلمية عند طلبها.

الاسم: زينة طلعت رشيد الونداوي

التاريخ: 30 / 5 / 2017

التوقيع: زيم

Middle East University

Examination Committee Decision

This is to certify that the thesis entitled "Security Enhancement of Image Steganography Using Embedded Integrity Features" was successfully defended and approved on 30/5/2017.

Examination Committee Members

(Supervisor)

Dr. Mudhafar Al-Jarrah

Middle East University

(Internal Member / Chairman)

Abdelrahman Moh'd A. Abuarqoub

Middle East University

(External member)

Dr. Nijad Al-Najdawi

Al-Balqa' Applied University



Signature

Acknowledgments

At first, thanks to God who gave me courage, patience and enabled me to achieve this work. Then, I would like to thank my supervisor, Dr. Mudhafar Al-Jarrah for his countless help, support, guidance, and knowledge he provided me throughout my research. I wish to express my deepest gratitude to the committee members for spending their precious time on reading my thesis. Also, I would like to thank the Information Technology Faculty members at the Middle East University. Finally, I would like to thank my family, especially my husband, my mother and my husband's mother for their support and help throughout my study years. This work would not be accomplished without them. I love you all.

بسم الله الرحمن الرحيم "وقل ربي زدني علما"

Dedication

This thesis is dedicated to all the people who never stopped believing in me and supporting me

To the pure heart, my father.

To my happiness in life, my mother which never stopped supporting me during the journey

of my life.

To the wonderful sisters.

To the beautiful brothers.

To my life partner, husband.

To the light of my eyes, to the love of my life, to my heart, my son, I wish to accomplish

my dreams with you.

To my husband's mother who encouraged me and supported me.

To my best friend and cousin, Jehan Bahjat.

Zinah

Table of Contents

Cover Page	I
Authorization Statement	
إقرار التفويض	
Thesis Committee Decision	IV
Acknowledgments	V
Dedication	VI
List of Abbreviations	IX
List of Figures	X
List of Tables	IX
List of Algorithms	XI
Abstract	XIII
الملخص	XVI
Chapter One	1
Introduction	1
1.1 Background	2
1.2 Problem Statement	
1.3 Research Questions	4
1.4 Objectives	4
1.5 Contributions.	4
1.6 Motivation	5
1.7 Thesis Organization	5
Chapter Two	7
Literature Review	7
2.1 Background and Definitions	
2.2 Types of Steganography	
2.3 Steganography Techniques	
2.4 Categories of Steganography	
2.5 Steganalysis Attacks	
2.6 Evaluation of Steganography Techniques	
2.7 Quality Evaluation Metrics	

2.8 Integrity Checking Methods	
The Checksum Integrity Method	
2.9 Grayscale Image	
2.10 Related Work	21
Chapter Three	
Methodology	
3.1 Outline of the Proposed Methodology	27
3.2 Main Functional Points of the Proposed Method	27
3.3 Design Considerations of the Proposed Model	
3.4 Data Layout of the Secret File	
3.5 Data Layout of the Cover (Stego) File	
3.6 The Alteration Detection Methods	
3.7 The Processing Method	
3.7.1 Embedding	
3.7.2 Extraction and Integrity Verification	
Chapter Four	
Experimental Results and Discussion	
4.1 Overview	40
4.2 Evaluation Metrics	40
4.3 Alterations Detection Output	40
4.4 Experimental Data Set	41
4.5 Implementation	
4.6 Experimental Work and Discussion of Results	
4.6.1 PSNR results comparison	44
4.6.2 Visual comparison	47
Chapter Five	
Conclusion and Future Work	
5.1 Conclusion	
5.2 Future Work	
References	61
Appendix A	65

Table 3.1	Example on Alteration Detection Cases Using Bit-Pairs	38
Table 4.1	PSNR Values for 30 Images	44
Table 4.2	Average PSNR Values for 1000 Stego Images	47
Table 4.3	Detection Rate for 30 Images Using Random Attacks 12800 Bytes Altered	52
Table 4.4	List of Location and Content of Altered Bytes Using Pair Comparison	54

List of Tables

List of Figures	5
-----------------	---

Figure 2.1	Steganography System Scenario	9
Figure 2.2	Types of Steganography	10
Figure 2.3	Tradeoff between Image Steganography Properties	17
Figure 3.1	Reading Secret Image as Steam of Bytes	30
Figure 3.2	Grayscale Cover Image	30
Figure 3.3	The Main Embedding Algorithm	33
Figure 3.4	Process the Secret Byte Algorithm	34
Figure 3.5	Extract-Verify Main Algorithm	35
Figure 3.6	Process Stego Bytes	36
Figure 3.7	Example on Embedding a Secret Byte with Inverse Decoy in	37
	Four Cover Bytes	
Figure 4.1	The Secret Image	43
Figure 4.2	Sample of a Clean Image	48
Figure 4.3	The Stego Image	48
Figure 4.4	The Altered Stego Image	49
Figure 4.5	The Stego Image with 2LSB Embedding in Alternative Bytes	49
Figure 4.6	Small Secret Image for 2LSB Embedding in Alternative	50
	Bytes	
Figure 4.7	The Extracted Secret Image After the Attack	51

List of Algorithms

The Main Embedding Algorithm	33
Process the Secret Bytes Algorithm	34
Extract-Verify Main Algorithm	35
Process Stego Bytes	36

List of Abbreviations

LSB	Least Significant Bit
MSB	Most Significant Bit
RGB	Red-Green-Blue
PSNR	Peak Signal -to-Noise Ratio
MSE	Mean Square Error
PGM	Portable Graymap Format
JPEG	Joint Photographic Experts Group
BMP	Bitmap Image File
HVS	Human Vision System
BPP	Bit Per Pixel
НС	Hiding Capacity
HCF	Histogram Characteristic Function
BAC	Byte Attack Count
SB	Secret Byte
CRC	Cyclic Redundancy Check
AWGN	Additive white Gaussian noise
MD5	Message Digest 5

Security Enhancement of Image Steganography Using Embedded

Integrity Features

By

Zinah Talaat Rashid AL-Windawi

Supervisor

Dr. Mudhafar Al-Jarrah

Abstract

Steganography is a security method that hides secret data inside cover media where the very existence of the embedded secret data is not perceptible. The cover object can be image, audio or video; the most commonly used is an image file.

This thesis presents a model for protecting the security and integrity of secret data embedded in grayscale images, to detect alterations to the secret data that can happen during transmission, and to protect secrecy of the secret data through adding decoy data.

A proposed model is presented in which a secret image is embedded in a grayscale cover image, together with file checksum of the secret data and an extra bit pair per byte to serve integrity verification. The secret data is read as a stream of bytes and the bytes are split into four pairs of bits, the group of four pairs are hidden inside uncompressed grayscale images using the 4LSB replacement technique. A decoy bit pair which represents the inverse of the data bit pair is combined with the data bit pair and stored in the right halfbyte of a cover byte. The decoy bit pair serves in verifying that the data bit pair has not been changed, and in protecting the secret data if an adversary manages to uncover the embedded data. Extraction of the secret file is achieved through merging the four hidden data pairs into bytes. During the extraction process, a pair comparison between data and decoy pairs is performed to detect alterations. Also, the file checksum is calculated for the extracted secret data. A checksum comparison between the embedded and re-calculated checksums is used to detect alterations, as a second detection method, in case an alteration is missed by the pair comparison. A list of location and content of the altered bytes are produced, to help in investigating the attacks. This model is implemented in the MATLAB R2015a environment. The model is evaluated using 1000 BOSSbase public grayscale images. The secret image size represents 25% of the cover image's size. The purpose of the evaluation is in two folds, detection accuracy and imperceptibility. The detection accuracy is the ratio of detected to altered bytes, which was %99.6. Measuring imperceptibility was based on the Peak-Signal-to-Noise Ratio (PSNR) metric value, between stego and clean images. The PSNR value was calculated as PSNR1, for embedding straight decoy with the data pair, PSNR2 for embedding with inverted decoy, and PSNR3 for embedding without decoy. The PSNR3 value was the highest because only 2 bits were embedded. The PSNR2 value was higher than PSNR1, due to the inversion of the decoy pair. Minor visual

differences were noticed in some clean / stego pairs, due to using grayscale images, however, it will not be observed without close examination of both images. An alternative 2LSB embedding scheme is proposed where the data and decoy pairs are embedded in alternate bytes, which has eliminated any visual discrepancy.

Keywords: steganography, secret data, stego image, embedding, extracting, pair comparison, checksum, decoy data.

التحسين الأمنى لإسلوب الإخفاء في الصور بإستخدام خصائص السلامة المتضمنة

اعداد زينة طلعت رشيد الونداوى إشراف الدكتور مظفر الجراح

الملخص

تعد تقنية "ستيغانوغرافي" اسلوب حماية، حيث تخفي البيانات السرية داخل أغطية وسائط رقمية كي لا يمكن إدراك وجود البيانات السرية المتضمنة. يمكن أن يكون محتوى الغطاء صورة أو صوت أو فيديو، والأكثر شيوعا هو الصورة.

تقدم هذه الأطروحة نموذجا لحماية أمن وسلامة البيانات السرية المضمنة في الصور ذات التدرج الرمادي، للكشف عن التغيرات على البيانات السرية والتي يمكن أن تحدث أثناء الإرسال،ايضا لحماية سرية البيانات المخفية من خلال إضافة بيانات تمويهية.

يعرض النموذج المقترح تضمين صورة سرية في غطاء صورة ذات تدرج رمادي، إلى جانب المجموع الاختباري لملف البيانات السرية وزوج بتات إضافي لكل واحد بايت من أجل التحقق من سلامتها. يتم قراءة البيانات السرية كتيار من وحدات البايت ويتجزأ البايت إلى أربعة أزواج من البتات، حيث تخفى مجموعة الأربعة أزواج داخل الصور غير المضغوطة وذات التدرج الرمادي باستخدام تقنية استبدال LSB4 . ويتم الجمع بين زوج البتات التمويهية الذي يمثل معكوس زوج بتات البيانات مع زوج بتات البيانات ويخزن في نصف البايت الأيمن لبايت التغطية. ويساعد زوج البتات التمويهي في التحقق من أن زوج بت البيانات لم يتغير، وايضا في حماية البيانات السرية إذا تمكن الخصم من كشف البيانات المضمنة.

يتحقق استخراج الملف السري من خلال دمج أزواج البيانات الاربعة المخفية في وحدات البايت. خلال عملية الاستخراج، ويتم إجراء مقارنة الازواج بين البيانات والازواج التمويهية للكشف عن التغييرات. أيضا، يتم حساب المجموع الاختباري لملف البيانات السرية المستخرجة. وتستخدم مقارنة للمجموع الاختباري بين القيمتين المضمنة والتي تم إعادة حسابها للكشف عن التغييرات، كطريقة كشف ثانية، في حالة غياب كشف التغيير من قبل مقارنة الازواج. يتم إنتاج قائمة الموقع ومحتوى البايتات المتغيرة، للمساعدة في التحقيق في طبيعة الهجمات. يتم تنفيذ هذا النموذج في بيئة ماتلاب و2015a المتغيرة، للمساعدة في التحقيق في طبيعة الهجمات. من الصور الرمادية العامة من مجموعة (BOSSbase).

يمثل حجم الصورة السرية 25% من حجم صورة الغطاء ويتمثل الغرض من التقييم في شقين، دقة الكشف، وعدم المحسوسية. دقة الكشف هي نسبة الكشف عن البايتات التي تم تغييرها، والتي كانت / 99.6. وقد استند قياس عدم المحسوسية إلى قيمة المقياس القصوى إلى نسبة إشارة التشويش (PSNR) بين الصور المخفية والصور النظيفة. تم حساب قيمة PSNR1 ك PSNR1، لتضمين التمويه المباشر مع زوج البيانات، PSNR2 للتضمين مع التمويه المعكوس، و PSNR3 للتضمين دون تمويه. وكانت قيمة PSNR3 هي الأعلى نظرا لتضمين اثنان بت فقط. وكانت قيمة PSNR2 أعلى من PSNR1، ويرجع ذلك إلى انعكاس زوج التمويه. وقد لوحظت اختلافات بصرية طفيفة في بعض أزواج نظيفة / تمويهية، وذلك بسبب استخدام الصور ذات تدرج رمادي، ومع ذلك، فإنه لن يلاحظ دون فحص دقيق لكلا الصورتين. أقترح مخطط التضمين البديل LSB2 بحيث يتم تضمين أزواج البيانات والبايتات التمويهية، مما أدى إلى إزالة أي تناقض بصري.

الكلمات المفتاحية: ستيغانوغرافي، بيانات سرية، صورة مخفية، تضمين، استخراج، مقارنة الازواج، المجموع الاختباري، بيانات تمويهية. **Chapter One**

Introduction

1.1 Background

Data security has gained more attention recently due to the rise in cyber espionage, and the massive increase in data transfer rate over the internet which resulted in more documents being exchanged in digital form. Security of data requires protecting data from access, modification, sharing or even viewing by unauthorized users, allowing only authorized users for such access. Data hiding is an approach that aims to protect data through concealing its existence from adversaries, but this approach needs strengthening to prevent an attacker from access to data in case the existence of hidden data is detected by analytical means.

There are many areas of security technology that deals with the protection of secret data; the most important of these techniques are cryptography and steganography.

The first technique is cryptography which is referred to as "the study of secret". It includes encryption and decryption processes, Encryption is the process of converting normal text to unreadable form, where the sender uses an encryption key to encrypt the message to transmit it through the insecure public channel. Decryption is the process of converting encrypted text to normal text in the readable form, thus the reconstruction of the original message is possible only if the receiver has the decryption key (Thakur & Kumar, 2011).

The second technique is steganography which is defined as a method of security that hides data among the bits of a cover file, where the secret message is inserted in another medium so that the very existence of the secret message is not detectable. The cover file can be image, audio or video; the most commonly used being the image files, in which unused or insignificant bits are replaced with the secret data (Singh & Siddiqui, 2012).

Cryptography differs from steganography in that cryptography is implemented by changing the data into a form that cannot be understood, while steganography is implemented by hiding the data itself. This research work focuses on protecting data security through hiding the data using steganography techniques, and detecting adversaries changes or modification to the data through using integrity enhancement features.

1.2 Problem Statement

The steganography approach relies on hiding secret messages inside innocentlooking messages or documents in order to dissuade the enemy from attempting to find the secret message. However, what if the enemy discovered the secret message and decided on a deception act by changing the secret message in some ways so that to feed the intended recipient with disinformation (disinformation is intentionally false or inaccurate information that is spread deliberately).

The main problem area to be tackled in this work is the detection of attackeralteration of a hidden secret message.

The research work deals with the problem of enhancing the steganography layer of a secret protection method by adding an integrity verification layer that will help to identify possible modifications of a secret message that was not part of the original message.

1.3 Research Questions

- 1. Is it possible to enhance the integrity of a secret message that is embedded in a carrier image, in case of detection of the secret message.
- 2. Is it possible to enhance the secrecy of a secret message that is embedded in a carrier image, in case of detection of the secret message.
- 3. Can the alterations to a secret message by an attacker be detected.
- 4. Will the inclusion of integrity features degrade the carrier image quality.

1.4 Objectives

The aim of this work is to enhance the integrity and security of the hidden secret message. To realize this aim, integrity verification data will be added to the secret data, so that any alteration to the secret message during transmission between sender and receiver are detected. The added integrity verification data will serve in strengthening of the secret message's secrecy, in case the existence of the secret message is detected.

1.5 Contributions

Enhancing security and integrity of the hidden secret data as in points below:

- 1. Using pair comparison and checksum make it possible to detect any alteration to the secret message.
- 2. Using pair comparison helps the user to identify location of alterations and contents of the altered bytes, for various purposes such as identifying patterns of the attacks.

3. Using an added decoy data helps to improve secrecy of the hidden secret message if it is uncovered.

1.6 Motivation

The steady increase in malicious attacks on private, business and government documents by adversaries of various kinds has motivated researchers and developers in the information security field to seek technical solutions to protect the privacy of documents sent over communication channels.

The present work is motivated by the need for a more secure solution for protecting secret data that is being transmitted over communication channels, to strengthen the privacy and integrity of the secret data.

1.7 Thesis Organization

This thesis contains five chapters:

Chapter one presents an introduction to the steganography, the problem statement, research questions, objectives, contribution and motivation.
Chapter two presents background overview of steganography, types of

steganography, steganography techniques, and related work.

• Chapter three presents the proposed work and methodology, design considerations of the proposed model, data layout of the secret file, data layout of the cover file, embedding and extracting algorithm.

• Chapter four presents implementations of the proposed model, the experimental work and discussion of results.

• Chapter five presents conclusions and future work.

Chapter Two

Literature Review

2.1 Background and Definitions

Steganography technique is generally defined as the art and science of writing hidden messages without anyone noticing the existence of this message, other than the sender (steganographer) and the intended recipient. Steganography is originally a Greek word that means concealed writing. The word "Steganography" is divided into two parts: Steganos which means "secret or covered" (where you want to hide the secret messages) and the graphy that means "writing" (text). Although different definitions of steganography exist, but the concept is one which means the hiding of sensitive information or secret messages into another media file such as image, text, sound, and video (AL-Shatnawi & AlFawwaz, 2013 and Vaman, et al., 2013).

The main purpose of using the steganography technique is to avoid attracting attention to the transmission of hidden information. However if doubt is increased about the contents of a document, the goal of concealing a hidden secret inside that document becomes less likely to achieve its objectives because once an observer notices any change in the sent document, he will try to know the hidden information inside the document.

The main components used in steganography systems are:

- **Cover message** (is the carrier of the secret message such as image, video, audio, text, or some other digital media)

- Secret message (is the information which needs to be hidden in a suitable digital media cover).

- Secret key (is used to control access to the hidden data).

- Embedding algorithm (is the way that is usually used to embed the secret data in the cover message).

- Extracting algorithm (is the way to extract the hidden data from the stego file).

In the Steganography system scenario, before the hiding process, the sender must choose the right message carrier, i.e image, video, audio, text, and then choose the effective secret messages as well as the strong password (which supposed to be known by the receiver). The effective and suitable steganography algorithm must be chosen that is capable of encoding the message in a more secure method. Then the sender can send the stego file by email or chatting, or by other modern techniques. The stego file is the cover document which carries within it a message with the secret information. After receiving the message by the receiver, the message can be decoded by using the extracting algorithm and the same password used by the sender (AL-Shatnawi & AlFawwaz, 2013). The steganography system general scenario is shown in figure 2.1.



Figure 2.1: Steganography System Scenario

2.2 Types of Steganography

Almost all digital file formats can be used in the steganography process, but the formats that are more suitable than other formats depends on the redundancy level which is available. The redundant bits of an object are those bits that can be changed without easily detecting this change. In fact, the most used carrier file on the internet is digital images. (Hamid, Yahya, Ahmad & Al-Qershi, 2012).

Text steganography is the hardest type of steganography compared with the other types of steganography because of the low degree of redundancy in text as compared to image, audio or video. Redundancy can be described as the bits of a media signal or file that provide more image accuracy than needed (Channalli & Jadhav, 2009).

Steganographic systems use media objects as cover medium such as video, image, audio and text. Digital images are often used in sending out pictures by email and other Internet communication (Bahirat & Kolhe, 2014).

Steganography can be classified into four types, as shown below:



Figure 2.2 Types of Steganography

2.2.1 Text Steganography

Hiding information in this method is historically the most important method of steganography. This method is an obvious method to hide a secret message in every *nth* letter of every word of a text message. This method has decreased in importance only since the beginning of the internet and all the different digital file formats. This method is not used very often because text files have a very small amount of redundant data (Morkel, Eloff, & Olivier, 2005).

2.2.2 Image Steganography

When taking the cover object as image to hide the secret data in steganography is referred as image steganography. In this type pixel intensities are used to hide the information. In digital steganography technique the images are widely used cover source because there are number of bits display in digital representation of an image (Singh & Kaur, 2015).

2.2.3 Video Steganography

Most of the presented techniques on images and audio can be applied to video files also because Video files are generally a combination of images and sounds. Video files have the great advantages which is the large amount of data that can be hidden within video file and the fact that it is a moving stream of images and sounds. Therefore, any small but otherwise noticeable distortions might go unobserved by humans because of the continuous flow of data (Kaur, Kaur & Singh, 2014).

2.2.4 Audio Steganography

In this type, the secret data are embedded in an audio file. The technique used in this type is disguising, which takes advantage of the ability of the human ear to hide information inconspicuously. In audio steganography soft audible sound can go undetected in the presence of another loud audible sound. The large size of audio files makes audio steganography is less preferable. (Chavda, Doshi, & Deulkar, 2014).

2.3 Steganography Techniques

2.3.1 Spatial Domain

The secret messages are embedded in the cover file directly. In the spatial domain, the least significant bits (LSB) are replaced with bits from the secret message. (Shelke, Dongre, & Soni, 2014).

Noticing the slight difference of colors is not easy. Therefore this method exploits the natural weakness of Human Visual System (HVS) therein. This method can be used in grayscale image and color image by changes some the 8 bit of image's data in the grayscale image so that the alteration of image's is not perceptible for human eyes. Also, when using RGB image the least significant bit of each color components can be used. Therefore, the potential capacity for hiding secret data in a RGB image is triple of the image which has the same size in the grayscale format (Bashardoost, Sulong, & Gerami 2013).

Advantages of the LSB technique are that the original image degradation is not easy to detect and the hiding capacity is more, which means more information can be stored in an

image object. Disadvantage of spatial domain of LSB technique is the robustness is low therefore the hidden information can be destroyed by attacks (Devi, 2013).

2.3.2 Transform Domain

The secret data is hidden in cover document by modulating such as: Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT) (Goel et al., 2013), as follows:

1. Discrete Cosine Transform Technique (DCT):

DCT is mathematical function that transforms digital image data from the spatial domain to the frequency domain. In this type, after transforming the image in frequency domain, the secret data is embedded in the least significant bits of the medium frequency components and it is specified for lossy compression.

2. Discrete Wavelet Transform Technique (DWT)

It is a mathematical function which transforms digital image data from the spatial domain to the frequency domain.

2.4 Categories of Steganography

Steganoraphic systems can be classified into three different categories depending upon the embedding and extraction procedures used (Mishra, Mishra & Adhikary, 2014), as follow:

- 2.4.1 Pure steganography (or No Key Steganography NKS): The secret message is hidden in a cover image directly without using any key. This form of steganography is the simplest and weakest. The success of this hidden communication depends upon the assumption that the attackers are not aware this cover contains the secret message.
- 2.4.2 Secret Key Steganography (SKS): Using secret key, the secret message is embedded into cover image and extracted out of the stego image. Both the receiver and transmitter have common agreed upon these keys in this type of steganography. The keys can be separately shared between both the receiver and the transmitter using some particular channel prior to the real transmission begins. Higher security is the strength of this system. Parties other than the intended receiver cannot recover the secret message or will require very high computational time and capacity to retrieve the secret message applying some brute force methods. The robustness of this system lies with the secrecy of the keys and the hardest part in this type of steganography is how to share the keys between the transmitter and the receiver with maintaining their secrecies.
- **2.4.3 Public Key Steganography (PKS):** To hide the secret information, this method of steganography uses a pair of public key and private key. In order to be capable to extract the hidden information, the parties other than the intended receivers need to know both the private key and public key used for embedding and the encryption algorithms used. Therefore, this method is robust.

2.5 Steganalysis Attacks

Steganalysis is the science of detecting secret messages hidden in a cover object using steganography techniques. The goal of steganalysis is to discover the presence of embedded message and to break the security of its carrier (Nissar & Mir A., 2010). There are three different types of steganalysis attacks as follow:

- **2.5.1 Visual attacks:** this type of attacks represents the easiest form of steganalysis. Visual attacks examine the stego files to detect any alteration may be noticed through comparison between cover image and stego image by the naked eye to see the difference between them(Qasem, 2014)
- 2.5.2 Statistical attacks: with true statistical analysis, we can determine if an image has been changed or not. Visual analysis and statistical analysis are two major techniques included in steganalysis. Visual analysis attempts to detect the presence of hidden data through inspection by the naked eye or ear in the case of sound. Statistical analysis tries to reveal tiny alterations in a carrier objects statistical behavior caused by steganographic embedding (Sarayreh, 2014). In this type, the attacks may be passive attack or active attack. A passive attack is used to identifying presence the secret message or absence the secret message or embedding algorithm used. An active attack is used to examine embedded message length or hidden message location or secret key used in embedding (Devi, 2013).
- **2.5.3 Structural attacks:** The data files format changes as the hidden information that is embedded; identifying these changes of characteristic structure can help

us to detect the presence of image file (Devi, 2013). For instance, when adding an alpha channel to an RGB BMP stego image the structure will change to 32 bits without changing the format file (Al-Bayati, 2016).

2.6 Evaluation of Steganography Techniques

Steganography techniques can be characterized into three properties: 1) imperceptibility 2) robustness 3) hiding capacity, as follows:

2.6.1 Imperceptibility

The main goal of steganography is imperceptibility. A person when views a cover object should not be able to distinguish that the cover object contains embedded information or not contains embedded information. The goal is that the cover medium before hiding secret information and after hiding secret information should appear identical (Bahirat & Kolhe, 2014).

2.6.2 Robustness

It refers to the degree of difficulty required to destroy embedded data without destroying the cover image (Sumathi, Santanam & Umamaheswari, 2013).

2.6.3 Capacity

It refers to the maximum amount of information that can be hidden inside the cover image. It is represented in bits per pixel (bpp) (Swain & Lenka, 2014).

These properties are used to measure the performance of the steganographic system. It is not possible to maximize imperceptibility, robustness and capacity at the same time; therefore, these items must meet an acceptable balance by the application. Imperceptibility becomes the most important requirement when steganography technique is used as a way for hiding communication, while robustness and possibly capacity can be sacrificed (Hamid N. et al., 2012). A tradeoff between those properties is shown in figure 2.3 below.



Figure 2.3: Tradeoff between Image Steganography Properties (Hamid N. et al., 2012).

2.7 Quality Evaluation Metrics

Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE) are two metrics used to evaluate the quality of an image. PSNR represents the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. This ratio evaluates the quality of a cover image embedded with secret data and a reference image. MSE represents the average squared difference between a reference image and a distorted image. The smaller value of MSE represents less differences between the two images, which results in higher PSNR according to equation (1). The MSE value is computed pixel-by-pixel by adding up the squared differences of all the pixels and dividing by the total pixel count (Singh & Kaur, 2015). The MSE value is calculated in equation (1) below, and the value of PSNR is computed in the equation (2).

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||O(i,j) - D(i,j)||^2 \qquad \dots \dots (1)$$
$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{MSE}\right) \qquad \dots \dots (2)$$

The symbol O is the original image pixel value and the symbol D is the distorted image pixel value, and "m \times n" is the size of image. The MAX is 255 which is the peak value of the pixels in an image when pixels are presented in an 8-bit format (Yalman, 2013).

2.8 Integrity Checking Methods

When data is transmitted between sender and receiver, the data can be altered during the transmission process, either unintentionally as a result of communication channel problems, or intentionally through the action of an adversary. Many integrity checking techniques have been used such as parity checking, CRC and similar methods. The most widely used integrity verification method is the checksum technique (Sivathanu, Wright & Zadok, 2005), which is often used to verify downloads from websites.
The Checksum Integrity Method

Integrity of transmitted data can be verified by comparing the stored check-sum values with the newly calculated checksum values for each data read. This method is generated using a hash function method.

The checksum method cannot help in recovery of data but can help in detecting integrity violations for two reasons. The first one, the mismatching between the saved checksum value and the calculated checksum value just means that one of these values was modified without offering information about which of them is legitimate. Stored checksums can be corrupted or modified. The second reason for recovery data problems of checksum is that it is generally calculated by using a one-way hash function and the data cannot be reconstructed to offer values of checksum (Sivathanu et al., 2005).

The purpose of using checksum in the present work is to verify that the extracted data is exactly the same as the original embedded secret data without any alterations by adversaries during the transmission process.

2.9 Grayscale Image

Grayscale image formats are used to represents images with picture elements that range from black to white, through several gray levels or shades.

A grayscale image can be of one channel (8 bit depth) or of three channels (24 bit depth). The most widely used grayscale image format is the one channel (8 depth) method. This format is used in research in information hiding (Ker, 2005). There are several formats for 8-bit grayscale images; the main formats are BMP, PGM and GIF.

In this research, we are going to deal with grayscale images of the PGM (Portable Graymap Format) format, which has been used in many research projects in information hiding, such the work that uses the BOSSbase 1.01 dataset (BOSSbase, 2017).

The following example demonstrate the hiding of data within a one-channel grayscale image. Suppose the original image eight pixels have the following grayscale values:

To conceal the binary value 10000011 that refers to the letter C, we would replace the LSBs of these pixels to have new grayscale values, as fallowing:

The difference between the cover image (original image) and the stego image will be hardly noticeable to the human eye (Goel et al., 2013).

2.10 Related Work

This part describes several previous studies which used the LSB technique, to improve the security of the embedded secret message, or to improve the capacity of the cover image.

The thesis by Qasem (2014) is based on the spatial method, it presents two models by 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. (Embed-All) and (Embed-Odd) are two algorithms presented in this thesis. The first algorithm (Embed-All) which stores the hidden image in the RGB channels of successive pixels (odd and even pixels). A hiding capacity of this algorithm gives of 50% of the available pixel capacity. The second algorithm (Embed-Odd) which 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, in the odd pixel to neutralize the color change, and to add noise to the even pixel in order to confuse the attacker. The presented two algorithms were implemented in Matlab 2012b, and used standard images as cover such as Lena, but 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 this work does not show any noticeable difference to the human eye, even for the successive pixels method (Embed-All). This thesis used image comparison metrics such as PSNR which has shown acceptable distortion values even when hiding to the maximum capacity of an image.

The paper by Kekre et al (2011) proposed a steganalysis method based on advantages that are extracted from co-occurrence matrix of an image. Two unlike distance measures: Absolute distance and Euclidean distance are used for the purpose of classification. This scheme beats previous works in steganalysis for LSB hiding. It works in case of both grayscale image and color image. The results using Euclidean distances are better than using Absolute distance by 265% in color images and by around 329% in grayscale images. Detection accuracy in case of the color images is better than that of grayscale images by nearly 18% in Absolute distance and almost same in Euclidean distance. Supremacy is observed for low embedding rates. The feature vectors which consist of the diagonal d0 exhibit poor results as compared to feature vectors that do not comprise the diagonal d0.

The paper by Salih and Al-Jarrah (2015) introduced several steganography methods to ensure secure use of internet, the current those methods cannot verify the presence of attacks in secret messages. Thus, this paper introduces the development of an advanced Least Significant Bit (LSB) technique; Bi-LSB to solve the low security and capacity problems of the traditionally used LSB techniques.

The proposed technique is evaluated based on adding an AWGN to the stego file before extracting the embedded messages to analyze its effect on the PSNR values and then comparing the extracted message with the original one based on checking the integrity. The Results show that there is an obvious reduction in the values of PSNR after adding the AWGN attack.

The thesis by Sarayreh (2014) investigated the hiding of text messages and documents within the alpha channel of RGBA color images. It is implemented in two phases: the first phase which the secret text is stored as bits in the LSB part of the alpha channel. while the second phase is to separate the alpha channel from the RGB channels of the stego image, and to attach the alpha channel to a different image, a semi-stego, with different RGB channels values, in other words to Swap the alpha channel of the two unrelated images. The proposed model discussed hiding capacity in the 3 bits per pixel (bpp), which is the same as changing one bit per color channel in an LSB method, but it has the advantage that no change can be detected in the analysis of the RGB channels. in the experimental work, The PSNR results for the maximum hiding capacity is considered acceptable as it is well above 30, and the proposed SWAP procedure will improve undetectability by separating the alpha channel containing the secret text from the indicator RGB channels.

The paper by Laskar and Hemachandran (2012) employed a method for applications that require high-volume embedding with robustness against certain statistical attacks. This method which presents 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 using LSB steganographic method the embedding capacity increases and thus we can 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 secret message which is hidden inside the stego-object, the attacker would still require the cryptographic decoding key to decipher the encrypted message. The main aim in this paper 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.

The paper by Ker (2005) uses the HCF method which in the detection of steganography in grayscale images. Two novel ways of applying the HCF are introduced: calibrating the output using a downsampled image and computing the adjacency histogram instead of the usual histogram. Extensive experimental results show that the new detectors are reliable, vastly more so than those previously known.

In conclusion, published research on steganography has focused on various techniques for enhancing the hiding of secret data in cover media. However, it did not consider integrity problems that can occur if the cover media is modified by an attacker. Therefore, the proposed work addresses the integrity verification feature that can be combined with the steganography method.

Chapter Three

Methodology

3.1 Outline of the Proposed Methodology

The methodology adopted in this thesis is based on experimental work for embedding a secret image within a cover image using the LSB method, attacking the resulting stego image, and verifying integrity of the stego image after the attack. The experimental data is based on using existing public image datasets that are the product of academic research. The proposed model is implemented in the MATLAB environment. Implementation of the proposed model is divided into three modules: Embed: deals with storing the secret message in a cover image and calculates the checksum; Attack: deals with modifying the stego image by randomly replacing bits; and Extract-Verify: deals with extracting the secret message as well as checking the integrity of the extracted secret message.

3.2 Main Functional Points of the Proposed Method

The main aim of the proposed work is to enhance the security of a secret message sent over communication networks by combining several functional points:

- 1. Concealing the secret message in a grayscale image which is sent over communication channel in order to prevent a potential attack by an adversary.
- 2. Adding decoy data, in order to confuse the attacker in case the presence of a secret message is detected by some steganalysis tool, and the attacker attempted to uncover the secret message.
- The embedding process should result in a stego image that is less likely to be detectable by meeting un-detectability criteria such as the visual imperceptibility and the PSNR metric.

- 4. Adding a checksum to the stego image in the embedding process, and calculating a second checksum during the extraction process. The checksums will be used to detect if any alteration has happened by comparing the two checksums.
- 5. The recovered secret file should be equal to the original secret file in contents and format if the stego has not been attacked.
- 6. Apply an attack to produce changes in the stego image that needs to be detected by the extraction process.
- 7. Extract the secret message and verify its integrity using the checksum comparison.
- 8. In case an attacked is detected, identify locations of bytes of the secret image that has been changed, by comparing the data bit pair and the decoy bit pair in every byte.

3.3 Design Considerations of the Proposed Model

- 1. The grayscale cover images are used to store the hidden secret multimedia files and the decoy data which is used to confuse the attackers on the real data. The hiding capacity will be 50% of the available data area of the stego image, where 25% of the hiding area will be used to embed the secret message, and 25% for storing the decoy data, which will be used in the integrity checking. Two bits of the real secret data and two bits of the decoy data will be stored in the right half-bytes of the stego images, without alteration to the left half-bytes to avoid perceivable visual distortion.
- During the embedding process, the secret multimedia file will read as a stream of bytes, regardless of its format.

- 3. The secret multimedia file which is embedded in grayscale cover image will be split vertically into four fragments, i.e. each fragment contains two bits.
- 4. The two-bit fragments of the secret message will replace 2-LSB fragments of bytes of the stego image. The third and fourth LSB bits of each byte will be used to store the two-bit fragments of the decoy data, which will be an inverted copy of the original secret data fragment.
- 5. Comparing the cover image with the stego image, the PSNR results should be identical to results produced by acceptable standard image comparison software such as Imagemagic.
- 6. The maximum hiding capacity that a grayscale cover image can store a secret image using 2-LSB, is:

HC = Width x Height / 4

For example, a cover image of 512×512 resolution (262.144 bytes) can be embedded with up to 65,536 bytes (64 kb).

3.4 Data Layout of the Secret File

The secret multimedia file which will be embedded in the cover image is processed as a stream of bytes; where each byte pixel is split into four two-bit fragments (bit pairs), and each fragment will be stored in 2-LSB bits of bytes of the cover image. As shown in figure 3.1



Figure 3.1: Reading Secret Image as a Stream of Bytes

3.5 Data Layout of the Cover (Stego) File

Each pixel of the grayscale cover image consists of one byte (8 bits), where only the right half-byte (4 bits) of each byte will store the embedded data. The least significant two bits of the right half-byte (q1) will store the secret message fragments, while the most significant two bits of the right half-byte (q2) will store the decoy data. As shown in figure 3.2



Figure 3.2 Grayscale Cover Image

3.6 The Alteration Detection Methods

Two techniques are used for detecting an alteration to the embedded secret message, during the extraction and verification process:

- **3.6.1** Pair comparison: using the right half-byte (4-LSB) for hiding the secret data, the right two-bit fragments (2-LSB) is used for hiding the original secret data and other two bit fragments (2-LSB) is used to embed the decoy data which is an inverse of the two-bit fragments of secret data. During the verification process, the data bit pair and the decoy pair of an extracted bytes of the secret message will be compared, after the decoy is re-inverted, and the two pairs should match if the byte is clean, i.e. it has not been altered. In case of pairs mismatched, that byte is flagged as an altered byte. This process will produce a list of the locations of altered bytes.
- **3.6.2** Checksum comparison: checksum function is used in the implementation of the proposed model to detect any alteration to the secret data. Some special cases of alterations that cannot be detected by the pair comparison, the checksum comparison will detect by comparing the embedding checksum and the extracting checksum values.

3.7 The Processing Method

3.7.1 Embedding

The secret data will be processed one byte at a time, regardless of its file type format, and then each byte is split into pairs of bits. Each secret bit pair is stored in the LSB bit pairs of the next available channel. A checksum of the secret file will be generated using the MATLAB function 'get-file-checksum', which implements the MD5 checksum algorithm (Sivathanu.et al, 2005). The file checksum will be embedded in the stego file, for later comparison at the extraction / verification stage. Alternatively the checksum value can be saved in a file for later use by the extract-verify module. The embedding process proceeds as in the following algorithms.

A-The Main Embedding Algorithm



Figure 3.3: The Main Embedding Algorithm

B- Process the Secret Bytes Algorithm



Figure 3.4: Process the Secret Bytes Algorithm

3.7.2 Extraction and Integrity Verification

The stego file bytes will be processed starting from the initial location of hiding.

Each group of four 2bit pair extracted from four bytes of the stego image will be combined into one secret byte. The extraction process will proceed as in the following algorithms:



A- Extract-Verify Main Algorithm

Figure 3.5: Extract-Verify Main Algorithm

B- Process Stego Bytes



Figure 3.6: Process Stego Bytes

Example on embedding a secret byte with inverse decoy in four cover bytes



E	tra	cteo	i By	vte (''N	L'')	
0	1	0	0	1	1	0	1
Left Half Right Half							

Figure 3.7: Example on Embedding a Secret Byte with Inverse Decoy in Four Cover Bytes Summary of detection cases: 12 cases of flips are detected by pair comparison, and 3 cases

of flips are detected by checksum comparison, as shown in Table 3.1.

Table 3.1: Example on Alteration Detection Cases Using Bit-Pairs

• • • •			
One Side Chang	e (Detected by T	wo Pairs comp	arison)
Decoy Bit-Pair		True Bit-Pair	
Bit1	Bit2	Bit1	Bit2
0	1	0	0
1	0	0	0
1	1	0	0
0	0	0	1
0	0	1	0
0	0	1	1
Decoy Bit-Pair Bit1 1 0 1 1 0 1	Bit2 0 1 1 1 1 0	True Bit-Pa Bit1 0 1 0 1 1 1 1	air Bit2 1 0 1 0 1 1 1
Two Sides Same Decoy Bit-Pair	Changes (Detect	ted by CheckSur True Bit	m Comparison) -Pair
Bit1	Bit2	Bit1	Bit2
0	1	U	1
0	T	0	T

Chapter Four

Experimental Results and Discussion

4.1 Overview

The proposed model was implemented in MATLAB R2015a environment. The R2015a version was chosen because the required file checksum function was not available prior to 2014. The experimental work used uncompressed grayscale images as cover objects of the PGM format. The implemented system utilized the least significant bits (LSB) replacement method which used 4 LSB bits to hide the secret data and the integrity verification /decoy data.

4.2 Evaluation Metrics

The performance of the proposed model is evaluated using the following metrics:

4.2.1 PSNR: the PSNR value gives a measure of the distortion in a stego image in comparison with the clean image.

4.2.2 Detection rate: ratio of the number of detected altered bytes to the actual number of altered bytes during the attack phase. It is used in evaluating the detection performance during the experiments.

4.3 Alterations Detection Output

The implemented system provides the following alteration detection output:

4.3.1 Pair comparison detection list: This list contains the byte position within an extracted secret image of every byte that has been detected, during the extract-verify phase, which has been altered during an attack. It is generated through pair

comparison between the original secret bit pair and the decoy bit pair. An image that has a zero detection list is an un-attacked image.

4.3.2 Checksum mismatch result: The implemented system compares the embedded checksum that is added during embedding process, with a checksum calculated during the extraction process. A checksum mismatch output is an indication that an alteration of the stego image has occurred. The checksum value for the image is generated using a function call in MATLAB.

4.4 Experimental Data Set

To evaluate the proposed model we used the BOSSbase1.01 dataset which contains 10000 grayscale 8 bit PGM images. The first 1000 images of the dataset were chosen for experiment. The dataset was downloaded from the URL address our (http://dde.binghamton.edu/dounloaded/). It is a research dataset developed by the Digital Data Embedding Lab, New York University of Binghamton. The dimensions of these image are 512 width x 512 height, which resulted in 256 KB image size. In this research, the secret image (Girl.BMP) was embedded within all of 1000 images, using the spatial domain LSB technique. Appendix A contains a sample of 1000 images from the BOSSbase1.01 dataset. The secret image that was used in the experiment is "Girl.BMP", whose image size is 59.4 KB and dimension is 142 x 142 pixels, was downloaded from USC-SIPI image Database (2017).

4.5 Implementation

The implementation of the experimental work consists of the following three modules:

- **4.5.1 Embedding module:** This module performs batch embedding of the secret image inside the 1000 grayscale PGM cover images, using the 4-LSB steganography method. Each byte of the secret image is split into 4 bit pairs, where each bit pair is stored in the 2-LSB bits of the stego byte. A copy of the secret bit pair is inverted and stored as a decoy data in the left most 2 bits of the right half-byte of the stego byte. Also, the checksum value for the secret data is calculated and embedded in the stego image.
- **4.5.2 Attack module:** this module modifies the stego image by replacing bits of the right half-bytes, where the secret data is stored, with random values between 0 and 15, as a simulated attack on the stego image. The number of attacked bytes will be fixed to a certain number so that it will be possible to evaluate the detection accuracy by comparing the number of detected attacked bytes with the number of real attacks.
- **4.5.3 Extract-Verify module:** this module extracts the secret image and at the same time verifies its integrity using the checksum comparison and the pair comparison of secret bit pair and decoy bit pair. The checksum comparison is based on matching the embedded checksum value and the checksum value that is generated in the extraction phase. In addition, this module produces a list of byte positions of bytes in the extracted secret image that have been detected as being altered bytes.

4.6 Experimental Work and Discussion of Results

In this experiment, the secret image (Girl.bmp), shown in figure 4.1, was embedded in the 1000 grayscale images of the BOSSbase1.01 dataset. The embedding process was repeated three times: in the first run, a straight decoy was embedded with the secret data; in the second run an inverted decoy was inserted with the secret data; and in the third run the secret data was embedded without a decoy.



Figure 4.1: The Secret Image

4.6.1 PSNR results comparison

Table 4.1 shows the three PSNR values for the first 30 images of the dataset, while Table 4.2 shows the average of PSNR values for the 1000 images. It is not unexpected that PSNR3 (for the case of embedding without decoy) would be much higher than PSNR1 and PSNR2, because only 2 bits per bytes were replaced in the PSNR3 case compared to 4 bits in the other cases.

However, it is worth noting that PNSR2 (embedding an inverted decoy) is higher than PSNR1 (embedding straight decoy), which indicates that inverting the decoy data has the added advantage of better imperceptibility as well as the stronger effect in camouflaging the secret data.

Cover image PSNR1 embedding		PSNR2 embedding	PSNR3 embedding
	with straight decoy	with inverted decoy	without decoy
1.pgm	29.42934	32.71444	43.07699
2.pgm	31.23558	33.36138	44.32859
3.pgm	30.62829	33.12095	44.05103
4.pgm	30.80837	33.20892	44.18352
5.pgm	31.20018	33.38259	44.24337

Table 4.1: PSNR Values for 30 Images.

6.pgm	31.05044	33.17394	44.31579
7.pgm	30.99863	32.70233	44.31576
8.pgm	31.03489	33.20105	44.33398
9.pgm	31.12844	33.25388	44.32324
10.pgm	31.1861	33.32287	44.33269
11.pgm	30.97208	33.23561	44.2968
12.pgm	28.20471	30.34952	41.40652
13.pgm	31.14236	33.42306	44.25829
14.pgm	31.11022	33.19963	44.29678
15.pgm	30.97415	33.30698	44.28409
16.pgm	30.09848	32.33202	43.23458
17.pgm	30.12707	33.08597	43.65041
18.pgm	31.08214	33.18784	44.3509
19.pgm	31.12556	33.25451	44.35945
20.pgm	31.00561	33.15437	44.21923

21.pgm	29.89013	32.14829	43.07374
22.pgm	31.10964	33.54375	44.35259
23.pgm	30.9307	33.19923	44.04722
24.pgm	31.1628	33.25672	44.33975
25.pgm	30.89126	33.26037	44.284
26.pgm	31.0805	33.21024	44.33093
27.pgm	31.08056	33.23125	44.33699
28.pgm	30.94401	33.41468	44.131
29.pgm	31.09565	33.26495	44.31866
30.pgm	31.21316	33.36413	44.33865

PSNR of stego with straight decoy	PSNR of stego with inverted decoy	PSNR of stego without decoy
30.6186	32.82157	43.86093

Table 4.2: Average PSNR Values for 1000 Stego Images.

4.6.2 Visual comparison

Figure 4.2 shows a sample of a clean image, figure 4.3 shows the stego image, and figure 4.4 shows the altered image in which 51,200 bytes were altered. There is no evident difference between the clean and stego images despite the 4LSB replacement. The altered stego image does not show any evident difference from the stego image. To eliminate any possible image difference, an alternative embedding method was used in which the data decoy pairs were stored in alternate bytes using 2LSB. figure 4.5 shows the stego image that was generated with alternate embedding, using the secret data image Warbler.jpg shown in figure 4.6. However, using alternate embedding reduces the maximum secret data size by 50% compared with 4LSB method.



Figure 4.2: Sample of a Clean Image



Figure 4.3: The Stego Image



Figure 4.4: The Altered Stego Image



Figure 4.5: The Stego Image with 2LSB Embedding in Alternative

Bytes



Figure 4.6: Small Secret Image for 2LSB Embedding in Alternative

Bytes

4.6.3 Attack detection results

The stego images were attacked (altered) using the attack program, in which 51,200 bytes of each stego image were altered. The alteration replaced the 4LSB bits of the bytes with random values in range the 0-15. The 1000 altered stego images were subsequently processed by the Extract-Verify program, to extract the hidden secret message and to detect any integrity violation.

Figure 4.7 shows the extracted secret image after the attack. The attack is obvious to notice just by viewing the extracted image, it is shown here as a visible demonstration of an

altered document. However, in a real-world application, attacks on documents might not be noticeable, depending on the format and contents of the document.



Figure 4.7: The Extracted Secret Image after the Attack.

Table 4.3 shows the detection rate of the first 30 images using the pair comparison method. The number of bytes per secret image that were attacked is 12,800, which is a quarter of the actual attacks on the stego images, as each 2 bits of a secret byte is embedded in a different byte. The few cases where the pair comparison did not detect the attacks, were detected by the checksum comparison method, but in this case locations of the altered bytes were not available.

Detected secret Image **# Detected stego Detection rate** bytes attacks bytes attacks 1.pgm 38412 12748 99.593 38211 12755 99.648 2.pgm 3.pgm 38275 12754 99.640 12754 38365 99.640 4.pgm 5.pgm 99.695 38461 12761 6.pgm 38328 12752 99.625 7.pgm 38308 12749 99.601 8.pgm 38516 12759 99.679 9.pgm 99.609 38463 12750 10.pgm 99.640 38206 12754 11.pgm 38439 12747 99.585 99.671 12.pgm 38364 12758 99.609 13.pgm 38456 12750 14.pgm 38303 12750 99.609 15.pgm 12755 99.648 38363

Table 4.3: Detection Rate Using Pair Comparison for 30 Images (12800

Secret Bytes Altered).

16.pgm	38262	12747	99.585
17.pgm	38441	12752	99.625
18.pgm	38357	12750	99.609
19.pgm	38420	12752	99.625
20.pgm	38441	12756	99.656
21.pgm	38413	12752	99.625
22.pgm	38487	12753	99.632
23.pgm	38320	12751	99.617
24.pgm	38291	12733	99.476
25.pgm	38266	12737	99.507
26.pgm	38319	12743	99.554
27.pgm	38380	12751	99.617
28.pgm	38320	12751	99.617
29.pgm	38475	12753	99.632
30.pgm	38501	12740	99.531

The Extract-Verify program provides a list of location of the altered bytes and contents of these bytes using pair comparison. Table 4.4 shows a sample from list of location and content of altered bytes for the first image in the dataset (1.pgm).

Byte Location	Byte Content
31977	233
31978	74
31979	63
31980	147
31981	223
31982	230
31983	131
31984	126
31985	100
31986	119
31987	75

 Table 4.4: List of Location and Content of Altered Bytes Using Pair

0	•
Comn	aricon
Comp	ai 15011.
31988	179
-------	-----
31989	195
31990	46
31991	23
31992	192
31993	158
31994	18
31995	177
31996	35
31997	86
31998	7
31999	86
32000	176
32001	65
32002	23

32003	139
32004	115
32005	108
32006	209
32007	14
32008	153
32009	244
32010	163
32011	150
32012	42
32013	134
32014	115
32015	14
32016	169
32017	136

32018	250
32019	133
32020	209
32021	226
32022	197
32023	175
32024	52
32025	124
32026	80

The altered bytes list can help the user in investigating the pattern of attacks especially if they happen frequently.

57

Chapter Five

Conclusion and Future Work

5.1 Conclusion

The work in this thesis presented a security enhancement scheme to protect secret data that are embedded in cover images using the steganography method. The protection scheme has two objectives:

a. Detect alterations to the stego image that is carrying the secret data, and identify locations and content of the bytes of the secret data that have been altered.

b. Provide a distortion or camouflage of the secret data if the adversary managed to uncover the hidden secret data, by adding extra data as a decoy.

The presented scheme was implemented in MATLAB 2015a, and an experimental work was carried out in which 1000 grayscale images were embedded with a secret data image, attacked to alter contents of the stego images, and the secret data was extracted and verified for any change. The verification process involved the proposed data-decoy pair comparison, as well as checksum comparison.

The obtained results demonstrated that a detection rate of alterations using the datadecoy pair comparison was 99.6%, very close to the 100% outcome of the checksum comparison method, but with the advantage over the checksum method of knowing which part of the secret document was changed, which can help the user to understand the nature of the attack and whether the secret data can be recovered.

The PSNR evaluation results showed that adding a decoy data of 2 bits per byte has caused a drop in PSNR value from an average of 43.8% to 33.8%, in case of inverted decoy. Despite this drop in the PSNR value as a result of adding the decoy data, the stego

image quality is still within the accepted PSNR imperceptibility criteria. However, it is possible to achieve the higher PSNR value by embedding the data and decoy pairs in alternate bytes using the 2LSB method.

5.2 Future Work

Based on the present work, the following suggestions for future work are presented:

- Applying the proposed model to multi-channel images such as RGB (24 bits) and RGBA (32 bits).
- 2. Adapting the proposed model to deal with text changes on secret text documents, where the changes are for multi-characters.
- Investigating alternative embedding techniques, in combination with adding the decoy data.
- 4. Investigating the recovery of altered data using replicated decoy data.

References

Al-Bayati, M. (2016). The hiding of multimedia secret files in dual RGB cover images using LSB steganography techniques, (Unpublished master dissertation), Middle East University, Amman, Jordan.

AL-Shatnawi, A. M., & AlFawwaz, B. M. (2013). An Integrated Image Steganography System with Improved Image Quality. Applied Mathematical Sciences, 7(71), 3545-3553.

Almohammad, A. (2010). Steganography-Based Secret and Reliable Communications: Improving Steganographic Capacity and Imperceptibility. Brunel University.

Bahirat R. & Kolhe A. (2014). Overview of secure data transmission using steganography.

International Journal of Emerging Technology and Advanced Engineering. Volume (4), issue (3), ISSN: 2250-2459

Bashardoost M., Sulong G. & Gerami P. (2013). Enhanced LSB image steganography method by using knight tour algorithm, Vigenere encryption and LZW compression. *IJCSI International Journal of Computer Science Issues*, Vol. 10, Issue 2, No 1.

BOSSbase Dataset, (http://dde.binghamton.edu/dounloaded/), retrieved on 14/4/2017.

Channalli, S., & Jadhav A. (2009). Steganography an Art of Hiding Data. International Journal on Computer Science and Engineering. Vol.1 (3): 137-141

Chavda R., DoshiA A., & DeulkarA K. (2014). Assessing Image Steganography Techniques. International Journal of Current Engineering and Technology, E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Devi, K. J. (2013). A sesure image steganography using LSB technique and pseudo random encoding technique, (Unpublished master dissertation), National Institute of

Technology, Rourkela.

Goel, S., Rana, A., & Kaur, M. (2013) A Review of Comparison Techniques of Image Steganography. Global Journal of Computer Science and Technology Graphics & Vision, Volume 13 Issue 4 Version 1.0.

Goel, S., Rana, A., & Kaur, M. (2013). Comparison of image steganography techniques. International Journal of Computers and Distributed Systems, 3(1), 20-30.

Hamid N., Yahya A., Ahmad R. B., & Al-Qershi O. M. (2012).Image Steganography Techniques: An Overview. International Journal of Computer Science and Security (IJCSS), Volume (6), Issue (3).

Hamid N., Yahya A., Ahmad R. B., & Al-Qershi O. M. (2012). AN IMPROVED ROBUST AND SECURED IMAGE STEGANOGRAPHIC SCHEME. International Journal of Electronics and Communication Engineering & Technology (IJECET), ISSN 0976 – 6464(Print), ISSN 0976 – 6472(Online) Volume 3, Issue 2, July-September (2012), © IAEME

Kaur S., Kaur A. & Singh K. (2014). A survey of image steganography. *International Journal of Review in Electronics & Communication Engineering (IJRECE)* Volume 2 -Issue 3 p-ISSN 2321-3159

Kekre H.B., Athawale A.A., & Patki S.A. (2011). Steganalysis of LSB Embedded Images Using Gray Level CoOccurrence Matrix. International Journal of Image Processing (IJIP), Volume (5), Issue (1).

Ker A. D. (2005). Steganalysis of LSB Matching in Grayscale Images. **IEEE SIGNAL PROCESSING LETTERS**, Volume (12), NO (6).

Kour J., & Verma D. (2014). Steganography Techniques – A Review Paper. International

Journal of Emerging Research in Management & Technology. Volume (3), Issue (5).

Mishra, M., Mishra, P., & Adhikary, M. C. (2014). Digital image data hiding techniques: A Comparative Study. *arXiv preprint arXiv:1408.3564*.

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

Nissar, A., & Mir, A. H. (2010). Classification of steganalysis techniques: A study. *Digital Signal Processing*, *20*(6), 1758-1770.

 $Qasem,\,M.\,(2014). \ \textbf{Hiding secret image within RGB images using an enhanced LSB}$

method, (Unpublished master dissertation), Middle East University, Amman, Jordan.

Salih, M. M., & Al-Jarrah, M. (2015). Secret Message Integrity of Audio Steganography Using Bi-LSB Embedding. International Journal of Computer Science and Network Security (IJCSNS), 15(7), 20.

Sarayreh, G. S. (2014). Text Hiding in RGBA Images Using the Alpha Channel and the

Indicator Method, (Unpublished master dissertation), Middle East University, Amman, Jordan.

Shelke F., Dongre A., & Soni P. (2014).Comparison of Different Techniques for Steganography in Images, Volume 3, Issue 2.

Singh, S., & Siddiqui, T. J. (2012). A security enhanced robust steganography algorithm for data hiding. *International Journal of Computer Science Issues (IJCSI)*, 9 (1). pp. 131-139.

Singh S., & Kaur J. (2015). Steganography in True Color Images Using Even Odd Bit Slicing. International Journal of Engineering and Computer Science. Volume 4 Issue 5 SIPI Dataset, retrieved on 14/4/1017. Sivathanu, G., Wright, C. P., & Zadok, E. (2005). Ensuring Data Integrity in Storage: Techniques and Applications. A report submitted to Stony Brook University.

Sumathi C.P., Santanam T. & Umamaheswari G. (2013). A study of various steganographic techniques used for information hiding. *International Journal of Computer Science & Engineering Survey (IJCSES)*, Vol.4, No.6

Swain G., & Lenka S. (2014). Classification of Image Steganography Techniques in Spatial Domain: A Study. International Journal of Computer Science & Engineering Technology (IJCSET).

Thakur, J., & Kumar, N. (2011). DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis. International journal of emerging technology and advanced engineering, I(2), 6-12.

Vaman,P,. Manjunath, C.R. Sandeep.K (2013).Integration of Steganography and Visual Cryptography for Authenticity. International Journal of Emerging Technology and Advanced Engineering. Volume 3, Issue 6.

Yalman, Y., & ERTÜRK, İ. (2013). A new color image quality measure based on YUV transformation and PSNR for human vision system. Turkish Journal of Electrical Engineering & Computer Sciences, 21(2), 603-612.

Appendix A

Sample of BOSSbase1.01 Dataset

Images







