University of Diyala - College of Science

Computer Science Department

Digital Image Steganography System

This research was presented to the Council of the College of Science - University of Diyala - Department of Computer Science as part of the requirements to get a bachelor's degree in Computer science

By:

Hassan Haithem Mohammed Ali

Ibrahim Ahmed Mahmood

Ruqaya Hassan Ibraheem

Under The Supervision of:

Dr. Bashar Talib Al-Nuaimi

1441A.H. 2020 A.D.

ABSTRUCT

Steganography is the art of science that is concerned with hiding communications by hiding secret information inside a medium as a carrier, called cover medium, to be sent over communication channels to meant parties. Information hiding could be done with simple and direct methods; however we should increase hidden information security as possible by developing and using more robust ways.

One of the most used techniques is Lest Significant Bit (LSB) Substitution. The direct approach that uses LSB Substitution is Sequential LSB that embeds data sequentially, but it is too simple and easy to attack. So, to increase hidden information security, we must use this technique of hiding in random way. Many algorithms were set to increase the security of the hidden data, each of which has its own mechanism of data hiding randomization.

This research introduces a relatively new algorithm of data hiding, called Indicator-based LSB, which embeds data depend on indicator to increase the hidden data security. The algorithm implements using indicators, which makes embedding operation moves forward through cover mediums during hiding process. By using random number that known by the sender and receiver.

There are several types of cover mediums as images, audios, videos, etc. However, image based steganography is the most common system used, since digital images are widely used over the Internet, so images were used through our research as the cover mediums for experiments. According to the tests and results, the randomness of the algorithm is extremely satisfied, so it is hard to attack the resulting stego files. Also the embedding operation of the algorithm results in stego files with high quality, so it doesn’t arouse any suspicion .

Keywords- Steganography, Information Hiding, Information Security, Steganalysis, Randomness.

الاهداء

إلى النور الذي ينير لي درب النجاح (أبي)

و يا من علمتني الصمود مهما تبدلت الظروف (أمي)

إلى من يضيئون لي الطريق ويساندوني ويتنازلون عن حقوقهم لإرضائي والعيش في هناء (أخوتي)

إلى من أنار لي الطريق وأمسك لي مشعل النور أستاذي الفاضل

د. بشار طالب وجميع اساتذة قسم علوم الحاسبات ،

وإلى كل من أضاء بعلمه عقل غيره ،

أو هدى بالجواب الصحيح حيره سائليه

فأظهر بسماحته تواضع العلماء

وبرحابته سماحه العارفين.

شكر وتقدير

اشكر الله العلي القدير الذي أنعم عليَّ بنعمة العقل والدين. القائل في محكم التنزيل "وَفَوْقَ كُلِّ ذِي عِلْمٍ عَلِيمٌ" سورة يوسف آية 76.... صدق الله العظيم .

وقال رسول الله (صلي الله عليه وسلم):"من صنع إليكم معروفاً فكافئوه, فإن لم تجدوا ما تكافئونه به فادعوا له حتى تروا أنكم كافأتموه" ..... (رواه أبو داوود) .

فبعد شـــكر المــولى عز وجل ، المتـفضل بجليل النعم ، وعـظـيم الجزاء.. يجدر بي أن أتـقـدم بـبالغ الامتـنـان ، وجزيــل العرفان إلى كل من وجهني ، وعلمني ، وأخذ بيدي في سبيل إنجاز هذا البحث .. وأخص بذلك مشرفي **الدكتور بشار طالب** الذي قوم ، وتابع ، وصوب ، بحسن إرشاده لي في كل مراحل البحث ، والذي وجدت في توجيهاته حرص المعلم ، التي تـؤتي ثمارها الطيــبة بإذن الله ...

كما أحمل الشكر والعرفان إلى كل من أمـدني بالعـلم ، والمعرفـة ، وأسدى ليَّ النصح ، والتوجيه ، وإلى ذلك الصرح العلمي الشامخ متمثلاً في جامعة ديالى ، وأخص بالذكر كلية العلوم ، قسم الحاسبات، والقائمين عليها ,كما أتوجه بالشكر إلى كل من ساندني بدعواته الصادقة ، أو تمنياته المخلصة

أشكرهم جميعاً وأتمنى من الله عز وجل أن يجعل ذلك في موازين حسناتهم.

Table of Contents

List of Figure

CHPTER 1
INTRODUCTION

1.1 Introduction

Due to the need for transmitting secret data, steganography plays an important role in secure communications, since steganography is concerned with hiding communications. Communication hiding is accomplished by hiding secret data inside an innocent-looking medium, and sending the medium over a communication channel to the meant party. Consequently, steganography is used for protecting secret information while it is being transferred, as in military issues, if some institution needs to transfer data and protect it from spying or even in data transferring between individuals. In this chapter we give an introduction to steganography field and a brief overview about its applications and usage. We also explain all the aspects of this research and how it is organized.

1.1 Background and Context

Steganography is the art of science that is concerned with hiding secret data inside other innocent-looking data, which is called the cover, carrier or container, in order to hide communications, so no one apart from the meant parties can suspect the existence of the secret data and thus, the covert communication taking place (Johnson & Jajodia, 1998; Krenn, 2004). The aim of steganography is hiding the very existence of the secret data, in order to hide the communication taking place. Therefore, we can transfer the carrier medium with the hidden data inside over some channel to the meant recipient while no one apart knows or can suspect that there is data transferring and communication in between. So, if we have some sensitive data that should not be exposed to unauthorized parties, and we need to transfer it over an open network as the Internet, simply we can hide the data into some medium and send the carrier medium with the hidden data to the meant recipient. The object into which the data is embedded and hidden is known as cover medium (Kipper, 2003), and the resulting output known as stego-medium (Kipper, 2003), or stegogramme (Bateman & Schaathun, 2008). The stego-medium should be as identical to the cover medium as possible, so while it is being transferred, it doesn’t raise any

suspicion. So, if anyone intercepts the stegogramme, it is difficult to tell that it has hidden data inside. One of the most used hiding techniques is Least Significant Bit Substitution (LSB), which depends on substituting the cover file binary sequences LSBs with secret data bits. Cover mediums could be of several types such as images, audios and videos, etc. (Neeta, Snehal, & Jacobs, 2006; Nguyen, Arch-Int, & Arch-Int, 2016). Through our work we introduce a new algorithm of data hiding using LSB substitution technique with high security and extra capacity compared to Hide & Seek or sequential LSB algorithm. We concentrate on image files as cover mediums for experiments.

1.2 Statement of the Problem

Many algorithms were set to hide secret data into cover data. LSB Substitution is a very popular technique which is used by too many algorithms. Some of those algorithms are straightforward or simple as Hide & Seek algorithm, and some are robust. So, it is necessary to develop with more security and robustness.

1.3 Objective

The main objective is to develop a approach of hiding secret data with high security and extra capacity compared to other LSB-based algorithms.

1.4 Scope

Our research is about developing a new algorithm of information hiding inside cover mediums. It concentrates on image files as cover mediums. LSB Substitution technique is used by the algorithm for embedding secret data into cover mediums. Since data is embedded into spatial domain, then lossless images as PNG and BMP could be used. Hence, we selected one of these types only for the experiments which is BMP. The algorithm was evaluated by evaluating the resulting files after embedding the data inside. Since images are used as cover mediums for experiments, so images evaluation means were used for evaluating the outcome of the algorithm. Also the resulting images by the embedding operation are subjected to a Steganalysis tool to measure how much the resulting mediums can withstand

the attacks. Steganalysis is out of the scope, but some of its techniques are used for experiments and testing purpose. Dataset images were gathered from USC-SIPI image database, which contains the famous images globally used for steganography algorithms evaluation such as Pepper, and randomly from Internet.

1.5 Signification

One of the most important issues for people and organizations is communicating securely. Most governments monitor communications means between people, organizations and even between other governments. Eavesdroppers spend too much effort for spying on some parties. Hence, communications may not be safe from monitoring or attacking. Therefore, both of these issues have increased the importance of finding secret communication methods. Steganography is considered one of the most fields satisfying the purpose, since its main aim is covert communication.

1.6 Limitations

Fortunately, there were no serious limitations encountered though the research, however :

- The algorithm can’t be applied to all types of images the same way. For lossless images as PNG and BMP we can embed the data directly into pixels. But it needs extra work to apply it to lossy images as JPEG, because we do the embedding through the transform domain.
- Also for audio mediums, we need to know how deal with signals to make them contain the secret data using LSB Substitution.
- The size of the data that can be embedded is restricted by the number of cover medium size, as an instance, for lossless images the more pixels we have, the more data can be embedded.
- On another hand, access to most studies and researches is limited due to monetary constraints, since most researches and studies require to be paid for in order to get them.

CHPTER 2
LITERATURE REVIEW

The idea of steganography is not new. It has been used long time ago for transmitting data. However, with the evolving of digital communication means, it has become possible to employ steganography for transferring secret data covertly through digital communication channels. Through this chapter we give in details explanations for all aspects related to steganography.

2.1 Ancient History

The term steganography is of Greek origin, Steganos means "covered" and graphia means “writing” (Gowda & Sulakhe, 2016; Holub, 2014). Then steganography which is the combination of them means “Covered Writing”(Sharif, Mollaeefar, & Nazari, 2016; Watkins, 2001). It has been used in several forms for thousands of years (Cheddad, Condell, Curran, & Mc Kevitt, 2010). In the 5th century BC Histaiacus shaved a slave’s head and tattooed a message on his skull to get the message hidden after the slave’s hair grew back. Then he dispatched the slave with the message (Bateman & Schaathun, 2008; Easttom II, 2016). Five hundred years ago, the Italian mathematician Jerome Cardan reinvented a Chinese ancient method of secret writing, which depends on using a paper mask with holes. The method was named Cardan Grille after him. Nazis invented several steganographic methods during World War II and have reused invisible ink and null ciphers (Cheddad et al., 2010). Today after the extreme development of information technology field, steganography became widely used in digital fields and its techniques evolve more and more day by day. Steganography can be used for multiple purposes, like watermarking, ownership identification and copyright protection, data authentication etc.

2.2 Steganography Nomenclature

Steganography refers to the process of hiding data within some cover medium to allow covert communication. When we need to send some secret data to some remote recipient over an open network that can be accessed by anyone, like Internet,

and the data shouldn't be exposed to unauthorized parties, then seriously the data need to be sent covertly where no one knows that there is data transferring. This covert communication is the main purpose of steganography, where we need to keep others from thinking that a secret message even exists within stego files. So, steganography aims to hide the communication by hiding the presence of the data passing. The strength of this advantage is that no one knows about the data, so it enables us to make the data avoid even the attempt of attack.

Hiding data is accomplished by embedding it into some medium which called the cover medium, such that the resulting object would be sent carrying the data to the meant party through the communication channel. The object into which the data is hidden is called the cover medium, and the resulting object is called the stego medium or stegogramme. The cover medium could be one of several types. It could be an image, audio, video, text, html or any other object. The resulting stegogramme after hiding data must be created with some restrictions to be of high quality, such that no one can suspect that it contains secret data. It should be identical to the cover medium as possible. Stego mediums with differences from the cover mediums may arouse suspicion and attract attacker's attention. So stego mediums should have visual and statistical properties as close as possible to the cover medium properties. As stegogrammes are sent to the meant recipient, they may get subjected to attacks to find out whether they carry hidden data or not. The art concerned with detecting stegogrammes and steganographic message is known as steganalysis (Kipper, 2003). Lots of steganographic algorithms have been developed to result in stegogrammes with high quality in order to make it as difficult as possible to detect them by steganalysis means. So the main purpose of steganalysis is detecting the stegogrammes. Since steganography is concerned with allowing secret communication, not just hiding data into files. It also takes advantage of network protocols, such as TCP and SOAP, by hiding the secret data inside them or their headers, since network headers contain many fields that are either optional or unused for normal transmission. (Al-Mohammad, 2010; Lubacz, Mazurczyk, &

Szczypiorski, 2012). These protocols such as ARP, TCP, UDP or ICMP protocols, are referred to as carrier-protocols (Lubacz et al., 2012).

2.3 Steganography and Cryptography

Steganography is a branch of information security field, since one of the information security concerns is protecting sensitive and secret data, which is the purpose for which steganography is used. Another branch of information security concerned with data protection is cryptography. So steganography and cryptography are cousins and intended to protect information from unauthorized parties, but the main difference between them is how each of which does so (Goel, 2008). Steganography is concerned with hiding data and hiding the very existence of the data, where cryptography is concerned with scrambling the data and make it unreadable "cipher", so the encrypted data does not make sense to anyone but the meant parties after they decrypt it (Bateman & Schaathun, 2008; Dunbar, 2002; Johnson & Jajodia, 1998; Krenn, 2004; Morkel, Eloff, & Olivier, 2005). Encrypted data could be vulnerable because eavesdroppers are aware of its existence (Jain & Boaddh, 2016). And since attackers have the chance to apply cryptanalysis techniques over the data, then it is possible to break down the security system (AlMohammad, 2010). So sometimes hiding the communication is more important than protecting it. This was clearly illustrated by Simmons (1983) in the "Prisoners' Problem" (Simmons, 1984). In Prisoners Problem, Alice and Bob are arrested and thrown in two different cells. They want to make an escape plan, but their communication is monitored and checked by a warden (Wendy). Alice and Bob must communicate invisibly in order not to arouse Wendy's suspicion since she will transfer them to a high-security prison if she notices any suspicious communication. Alice and Bob can succeed only if they can transfer messages covertly without making Wendy suspicious. Thus, this vulnerability can be solved by hiding the message transferring from Wendy (Chandramouli, Kharrazi, & Memon, 2003). So, the communication could be hidden by hiding the data messages within an innocentlooking cover medium that does not

arouse suspicion of eavesdroppers (AlMohammad, 2010). So, Steganography which is a kind of covert communication is concerned with not detecting the existence of secret the data because it aims at making it unknown that there is secret message passing (Stanley, 2005), where cryptography is concerned with not understanding the secret data by altering its structure (Thangadurai & Sudha Devi, 2014; Wang & Wang, 2004). Even though both cryptographic and steganographic systems provide secret communications, they are different in terms of system breaking. A cryptographic system is considered broken if an attacker can read the secret message. However, a steganographic system is considered broken if merely an attacker detects the existence or read the hidden message (Al-Mohammad, 2010). We can combine both of them to make our data more secure. While we can encrypt data by one of cryptography techniques, next we can hide it within a carrier using a steganography technique to provide extra layer of protection which is advisable by most researchers (Al-Mohammad, 2010; Bateman & Schaathun, 2008; Goel, 2008; Johnson & Jajodia, 1998; Mahmood, Azeez, & Rasool, 2014; Satar et al.). Therefore steganography role is to complement cryptography (AlAni, Zaidan, Zaidan, & Alanazi, 2010; Al-Mohammad, 2010). Cryptography can be divided into two types, symmetric and asymmetric (Mahmood et al., 2014). In symmetric cryptography, encryption and decryption is done using the same key. Asymmetric encryption involves pair of keys, public and private. When encryption is done using one key, decryption is done by the other (Mahmood et al., 2014 .(

2.4 Steganography Architecture

A steganographic system is comprised of two algorithms, the first is for hiding and the second is for retrieving. The hiding process is concerned with embedding data within the cover medium and resulting in the stegogramme. Therefore, this process should be constructed carefully to be sure the stegogramme is identical to the cover medium as possible; thus the message is sent unnoticed. Therefore, basically the components of the embedding process system consists of a secret message and a

cover medium as inputs, a steganography algorithm as the method of hiding and a resulting stegogramme as the output. Also a secret key can be used for hiding the data as a third input to increase the robustness and security of the hidden data, such that there is no way the data is retrieved in the absence of the secret key even though the algorithm of hiding is known (Al-Ani et al., 2010; Al-Mohammad, 2010; Goel, 2008). On the other hand, the retrieving process is concerned with extracting data from the stegogramme. Simply this process is the inverse of the hiding process. Retrieving process takes the stegogramme and the secret key as inputs, and returns back the secret data as the output (Al-Ani et al., 2010; Al-Mohammad, 2010; Goel, 2008). Figure 2.1 shows the architecture of a steganographic system.

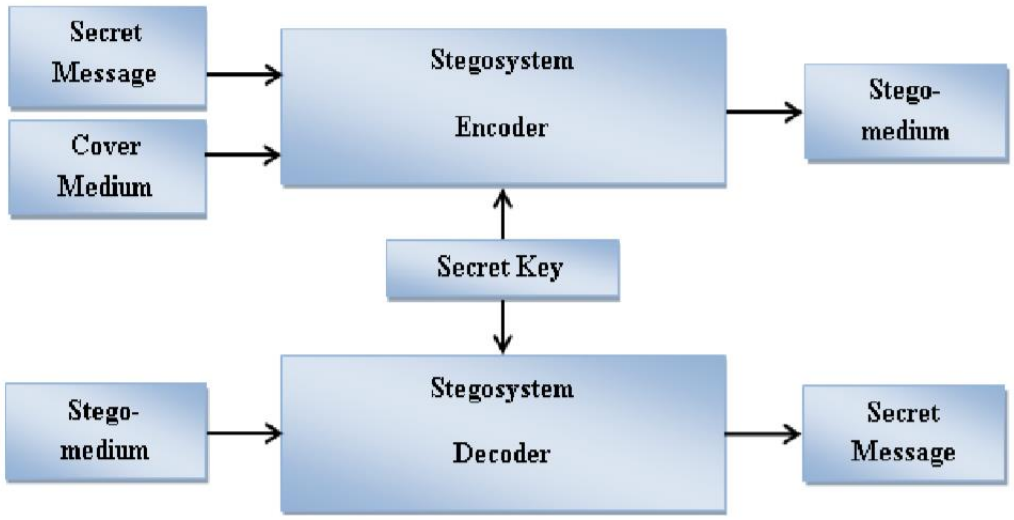


Figure (2.1): Steganography System Architecture

When we desire to hide secret data into some cover medium, the stegosystem should be designed carefully to embed the data and create a stegogramme which is an exact copy of the cover medium, or at least as close as possible, so that the adversary regards the stegogramme and the communication taking place as innocuous. After obtaining the stegogramme, it is in most cases sent to a remote recipient along with the secret key to extract the hidden message (Al-Ani et al., 2010; Al-Mohammad, 2010; Goel, 2008) .

2.5 Steganography in Depth

Over time since steganography has been started being used, it has evolved and many new algorithms and techniques were developed to improve the hiding operation and increase the hidden data security. The embedding process is done by altering the contents and tweaking the values of cover mediums to make them contain the data and result in the stegogrammes. However, we can't modify the values of all areas of the cover file. Changing values of some parts of the cover file may destroy the cover file or result in some noticeable and detectable distortion. Thus, if the distortion was perceptible, the chances of detecting the stegogramme would be so high. So, the lower the distortion, the better the chances of undetectability. Therefore essentially, steganographic systems must identify the redundant data of the cover medium. Redundant data is insignificant data that when gets modified, it has no direct impact on the overall perceptibility of the cover file, therefore the alteration of the data is not detected easily (Al-Mohammad, 2010; Morkel et al., 2005). So, any modification to these redundant bits should not destroy the integrity of the cover medium, and thus preserving the quality which in turn would enhance the imperceptibility and the undetectability of the steganographic system and resulting stegogrammes. Moreover, even if the hiding algorithm used is publicly known, if the stegogramme has no suspicious changes or indications, no one can figure out the presence of hidden data. So, steganographic systems should produce stegogrammes as identical to the cover medium as possible, such that it doesn't arouse suspicion and it makes it hard for steganalysts to detect steganography in stego mediums that is identical to innocent mediums (Bateman & Schaathun, 2008).

Hidden data security is enhanced by enhancing the imperceptibility or the robustness. Also, some algorithms increase the capacity of cover mediums for secret data. So the key properties of steganographic systems that must be considered are imperceptibility, robustness and capacity (Bateman & Schaathun, 2008; Saidi, Hermassi, Rhouma, & Belghith, 2016; Sumathi, Santanam, & Umamaheswari,

2014). So, we can consider them as criteria of efficiency of steganographic algorithms and systems:

1. **Imperceptibility or Undetectability:** Imperceptibility is how much the stego file has no perceptually detectable change or distortion. Thus, it depends on the quality of the resulting stego file to be as identical to the cover object as possible. This is done by avoiding making noticeable change in the resulting stego medium. (Bateman & Schaathun, 2008; Krenn, 2004; Morkel et al., 2005; Sumathi et al., 2014). Additionally, the stego-medium must not be statistically perceptual, thus it should has statistics identical to the cover medium (AlMohammad, 2010).

2. **Robustness:** Robustness is the degree of how much the steganographic system can withstand against steganalysis and attacks, and how difficult to determine whether the stegogramme contains hidden data or not (Bateman & Schaathun, 2008; Sumathi et al., 2014; Wang & Wang, 2004). Robustness involves withstanding hidden data detection, extraction and destruction by steganalysis means.

3. **Capacity or Payload:** Capacity is determined by the maximum amount of secret data that can undetectably be embedded inside the cover medium. Hiding data within cover files could be done sometimes with huge amount, but it would be so obvious that the resulting stego files have hidden data inside. So, increasing the capacity of an algorithm must be done with maintaining the quality of the cover files, and with least possible affecting to its properties (Al-Mohammad, 2010; Kipper, 2003).

However, there is tradeoff between imperceptibility and capacity, where embedding more data introduces more artifacts into cover mediums and then

increases the perceptibility of hidden data (Al-Mohammad, 2010). Subsequently, data embedding should be as small as possible, since typically the more the embedded data, the more the cover medium is altered, the easier for steganalysts to detect the stegogramme (Bateman & Schaathun, 2008; Kipper, 2003; Sumathi et al., 2014). So it is difficult to increase the capacity and maintain the imperceptibility at the same time (Al-Mohammad, 2010).

2.6 Steganography Classifications

Several approaches were set for classifying steganographic systems, but there are two general approaches. The first is based on the type of cover file while the second is based on the hiding method used (Al-Mohammad, 2010).

2.6.1 Cover Type-Based Classification

Since we can hide the data inside multiple types of cover mediums, Thus steganography could be classified according to the cover medium type that is used for hiding the data within as :

1. Image steganography.
2. Audio steganography.
3. Video steganography.
4. Text steganography.
5. HTML steganography.
6. Network steganography.

The classification is shown below in Figure 2.2:

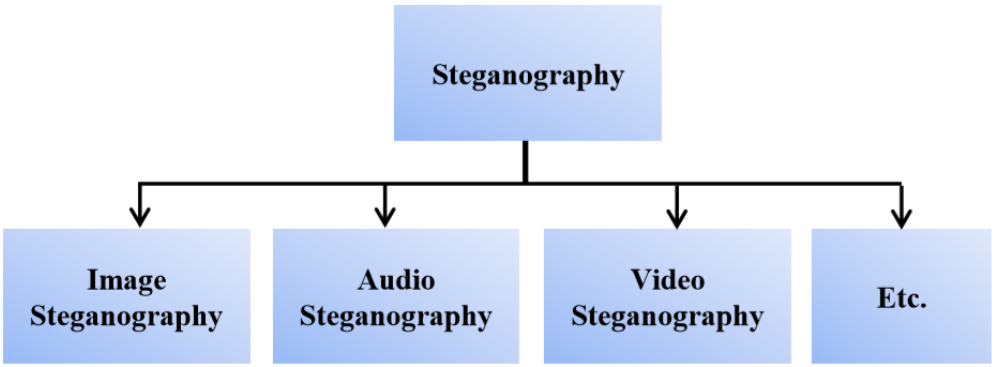


Figure (2.2): Cover Type-Based Classification

However, the properties of cover files types vary, thus the hiding processes themself vary in accordance with the type of the cover medium.

2.6.2 Hiding Method-Based Classification

Steganography could be also classified according to the method of data hiding. Subsequently, steganography can be split into three approaches of hiding data (AlMohammad, 2010):

1. Insertion-based method.
2. Substitution-based method.
3. Generation-based method .

The classification is shown below in Figure 2.3:

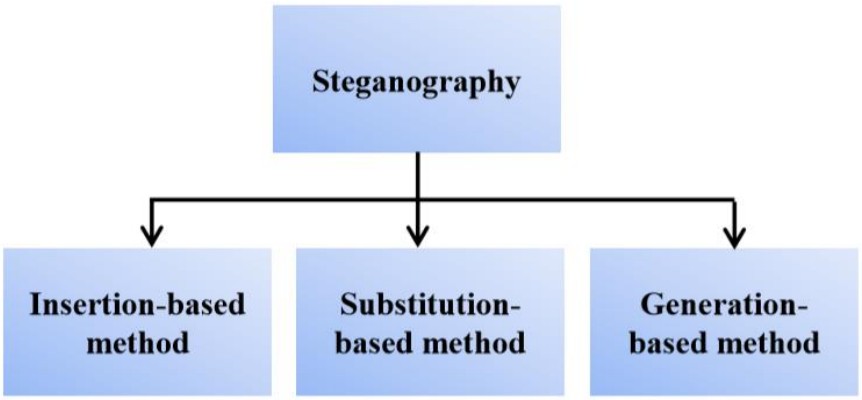


Figure (2.3): Hiding Method-Based Classification

2.6.2.1 Insertion-based method

Insertion-based method works by finding areas in cover files which are ignored by applications that read these cover files, and hiding the secret data within these areas. This method involves a defect and an advantage. The defect is, since this method inserts the data inside the cover file, then the file of the resulting stegogramme would be larger than the cover medium size. However, since in most cases the original cover file would not be available for comparing, this method may be good as long the stego medium size is reasonable. The advantage is that it doesn't change the content of the cover file, so it preserves the quality of the cover file and there wouldn't be any detectable or perceptible change in the stegogramme. Also by using this method, we can hide any amount of data inside the cover medium, however with adding too much data, the resulting stegogramme will be very suspicious.

As an example, some files have a flag called EOF or end-of-file marker. This flag is used by the applications to find the end of a file in order to stop processing it. One of the ways to hide data is just to insert it after the EOF marker of the cover file and the application will ignore the hidden data when reading the resulting stego-file (Cheddad et al., 2010; Eric, 2003). As an instance, if an image is used as a cover file, simply the message is inserted after the EOF tag of the image file and when opening the stego-image by any photo application, it will just display the image ignoring anything coming after the EOF marker. On the other hand, as mentioned the weakness is that the size of the resulting stegogramme is the sum of the sizes of both the secret data and the cover file, which may arouse the suspicion if the size is too large to be of the cover file (Cheddad et al., 2010; Eric, 2003). Another example is writing the secret data between end-text and begin-text markers in Microsoft Word files (after the end-text and before the next begin-text). And according to the configuration of Microsoft Word, it ignores anything written in such areas, so the

hidden message would not appear when the document is read by Microsoft Word application (Eric, 2003).

2.6.2.2 Substitution-based method

Substitution-based method depends on finding insignificant areas or information in cover files and replacing these areas values with the secret data. So the sizes of both the cover file and the stegogramme are identical, since the cover data are just modified without any data adding, and this is the main advantage of this method over insertion-based method. On the other hand, the quality of the cover file could be degraded because of the modification. And the amount of secret data that can be embedded is restricted by the size of the insignificant information that can be replaced or overwritten (Eric, 2003).

2.6.2.3 Generation-based method

Generation-based method works by generating the cover file into which the data would be hidden into. So it doesn't require an existing cover file. The main advantage of this method is that the stegogramme is not a cover file with distortion or extra size than the original cover file. However, the generated files might be unrealistic to end users ,since they consist of random content. So, probably randomlooking images is suitable for this kind of information hiding (Eric, 2003).

2.7 Least Significant Bit Substitution

One of the earliest and most popular steganography techniques is Least Significant Bit Substitution technique (LSB) (Easttom II, 2016; Juneja & Sandhu, 2013). In Computer science, the term Least Significant Bit refers to the smallest (right-most) bit of a binary sequence (Bateman & Schaathun, 2008). LSB substitution technique is defined as hiding the secret data bit into the LSB of the cover binary sequence, by replacing the LSB value of the cover binary sequence

with secret bit value, regardless of the order of the cover binary sequences used to contain the secret data, and if the hiding order is sequential or random. So if we have a cover binary sequence of 01101100 and a secret bit with value of 1, and we need to hide it using LSB substitution technique, then simply we replace the LSB of the cover binary sequence with the secret bit and the cover binary sequence becomes 01101101. The simplest algorithm that hides data using LSB substitution is the Hide & Seek algorithm, which embeds secret bits into the LSBs of the cover binary sequences in sequential fashion from the beginning to the end of the cover medium. When we want to embed a byte of secret data, we just take the eight bits of the secret byte, and replace the least bit of a series of eight binary sequences of the cover data with these secret bits and so on. Simply we replace the least bit of each binary sequence of the cover binary sequences with our secret bits. Thus, every secret byte consumes eight cover binary sequences to get embedded into (Morkel et al., 2005).

Suppose a binary sequence unit of the cover data is one byte and we have the secret byte 01101010 which we want to embed into a series of cover bytes using LSB substitution technique. The process would be done as shown below in Table 2.1:

Table (2.1): Series of Cover Bytes before and After Embedding by Sequential LSB

Cover byte Index	The series of cover bytes before the embedding process	The resulting stego bytes after the embedding process
0	10111000	1011100(0)
1	10000001	1000000(1)
2	10100001	1010000[0]
3	01111100	0111110[1]
4	01110110	0111011(0)
5	10010000	1001000[1]
6	11010011	1101001(1)
7	01010010	0101001(0)

(b) The bit new value is identical to its original
[b] The bit new value is different from its original

There are several advantages of LSB substitution technique. First it doesn't affect the size of the cover data because it does not increase or decrease the number of the cover data bytes. It just replaces some of the cover data bits with our secret bits without affecting the size. Next, it does not make noticeable changes to the cover data and this is according to two factors. Firstly, the change occurs in the least bit (right-most) which has the least weight among all the bits in a byte. Simply if the value of this bit is changed from 0 to 1 or from 1 to 0, in either cases the value of the byte is changed only by 1 increasingly or decreasingly as shown in Table 2.1 at byte 3 and 2 respectively. This amount of change is neither noticeable by human visual system nor human auditory system. Secondly some of the bits are replaced with its same value as shown in Table 2.1 above, for example, at byte 7 there is no change done to the byte. Thus, on average half of the cover bits are modified using the maximum cover size (Johnson & Jajodia, 1998; Morkel et al., 2005; Wang & Wang, 2004). So, if we hide some data into, for example an image, the change of the image is not detectable at all by human eye, even if we also used the second least significant bit for hiding (Morkel et al., 2005), where it has been claimed by (Ker, 2007) that it is better to use two bit planes than one. LSB from the viewpoint of capacity consumes moderate amount of cover bytes for embedding certain amount of secret data. Where for each secret byte to get embedded, it needs 8 bytes of the cover. So, to hide the data we just need to use a suitable-sized cover, and secret data could be compressed before embedding.

2.8 LSB Substitution and Image-Based Steganography

To a computer, an image is collection of numbers that constitute different light intensities of image areas, and the numeric values forms a grid of points referred to as pixels (Morkel et al., 2005). Image steganography is concerned with developing and enhancing techniques and algorithms of hiding data inside images. Image

steganography is the most common system used among steganography categories, because images are widespread over Internet and web (Al-Mohammad, 2010). Hiding data into images could be done either into spatial domain or into transform domain images (Morkel et al., 2005; Silman, 2001; Sumathi et al., 2014; Wang & Wang, 2004). Spatial domain techniques use lossless images as PNG and BMP, and transform domain techniques use lossy images as JPEG. Spatial domain techniques have more capacity than transform domain techniques, however transform domain techniques are more robust (Al-Mohammad, 2010).

2.8.1 LSB Substitution into Spatial Domain

Spatial domain is the image plane itself; the collection of pixels that composes an image (Kipper, 2003). In spatial domain techniques, data embedding is done by encoding the secret data bits directly into image pixels values (Goel, 2008; Jain & Boaddh, 2016). The image pixels are tweaked to contain the secret data bits. The most technique used in spatial domain is LSB substitution, since LSB substitution technique embeds the secret bits directly into the cover file. So, in image steganography, the secret data is being embedded directly into the LSBs of cover image pixels values. LSB substitution technique could be used for embedding secret data either in sequential or random fashion. The simplest form of spatial domain LSB substitution is the method known as Hide & Seek (Bateman & Schaathun, 2008). In Hide & Seek, the LSBs substitution is done sequentially, starting from the first binary sequence unit until the end of the cover file. However, it would be easy for a steganalyst to retrieve the secret data (Morkel et al., 2005). Therefore, many researchers developed plenty of algorithms that do the embedding in randomized manners, in which the locations that contain the data are scattered and not sequential (Bateman & Schaathun, 2008). For instance, the Hide & Seek method itself was applied in randomized mode by shuffling the image pixels using a Pseudo Random Number Generator (PRNG) according to a seed before embedding the secret data. Then the secret data is embedded within the shuffled image data using Hide & Seek

method. Finally, the image pixels is inversely shuffled back using the same seed to obtain the image in the original order, but with scattered hidden data inside (Bateman & Schaathun, 2008). Spatial domain techniques are applicable to lossless image compression as PNG, BMP and GIF (Goel, 2008).

2.8.2 LSB Substitution into Transform Domain

Transform domain techniques are methods with Lossy compression used for reducing image size, without reducing its quality to noticeable degree by naked eyes. It involves several methods as Discrete Cosine Transform (DCT), Fast Fourier Transform, etc. Images of transform domain like JPEG could be used for steganography, where data embedding is done through the transform process by encoding the data bits into transform domain coefficients (Bateman & Schaathun, 2008; Jain & Boaddh, 2016). Hiding in this fashion is more difficult to detect than spatial domain as steganalysts have to do more effort to find the embedding artifacts (Al-Ani et al., 2010; Bateman & Schaathun, 2008). LSB substitution could be used with JPEG, but with some difference from spatial domain. For instance, JSteg algorithm embeds the secret data inside the LSBs of the DCT coefficients, rather than pixel values as in spatial domain (Bateman & Schaathun, 2008). Also sequential hiding was not considered very secure. So OutGuess algorithm was developed to improve JSteg algorithm by randomizing the embedding process (Bateman & Schaathun, 2008). The randomizing is done in the same way of randomized Hide & Seek approach, where the coefficients are shuffled randomly using a PRNG according to a seed. The embedding within the shuffled coefficients is performed using the technique of JSteg. Finally the shuffle operation is inversed in order to get the coefficients back in the correct order.

2.9 Related Work

Many steganographic methods were set with various advantages and weaknesses. The study in (Sumathi et al., 2014) covers in details various

steganographic techniques and classifications. Too many LSB-based approaches were set for hiding data. The main purpose of these approaches is increasing security of hidden data. Security of LSB-based approaches is implemented mostly by hiding secret data with randomness and scattering it within the entire cover medium.

2.9.1 Image Steganography

Image steganography techniques are concerned with hiding data inside image files (Gowda & Sulakhe, 2016). Many researchers proposed LSB-based approaches for hiding data within images. (Karim, Rahman, & Hossain, 2011) proposed using a secret key to decide the positions within the cover image to hide the secret data into. Also for deciding the positions of hiding, beside the secret key they use the red channel of the cover image. Simply for each cover pixel, the LSB value of the red color is XORed with the current bit of the secret key. If the result of the XOR operation is 0, then the current secret bit is embedded into the LSB of the blue color of the current pixel. Otherwise it is embedded into the LSB of the green color and so on. So, it is difficult to retrieve the hidden data in absence of the secret key, which in turn increases the robustness of the stego-image.

The algorithm is robust due to hiding with randomness and secret stego-key, but on the other hand, it doesn’t add much randomness, since it moves among the pixels sequentially from the beginning to the final pixel and just hide either inside green or blue channel. For imperceptibility, it is so high due to embedding into one color per pixel, which means one byte out of three for RGB images, or one byte out of four for RGBA images whose LSB is altered (RGBA pixel is a pixel with alpha value. Alpha is used for pixel transparency to determine how opaque or how transparent a pixel is). However on the other hand, the capacity is so low, since we need 3 cover bytes to embed one bit, which means 24 cover bytes for each secret byte.

In research of (Akhtar, Johri, & Khan, 2013) Rivest Cipher 4 algorithm (RC4) is used with a stego-key to randomize the embedding of secret data over the entire cover image. They use RC4 algorithm with a stego-key to generate random order of the cover pixels locations. Then they hide the secret data into the pixels according to the random order. They also have introduced a new technique called bit-inversion to improve the stego image quality. The technique works by splitting the pixels into four sections according to the third and second bits values. The first section is of all the pixels that have the third and second bits with 00 values. The second section is of all pixels with third and second bits of 01 values, and so on, the third is of 10 values and the fourth is of 11 values. Finally, for each section, the count of changed and unchanged pixels is found, and if the number of changed pixels is greater than the unchanged pixels, then the LSBs of the section is inverted.

The robustness is increased due to adding the randomness, by using RC4 algorithm and a secret stego-key to generate random order for pixels locations. So the robustness is increased because to retrieve the hidden data, the sego-key must be known in order to find out the pixels locations order of the hiding process.. Imperceptibility was really improved by bit-inversion technique which reduces the number of changed pixels and ensures that in any case the changed pixels are less than 50% of the entire pixels. The capacity is as much as sequential LSB, so it is relatively good.

(Islam, Siddiqa, Uddin, Mandal, & Hossain, 2014) proposed an algorithm called filtering-based that uses LSB in different way than usual. The algorithm doesn’t embed within the LSBs the secret data bits, instead it embeds indications about whether a pixel contains hidden data or not. First they check what pixels are more, the lighter or the darker. Since the pixel consists of three colors, and each color is represented by one byte, then pixels are considered lighter when their colors MSBs have two or three bit values of 1. Thus, darker pixels have two or three MSBs of 0

value. The algorithm hides the secret data into the pixels with greater count among the darker and lighter. Also not all of the selected pixels are used to contain secret data, only those which match a condition that is part of the hiding process. The algorithm finds the decimal value Pn of the MSBs of the three colors of each pixel. The Pn value would be between 0 and 7. Inside the third byte of the pixel, if the value of the bit with index equal to Pn is equal to the secret bit, then LSB of the third byte is set equal to 1 as indication that this byte contains a secret bit, otherwise the LSB is set equal to 0.

The most advantage is that the secret data is not embedded into cover image, instead embeds the result of XOR operation between the cover image and the secret data. Therefore, for retrieving the hidden data, the cover image must be available, which make it hard for those who don't have the original cover object to extract the data, which in turn increases the robustness.

In the research of (Singh & Kaur, 2015), hiding into image process is done first inside odd pixels then even pixels. When hiding into a pixel, the algorithm embeds two bits into red byte LSBs, two bits into green byte LSBs and four bits into blue byte LSBs .

The algorithm is not very robust, because the hiding is not completely random. The hiding is done into odd pixels sequentially, then into even pixels in the same way. Also it exploits two LSBs of red and green colors and four bits of blue color, which in turn reduce the imperceptibility. The capacity is very high due to using eight bits per pixel.

2.10 Image Quality Metrics PSNR and MSE

Peak signal-to-noise ratio (PSNR) and mean square error (MSE) are the most common and widely-used metrics for image quality evaluation (Al-Mohammad,

2010). PSNR measures the similarity between two images (how two images are close to each other), while MSE measures the difference between two images (how different two images are from each other) (Al-Mohammad, 2010). Therefore, image quality is better with higher value of PSNR and smaller value of MSE. The best image quality is when MSE value is very small or going to be zero, since the difference between the original image and the reconstructed image is negligible (AlMohammad, 2010). For PSNR, the higher the PSNR value, the better the degree of imperceptibility, since the similarity between the original image and the reconstructed image is high (Al-Mohammad, 2010). However, PSNR values between 20 and 40 can be considered as typical values (Eric, 2003). For example, it is difficult to recognize any difference between a grey-scale cover image and its stego image if the PSNR value exceeds 36 dB (Al-Mohammad, 2010). PSNR and MSE are defined as follows (Al-Mohammad, 2010):

$$\mathrm{MSE} = \left(\frac{1}{\mathrm{MN}}\right) \sum_{i=1}^{M} \sum_{j=1}^{N} \left(X_{ij} - \overline{X}_{ij}\right)^2 \qquad (2.1)$$

$$\mathrm{PSNR} = 10.\log_{10} \frac{I^2}{\mathrm{MSE}} \mathrm{db} \qquad (2.2)$$

Where:

X_{ij} is the i^{th} row and the j^{th} column pixel in the original image,

$\overline{X}_{ij}$ is the i^{th} row and the j^{th} column pixel in the reconstructed image,

M and N are the height and the width of the image,

I is the dynamic range of pixel values, or the maximum value that a pixel can

take, for 8-bit images: I=255.

However, the MSE for color images is defined as follows (Al-Mohammad, 2010):

$$MSE_{AVG} = \frac{MSE_R + MSE_G + MSE_B}{3} \quad (2.3)$$

Where: R MSE, G MSE and B MSE are the MSE of red, green, and blue respectively.

2.11 Steganalysis Principles

Steganalysis is the art of identifying and detecting stegogrammes that contain hidden data (Bateman & Schaathun, 2008). So the main aim of steganalysis is merely detecting stego files, however beside stego-mediums detection, it involves data extraction and data destruction (Al-Mohammad, 2010). Whilst the aspiration of recovering the message should be considered extremely unlikely because to recover the hidden data, the steganalyst needs to discover the hiding algorithm, and the hidden data itself in most cases would have been encrypted before getting embedded. So, steganalysis succeeds and the steganography system is broken if merely an attacker detects the existence of the hidden data. Persons apply steganalysis with the intent of intercepting and detecting stego-mediums and the hidden data in the communication channels are referred to as steganalysts (Kipper, 2003). Generally, modifying some parts of a cover file to embed secret data inside, changes the properties of this file in some way, and this can be a sign of the presence of hidden data (Al-Mohammad, 2010). Therefore, applying a comparison between a stego file and its corresponding cover file may reveal the existence of the hidden message. Thus, to avoid such a comparison, cover files used for hiding data should not be publicly available, and after the embedding, they may have to be destroyed (AlMohammad, 2010). After embedding the secret data within a cover medium, the

resulting stegogramme is in most cases sent to a remote recipient over some communication channel. During the way it may get subjected to steganalysis. After detecting a stego-medium by steganalysis techniques, attacker may attack the communication in different forms. Steganography attacks could be categorized into three kinds in accordance with the role of the steganalyst as passive attack, active attack and malicious attack (Al-Mohammad, 2010; Bateman & Schaathun, 2008; Kipper, 2003):

1. **Passive Attack** is when the warden just observes the communication and permits or prevents the message delivery without performing any modification to the stego mediums. Therefore, the communication between two parties will be blocked in case the warden suspects that a secret communication is taking place.

2. **Active attack** is when the warden alters the detected stegogrammes and causes distortion to them during the communication, so that the communication is prevented. In such attack, the attacker may aims to alter the passing files, even though there is no suspicion, in order to destroy any hidden data might be existed.

3. **Malicious Attack** is when the warden replaces the hidden message with a fake message, and tries to impersonate one of the communicating parties to trick them. On the other hand, this type of attack is too hard to apply, because the attacker must know everything about the hiding process, like the algorithm used and the secret key if exists. Moreover, this attack is kind of easy to be detect by the receivers, since they would notice that the hidden messages don’t make sense.

2.11.1 Steganalysis Techniques

As defined before, steganalysis is the science concerned with detecting the covert communication taking place by detecting hidden data within stego mediums passing in between. Steganographic systems leave behind in stego files some traces as a result of embedding the secret message inside. These traces make the stego files detectable in some way. So, steganalysis focuses on taking advantage of these traces to detect the stego files.

2.11.1.1 Targeted Steganalysis

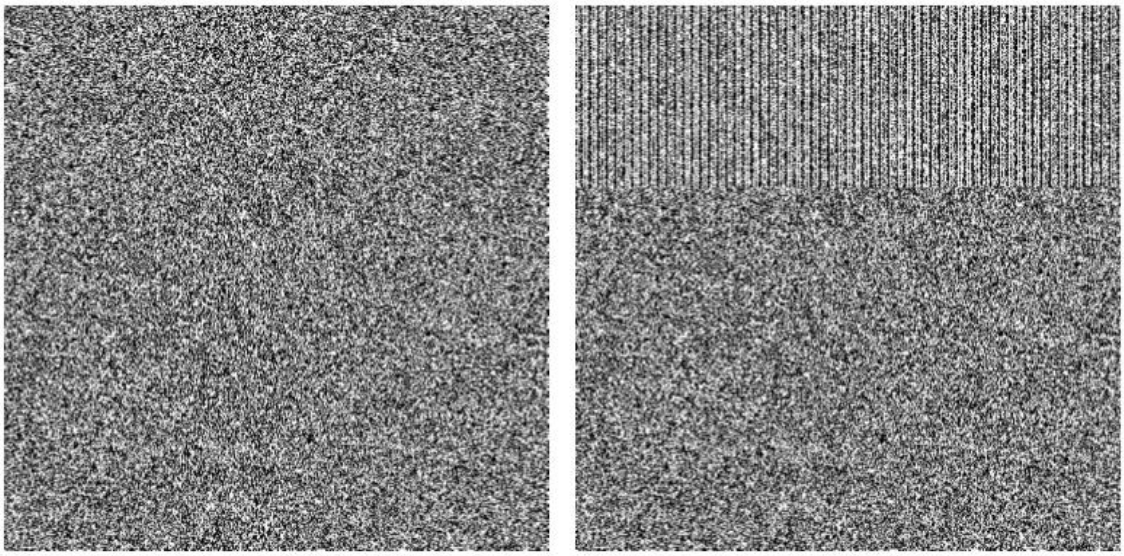
Targeted steganalysis techniques are designed in direct accordance with a specific methods of embedding, where they attempt to discover stego files by checking the known side-effects of specific steganographic algorithms, so it requires to have deep knowledge about the steganographic algorithm that the attack is targeted to (Bateman & Schaathun, 2008).

2.11.1.1.1 Visual Attacks

Visual Attacks are the process of examining the subject file or certain components of it by naked eye to identify any obvious inconsistencies with assistance of software (Bateman & Schaathun, 2008). Of course stego files with quality degradation as a result of steganographic manipulation look suspicious than the cover files, and could be detected by naked eye. So, the first rule to avoid visual attacks is that a steganographic system should keep quality of cover files as hiding data inside. When steganalysts perform visual attacks, they concentrate in isolation on the likely areas of embedding inside stego files to detect signs of manipulation.

The most common visual attack combats LSB-based steganography, and is made based on the fact that the structure of LSBs of a cover image does not match the structure of message bits (Bateman & Schaathun, 2008). The attack depends on viewing the LSB plane of the suspect image to identify any inconsistency that indicates existence of hidden data. For Images, there are almost even values as there

are odd, which means there are as many 0's as there 1's in its LSB plane (Bateman & Schaathun, 2008). When the text meant to be hidden, it is converted to binary or ASCII, the resulting bit stream would contain unequal numbers of 0's and 1's, where the number of 0,s is larger than the number of 1's (Bateman & Schaathun, 2008). Thus, replacing the LSB values with the ASCIIs of the text would increase 0's and in turn result in inconsistency in the LSB plane. So, the part of LSB plane that has hidden data would be visually different from the clean part. Steganalysts who perform visual attacks search for signs of embedding in the LSB plane by searching for such difference. Figure 2.4 shows an example of visual attack on a true color PNG image, where image (a) is clean and image (b) is the same image after being manipulated to contain secret data.

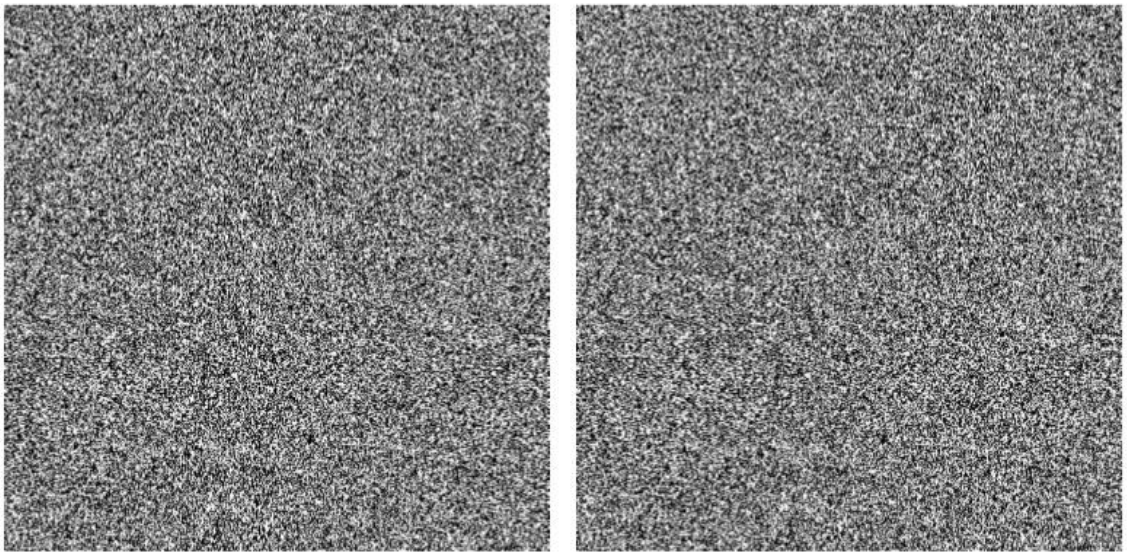


(a) Clean Image (b) Stego Image

Figure (2.4): The LSB Bit Plane Before and After Embedding Unencrypted Data

It is clear that image (b) has hard indication of embedding data within the first 33% of the LSB plane of the image. This kind of attack enables steganalyst to figure out the length of the hidden message. Also it is obvious that this attack doesn't need the original image to work, since the discrepancy is detected by checking the LSB plane of the suspect image. Visual attack fails when the secret data is encrypted before getting embedded inside the cover image. This is because the binary ASCII of the text has often 0's more than 1's, and by encrypting the secret data there will

usually be a more even distribution of 1's and 0's (Bateman & Schaathun, 2008). Figure 2.5 shows the same image of Figure 2.4, but image (b) here contains encrypted data.



(a) (b)

Figure (2.5): The LSB Bit Plane Before and After Embedding Encrypted Data

It is obvious that the image after embedding the secret data has no discrepancy, and this is due to hiding encrypted data. Also hiding the data randomly not sequentially results in no discrepancy within the image LSB plane. This will be shown later in the experiments and results chapter when our algorithm is used for hiding the data randomly. So, to defeat the visual attack, first the secret data must be encrypted before the embedding, next the embedding must be performed randomly and not sequentially (Bateman & Schaathun, 2008).

Another way of visual attacks is done if the steganalyst has access to the clean image. This is referred to as known cover attack. When the clean image is available, the steganalyst has the ability to get the LSB plane of both, the clean and the suspect image, and calculate the difference between them by subtracting one from the other to identify the identical and the different regions of the images .

Finally, the most trait that makes visual attacks too hard to perform is that visual attacks are not automated, and needs human to check each image for identifying the suspect images.

2.11.1.1.2 Statistical Attacks

These attacks depend on detecting the modifications which have occurred in the statistical properties of cover files (Al-Mohammad, 2010). Statistical Attacks may reveal that a file had been modified, but it can't identify which technique was used for modification. Statistical Attacks are often preferred because they can be automated (Bateman & Schaathun, 2008). For images, there are several statistical properties which can be analyzed such as standard deviation, differential values, median, skew and kurtosis (Al-Mohammad, 2010). There are several statistical attacks, and here are some of them:

Chi-square Test (Westfeld & Pfitzmann, 1999) enables steganalysts to compare the statistical properties of a suspect image with the theoretically expected statistical properties of its counterpart, where the degree of similarity of them is a measure of the probability of embedding. The test is based on statistical analysis of Pairs of Values (PoVs) that are exchanged during message embedding (Bateman & Schaathun, 2008; Westfeld & Pfitzmann, 1999).

Histogram Attacks depend on histogram analysis to identify whether there is steganography or not (Bateman & Schaathun, 2008). For instance, the Difference Histogram Analysis (Zhang & Ping, 2003) is a statistical attack on an image's histogram, measuring the correlation between the LSB and all other bit planes .

RS Analysis (Fridrich, Goljan, & Du, 2001) can detect 24-bit color images and 8-bit grayscale with randomly scattered LSB embedding by inspecting the differences in the number of regular and singular groups for the LSB and shifted LSB plane.

Sample Pair Analysis (Dumitrescu, Wu, & Wang, 2003) is a steganalysis

technique for LSB substitution depending on a finite state machine that can detect randomly embedded messages in LSB Plane.

Primary Sets (Dumitrescu, Wu, & Memon, 2002) is a steganalysis technique that can detect the randomly embedded messages in LSBs of natural continuous-tone images.

2.11.1.2 Blind Attacks

These attacks attempt to evaluate the probability of embedding based solely on the data of the suspect image, even when it is not known how the data might has been embedded. It assumes that nothing is known about either the algorithm or the cover image (Bateman & Schaathun, 2008). Some of the most popular blind attacks are Wavelet Moment Analysis (WAM), Calibration Based Attacks and Farid's Wavelet Based Attack (Goel, 2008).

2.11.2 StegExpose – Steganalysis Tool for Detecting Steganography in Images

StegExpose is a steganalysis tool specialized in detecting steganography of LSB substitution in lossless images such as PNG and BMP (Boehm, 2014). StegExpose can be run in the background analyzing multiple images without human supervision, returning a detailed steganalytical report once the tool has finished its job. StegExpose is derived from an intelligent and thoroughly tested combination of pre-existing LSB steganalysis methods which are Chi-square Attack (Westfeld & Pfitzmann, 1999), RS Analysis (Fridrich et al., 2001), Primary Sets (Dumitrescu et al., 2002), Sample Pairs Attack (Dumitrescu et al., 2003) and Difference Histogram analysis (Zhang & Ping, 2003).

2.12 Summary

In this chapter we have introduced the reader to the main issues of steganography. We have given an introduction and identified the core concepts and principles of this field. We have illustrated steganography and its aspects in depth. Also we have covered some other subjects concerning steganography. We have explained image steganography and its categories. Furthermore, we have discussed some of the most related work. Finally we have covered Steganalysis and its techniques, and talked about one of its tools which is going to be used for testing.

CHPTER 3
SYSTEM IMPLEMENTATIONS

3.1 Development Environment

The project was built and developed by Matlab using graphical user interfaces (GUI). Libraries and image processing functions have been used to complete the task of the project.

3.2 Embedding Process

In our approach of concealing data, we hide the secret image bits into the least significant bit (right-most bits), but this process is done depending on indicators. Images are chosen as cover mediums. Before the embedding process starts, the cover image will be converted to gray levels so it will consist of many bytes that consist of 8 bit. Each byte represents one pixel in the cover image. Therefore, data is hidden into one bit of each pixel. So, the cover image is treated as a stream of bytes through the hiding process.

Through the hiding process, an indicator is used to identify the number bit into which we embed the secret bit(s). Indicator value will be from 1 to 8 that represent the bits of the secret image that will embed in 1 to 8 bits of the cover image.

3.3 Retrieving Process

The retrieving process is simply the inverse of the embedding process. So as we depend on indicators through embedding process, we use the same indicators for retrieving. When we want to extract hidden data from the stego object, we check, the Location Indicator of each byte of the cover image.

Below, the entire project will be explained from the main interface and how to use and how to hide the image you want to hide into the second image and how to decode it to retrieve the original secret image that was hidden in the other image.

3.4 Design

The main interface of the program consists of a bar at the top where all the commands are used in encoding or decoding and in selecting the images to be applied to and down slightly there are a number of buttons that are similar to tasks of the top menu, and it consist of number of images that are displayed when you select image for processing and before or after processing. The Figure 3.1 illustrates the main interface of the project.

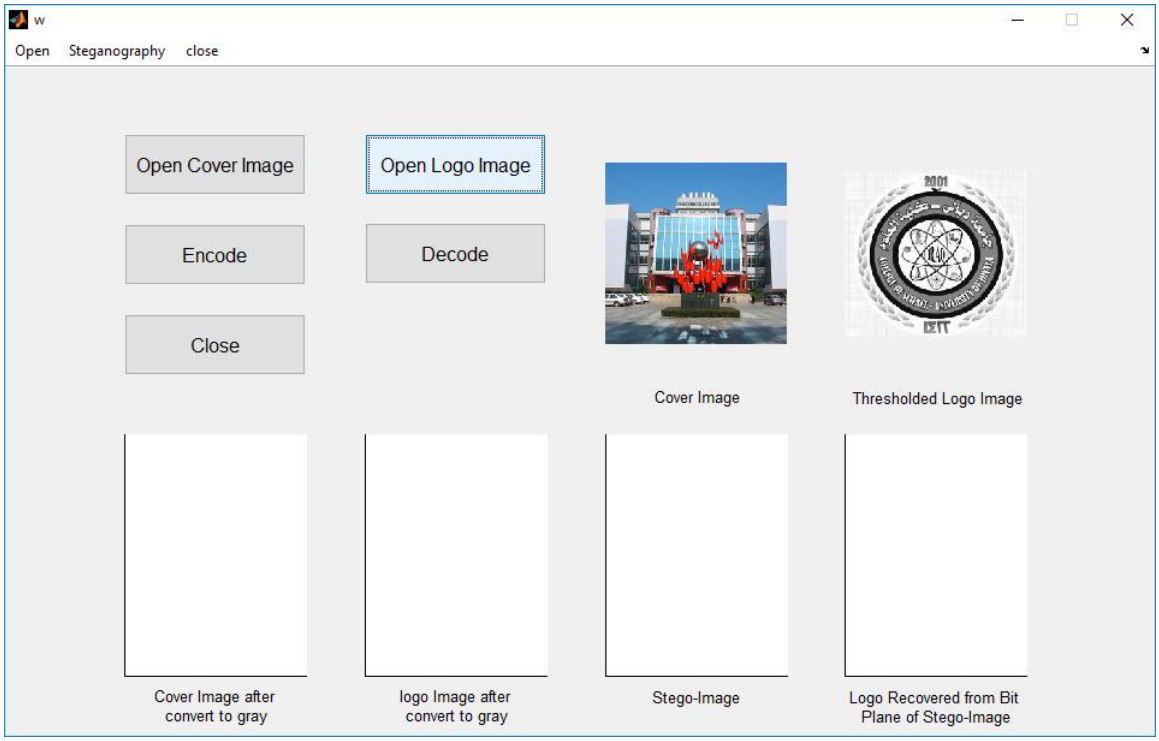


Figure 3.1 illustrates the main interface of the project.

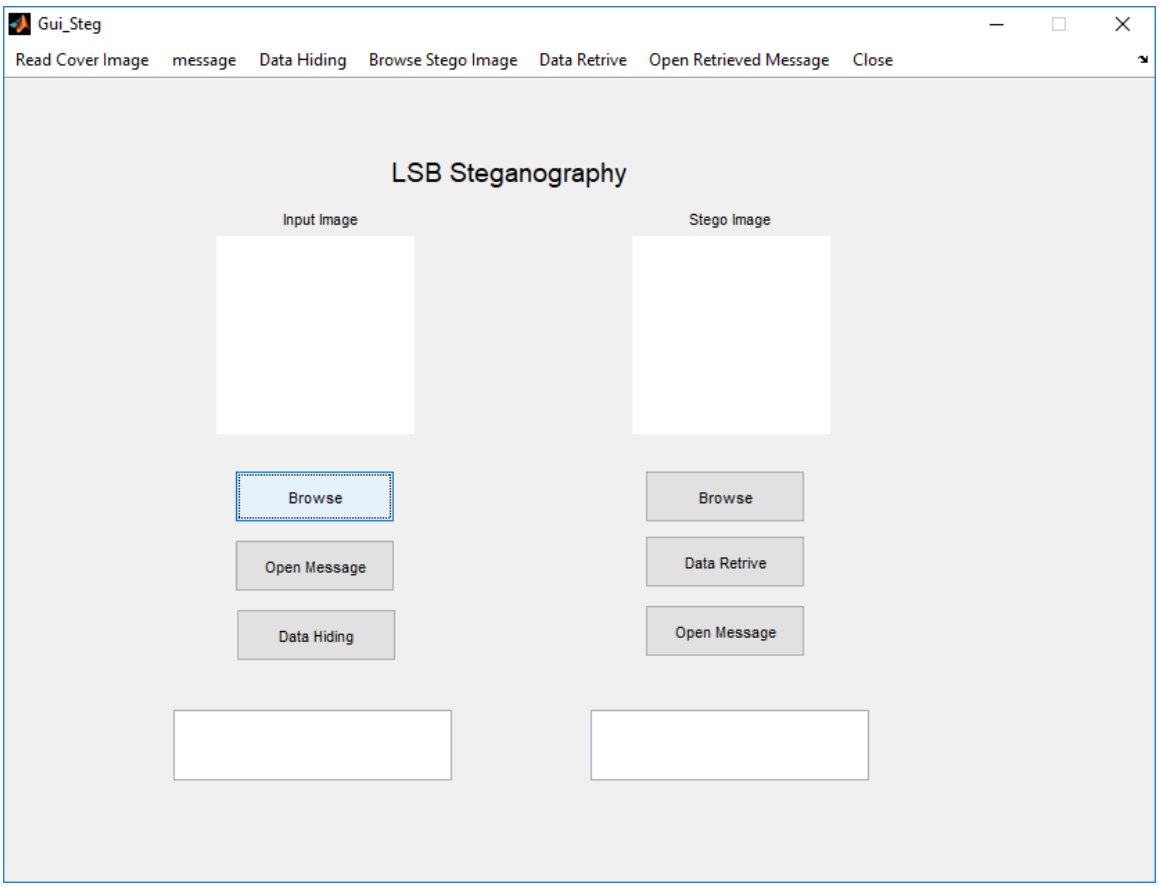


Figure 3.2 illustrates the main interface of the project

When you press the "Open Cover Image" button or from the Open menu at the top, choose the "Open Cover Image" also. A window will open to select an image for processing (same for open logo image button). Figure 3.2 illustrate when you click on the open cover image button.

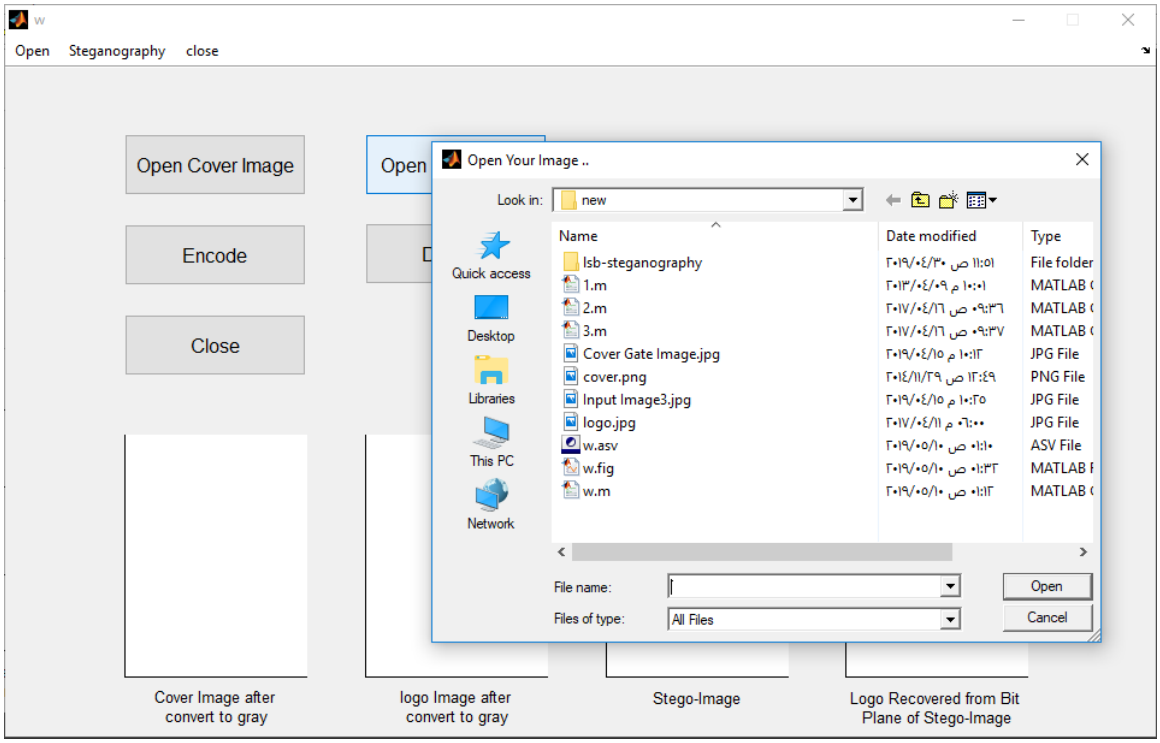


Figure 3.3 illustrate when you click on the open cover image button.

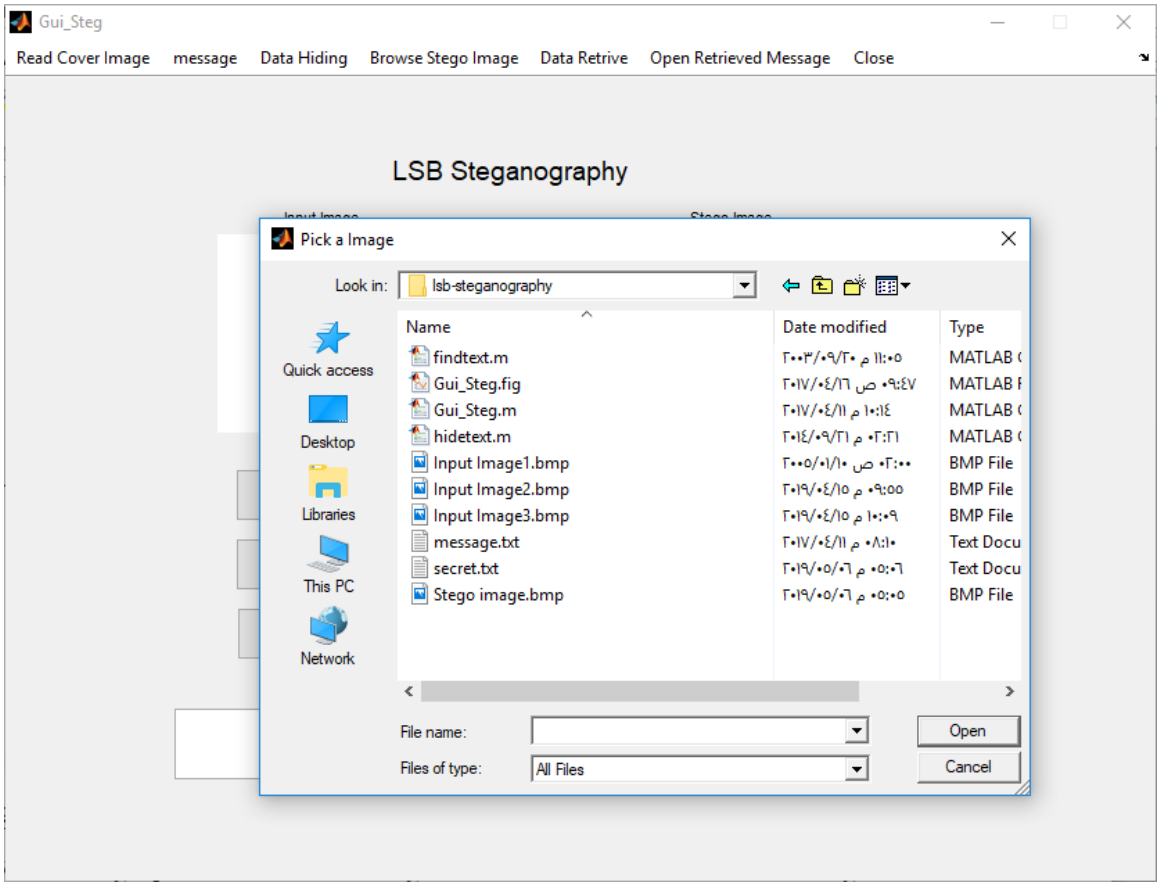


Figure 3.4 illustrate when you click on the open cover image button.

Figure 3.3 shows the case when you have chosen two images, one of which is the image that we want to hide and the second is the image that will hide the first image inside it and when you click the button encoding, cover image will converted to gray and the secret image will be convert to threshold image and then the prompt window will appears to request from the user to specifies where he wants to hide his or her own information, depending on the level of the bit.

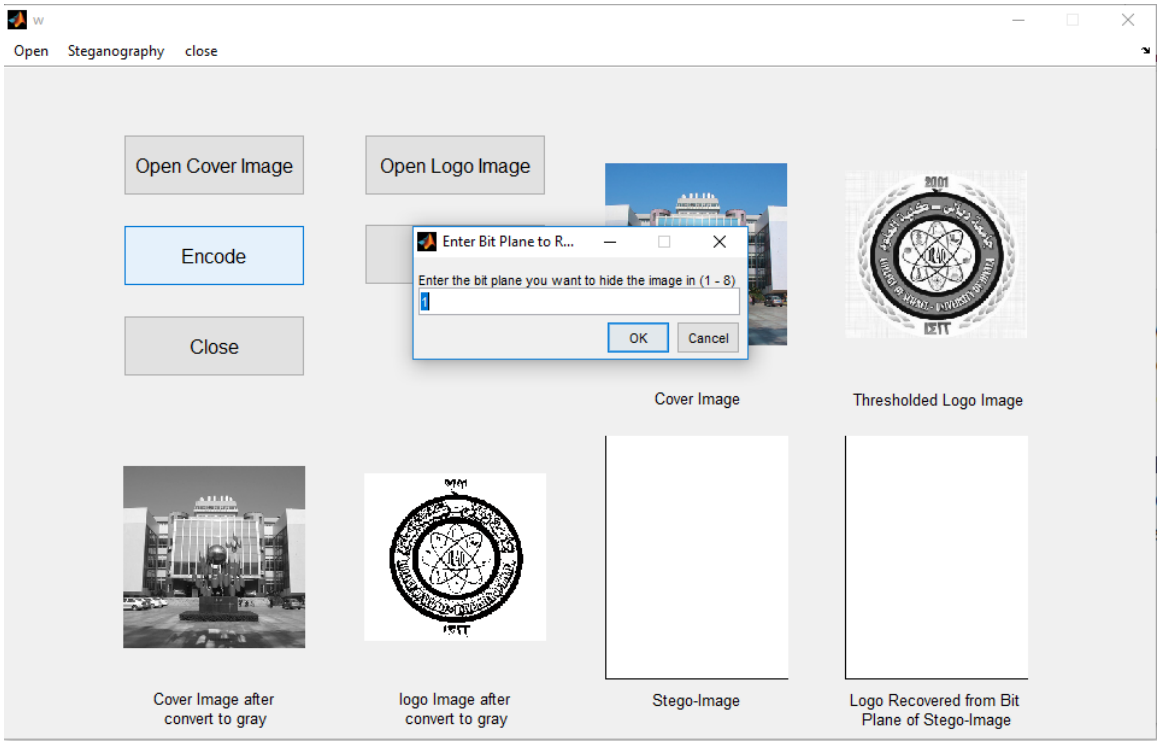


Figure 3.5: two images was selected and user must enter number of bit that he won't to hide the information inside it

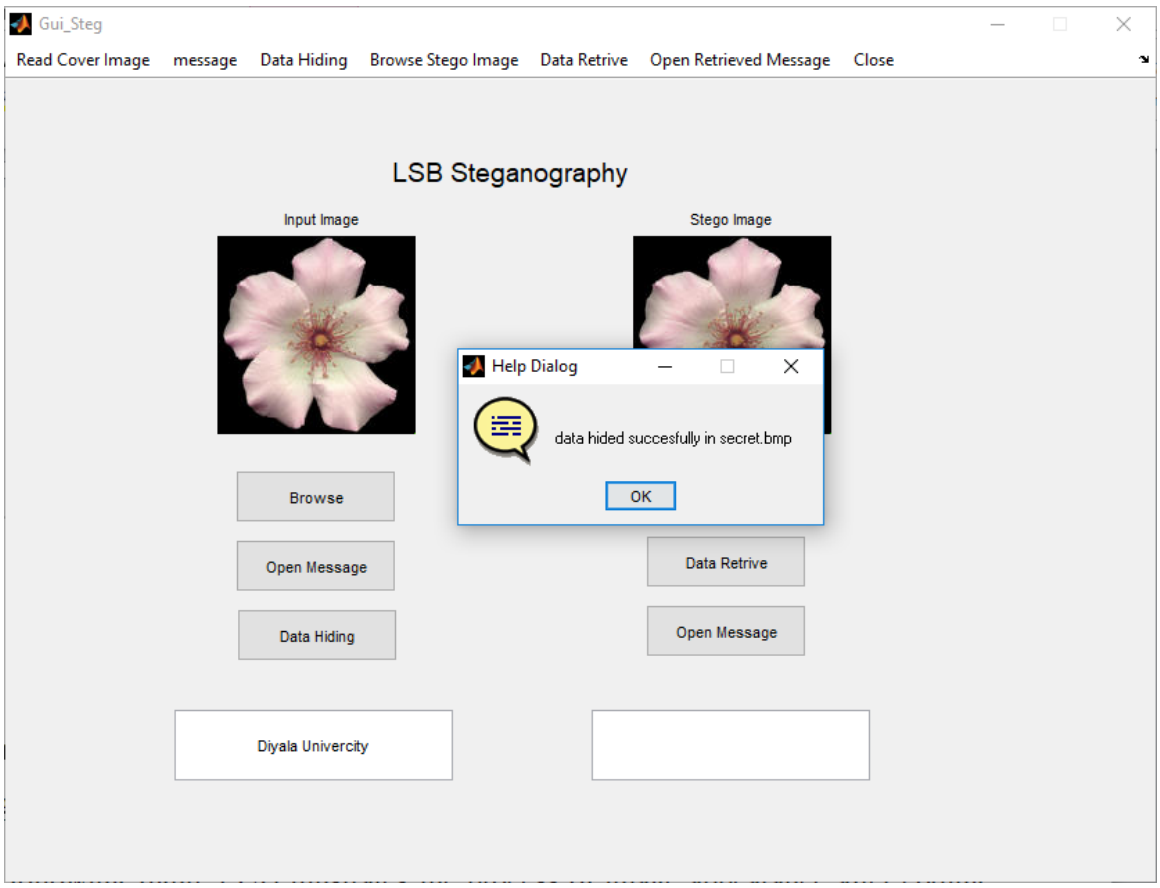


Figure 3.6 hideing text in image

when the number of bits is small, the image will be hide better , efficiently, the image will not fade into any form, and it will not be recognized.

The following figure (3.4) illustrates the process of image appearance after coding the second image inside. The coding process include taking the secret image which is in the form of 0, 1 (threshold image) and take each bit of it and placing this bits in the location specified by the user in the cover image (the place of this bit can be randomly generated and used on Form secret key for encryption and decryption between parties)

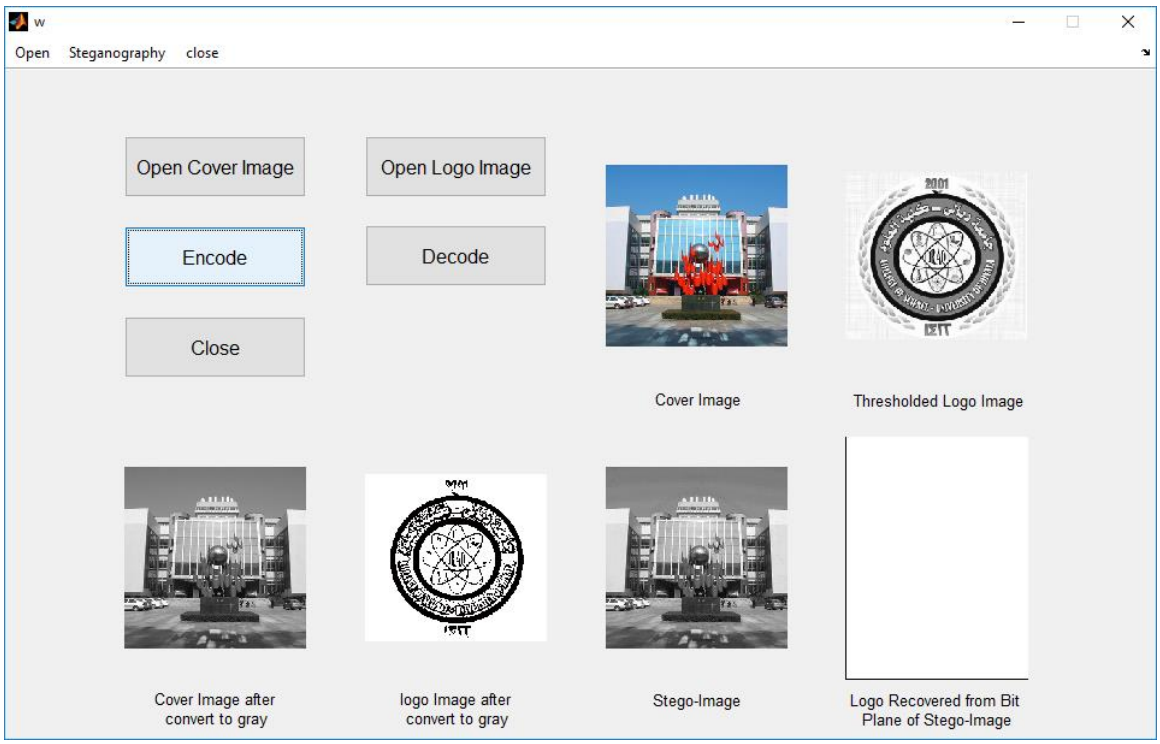


Figure 3.7: the cover image after the secret image encrypted inside it (stego-image)

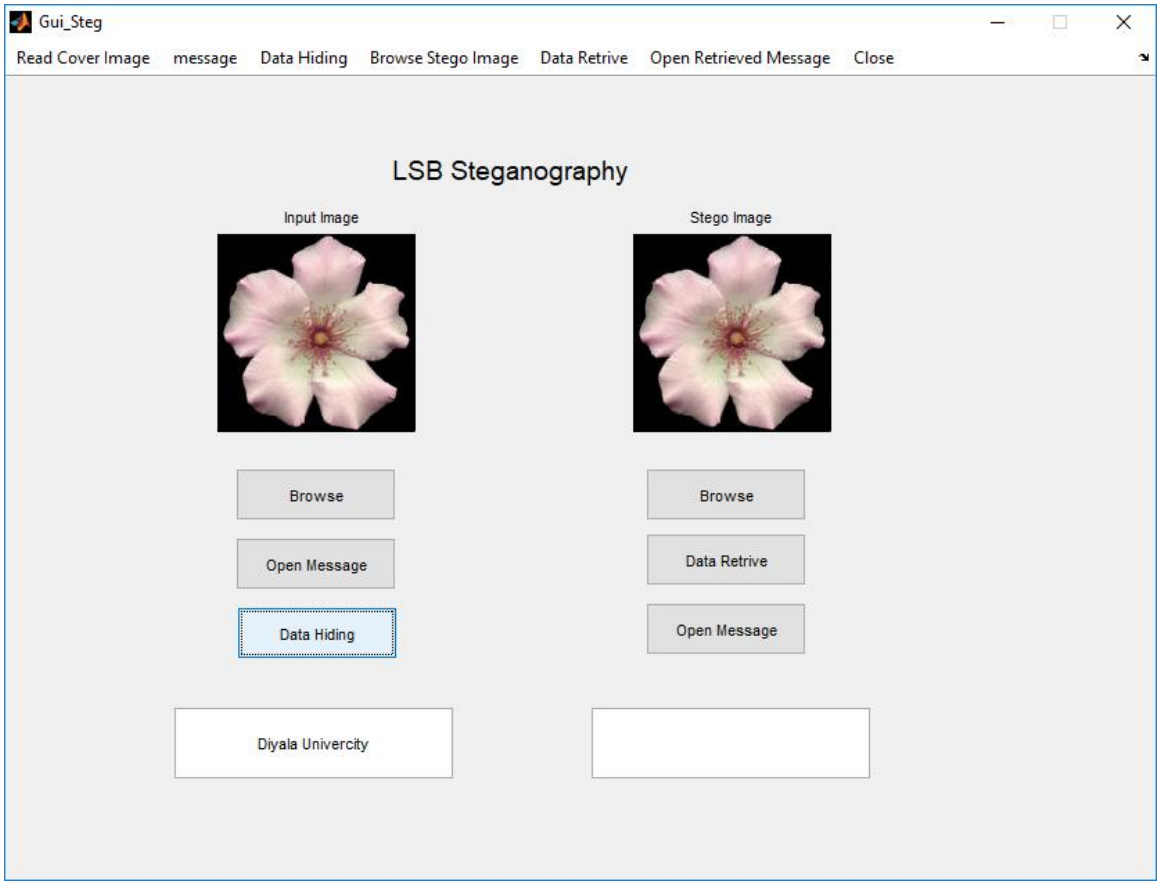


Figure 3.8: the cover image after the secret text encrypted inside it (stego-image)

Now the secret image has been encrypted in the cover image and once you press the decrypt button the algorithm will retrieve the original image (hidden image) that was hidden dependent on the indicator. The following figure (3.5) shows the process of retrieving the original image (secret image) after it was concealed in the image of the envelope by relying on the indicator

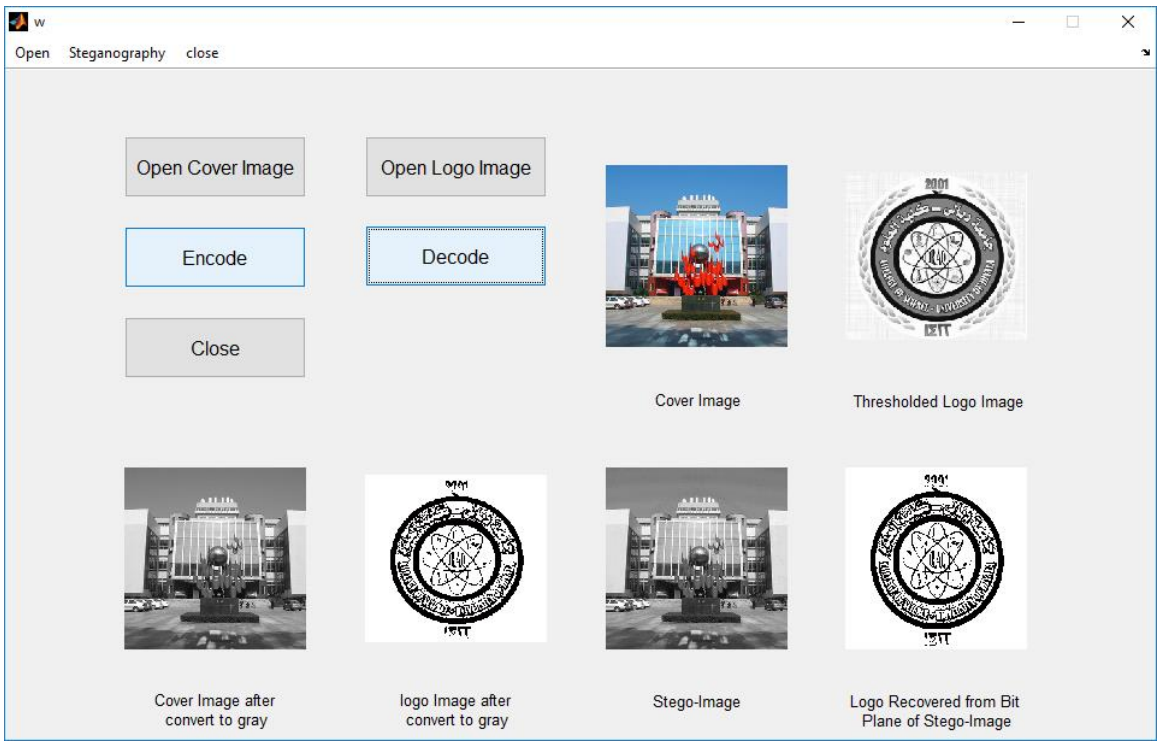


Figure 3.9 shows the retrieved the original image.

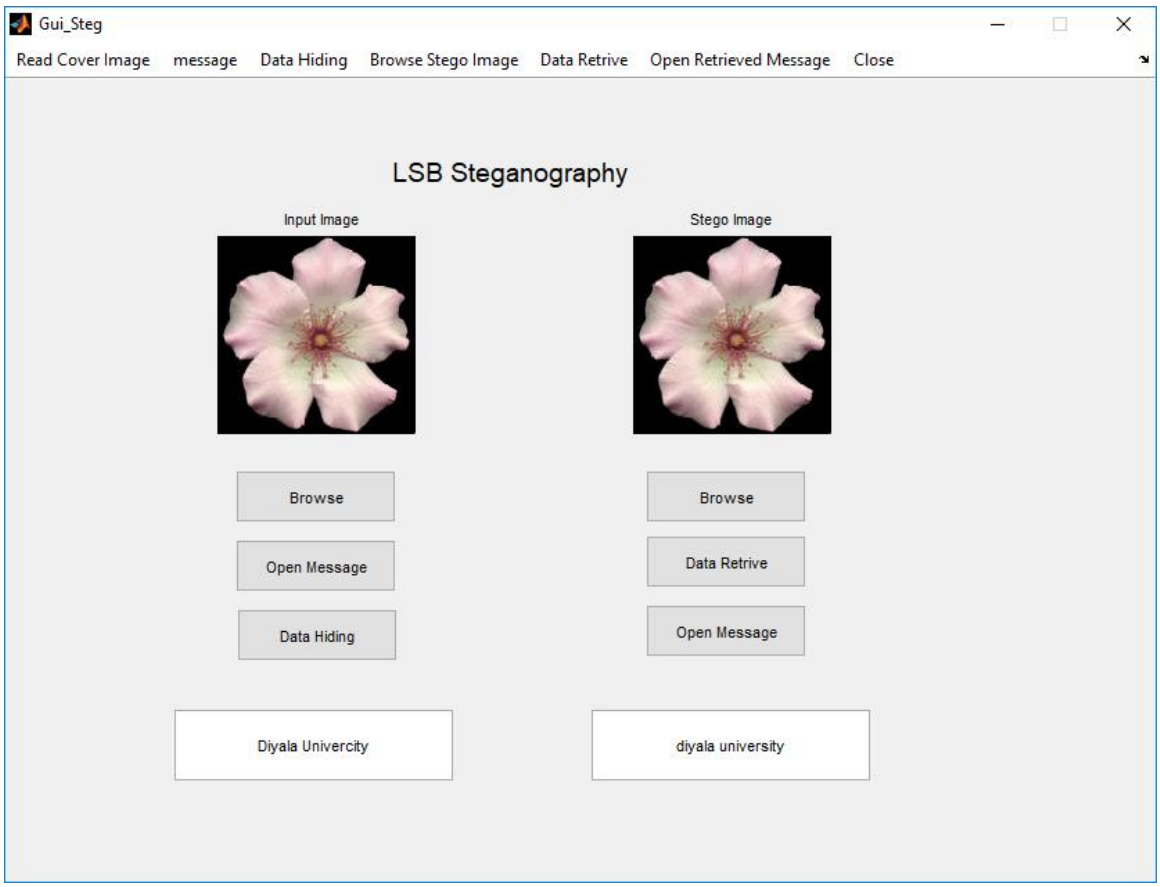


Figure 3.10 shows the retrieved the original text.

3.5 Indicator value

The value of the **Indicator** should always be between 1 and 4 so that the concealment process is efficient and effective at the same time and do not show any details of the secret image. The following figure (3.6) shows the difference between whether the value of the indicator is small or large.

(b) indicator = 3 (b) indicator = 6

Figure (3.6) shows the difference between whether the value of the indicator is small or large

CHPTER 4

CONCLUSION AND SUGGESTIONS

Here in this chapter, the conclusion is given. We also give some recommendations for using steganography the best way and reducing the risk of failure as possible. Finally, we talk about some future work and research.

4.1 Conclusions

Steganography techniques and algorithms are developed to improve data hiding process aspects which are imperceptibility, capacity and robustness. This project based on LSB substitution to address and improve the robustness and capacity while keeping high imperceptibility. The main contribution of our work is the new way of data hiding that apply randomization based on indicators.

4.2 Recommendations

Steganographic systems cannot be absolutely secure, so it is important to take care when hiding secret data. First, the cover image should be of suitable size to contain the secret data, such that the data dose not fill on average more than 20% of a cover image. Second, it is preferred that the cover image is of medium dimensions or larger such as 512 × 512. Third, when intend to embed data, the cover image should be new and not be available. Fourth, the data must be minimized as possible, for example, if the secret data is of text type, then spaces must be removed, where the ASCII of the space is 32 which is 0010 0000 in binary. As we see the ASCII of the space consists of seven zeros and a one. Since each word of a text is followed by a space, then there would be a lot of spaces to hide, and that means for each space, seven zeros are hidden to a one, which would increase zeros in LSBs for the ones. We can remove the spaces and each word is capitalized to separate words of the text. Fifth, secret data could be encrypted before embedding, to increase the protection and to change ASCII of the text characters.

4.3 Future Work

In some researches it has been claimed that embedding data into Least Two Significant Bits is less detectable than into only the Least Significant Bit. Also we noticed that embedding into only the 2nd LSB is less detectable than into only the LSB or into the Least Two Significant Bits at once. So, we intend to do experiments on huge dataset of images to figure out characteristics of using this approach. Also we intend to do more researches to find out how statistical attacks work through detecting stego images to improve the mechanism of data hiding.

References

1) Akhtar, N., Johri, P., & Khan, S. (2013). Enhancing the security and quality of LSB based image steganography. Paper presented at the Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference on.
2) Al-Ani, Z. K., Zaidan, A., Zaidan, B., & Alanazi, H. (2010). Overview: Main fundamentals for steganography. arXiv preprint arXiv:1003.4086.
3) Al-Mohammad, A. (2010). Steganography-based secret and reliable communications: Improving steganographic capacity and imperceptibility. Brunel University, School of Information Systems, Computing and Mathematics Theses.
4) Asad, M., Gilani, J., & Khalid, A. (2011). An enhanced least significant bit modification technique for audio steganography. Paper presented at the Computer Networks and Information Technology (ICCNIT), 2011 International Conference on.
5) Balaji, R., & Naveen, G. (2011). Secure data transmission using video Steganography. Paper presented at the Electro/Information Technology (EIT), 2011 IEEE International Conference on.
6) Bateman, P., & Schaathun, H. G. (2008). Image steganography and steganalysis. Department Of Computing, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey, United Kingdom, 4th August.
7) Chandramouli, R., Kharrazi, M., & Memon, N. (2003). Image steganography and steganalysis: Concepts and practice. Paper presented at the International Workshop on Digital Watermarking.
8) 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.

9) Dumitrescu, S., Wu, X., & Memon, N. (2002). On steganalysis of random LSB embedding in continuous-tone images. Paper presented at the Image Processing. 2002. Proceedings. 2002 International Conference on.

10) Dumitrescu, S., Wu, X., & Wang, Z. (2003). Detection of LSB steganography via sample pair analysis. IEEE transactions on Signal Processing, 51(7), 19952007. Dunbar, B. (2002). A detailed look at Steganographic Techniques and their use in an Open-Systems Environment. Sans Institute, 2002, 1-9.

11) Easttom II, W. C. (2016). Computer security fundamentals: Pearson IT Certification. Eric, C. (2003). Hiding in plain sight, Stegnography and the art of Covert Communication. Wiley, Indianapolis, Indiana, ISBN, 10, 0471444499.

12) Fridrich, J., Goljan, M., & Du, R. (2001). Reliable detection of LSB steganography in color and grayscale images. Paper presented at the Proceedings of the 2001 workshop on Multimedia and security: new challenges.

13) Goel, P. (2008). Data Hiding in Digital Images: A Steganographic Paradigm. Indian Institute of Technology–Kharagpur.

14) Gowda, S. N., & Sulakhe, S. (2016). Block Based Least Significant Bit Algorithm For Image Steganography.

15) Holub, V. (2014). Content Adaptive Steganography–Design and Detection. Citeseer.

16) Islam, M. R., Siddiqa, A., Uddin, M. P., Mandal, A. K., & Hossain, M. D. (2014). An efficient filtering based approach improving LSB image steganography using status bit along with AES cryptography. Paper presented at the Informatics, Electronics & Vision (ICIEV), 2014 International Conference on

17) Jain, R., & Boaddh, J. (2016). Advances in digital image steganography. Paper presented at the Innovation and Challenges in Cyber Security (ICICCSINBUSH), 2016 International Conference on.

18) Johnson, N. F., & Jajodia, S. (1998). Exploring steganography: Seeing the unseen. Computer, 31(2), 26-34.

19) Juneja, M., & Sandhu, P. S. (2013). Data Hiding with Enhanced LSB Steganography and Cryptography for RGB Color Images. International Journal of Applied Research, 3(5), 118-120.

20) Karim, S. M., Rahman, M. S., & Hossain, M. I. (2011). A new approach for LSB based image steganography using secret key. Paper presented at the Computer and Information Technology (ICCIT), 2011 14th International Conference on.

21) Ker, A. D. (2007). Steganalysis of embedding in two least-significant bits. IEEE Transactions on Information Forensics and Security, 2(1), 46-54.

22) Kipper, G. (2003). Investigator's guide to steganography: crc press.

23) Krenn, R. (2004). Steganography and steganalysis. Retrieved September, 8, 2007.

24) Laha, S., & Roy, R. (2015). An improved image steganography scheme with high visual image quality. Paper presented at the Computing, Communication and Security (ICCCS), 2015 International Conference on.

25) Lubacz, J., Mazurczyk, W., & Szczypiorski, K. (2012). Principles and overview of network steganography. arXiv preprint arXiv:1207.0917.

26) Mahmood, N. R., Azeez, A. A., & Rasool, Z. N. (2014). Public Key Steganography. International Journal of Computer Applications, 100(8).

27) Morkel, T., Eloff, J. H., & Olivier, M. S. (2005). An overview of image steganography. Paper presented at the ISSA.

28) Neeta, D., Snehal, K., & Jacobs, D. (2006). Implementation of LSB steganography and its evaluation for various bits. Paper presented at the Digital Information Management, 2006 1st International Conference on.

29) Nguyen, T. D., Arch-Int, S., & Arch-Int, N. (2016). An adaptive multi bit-plane image steganography using block data-hiding. Multimedia Tools and Applications, 75(14), 8319-8345.

30) Saidi, M., Hermassi, H., Rhouma, R., & Belghith, S. (2016). A new adaptive image steganography scheme based on DCT and chaotic map. Multimedia Tools and Applications, 1-18.

31) Satar, S. D. M., Hamid, N. A., Ghazali, F., Muda, R., Mamat, M., & An, P. K. Secure Image Steganography Using Encryption Algorithm.

32) Sharif, A., Mollaeefar, M., & Nazari, M. (2016). A novel method for digital image steganography based on a new three-dimensional chaotic map. Multimedia Tools and Applications, 1-19.

33) Silman, J. (2001). Steganography and steganalysis: an overview. Sans Institute, 3, 61-76. Simmons, G. J. (1984). The prisoners' problem and the subliminal channel. Paper presented at the Advances in Cryptology.

34) Singh, S., & Kaur, J. (2015). Odd-Even Message Bit Sequence Based Image Steganography. International Journal of Computer Science and Information Technologies, 6(4).

35) Stanley, C. A. (2005). Pairs of Values and the Chi-squared Attack. Department of Mathematics, Iowa State University.

الخلاصة

Steganography هو فن العلم الذي يتعلق بإخفاء الاتصالات عن طريق إخفاء المعلومات السرية داخل وسيط باعتباره الناقل ، ويدعى الغطاء، ليتم إرسالها عبر قنوات الاتصال إلى الأطراف المعنية. يمكن أن يتم إخفاء المعلومات بطرق بسيطة ومباشرة ؛ ومع ذلك ، يجب زيادة أمان المعلومات المخفية قدر الإمكان من خلال تطوير واستخدام طرق أكثر قوة.

واحدة من أكثر التقنيات المستخدمة هي Lest Significant Bit (LSB). الطريقة المباشرة التي تستخدم LSB Substitution هي LSB التسلسلية التي تتضمن البيانات بالتسلسل ، لكنها بسيطة للغاية وسهلة الهجوم. لذلك ، لزيادة أمن المعلومات الخفية ، يجب علينا استخدام هذه التقنية للاختباء بطريقة عشوائية. تم تعيين العديد من الخوارزميات لزيادة أمان البيانات المخفية ، ولكل منها آلية خاصة بها لإخفاء البيانات العشوائية.

يقدم هذا البحث خوارزمية جديدة نسبيًا لإخفاء البيانات ، تسمى LSB المستندة إلى المؤشر ، والتي تضم البيانات التي تعتمد على المؤشر لزيادة أمان البيانات المخفية. تنفذ الخوارزمية باستخدام مؤشرات ، مما يجعل عملية التضمين تتحرك للأمام عبر وسيط الغطاء أثناء عملية الاختباء. باستخدام رقم عشوائي يعرفه المرسل والمستقبل.

هناك عدة أنواع من الـ covers مثل الصور والتسجيلات الصوتية ومقاطع الفيديو ، إلخ. ومع ذلك ، فإن إخفاء المعلومات المبني على الصور هو النظام الأكثر استخدامًا ، حيث يتم استخدام الصور الرقمية على نطاق واسع عبر الإنترنت ، لذلك تم استخدام الصور من خلال بحثنا كوسيط للتغطية التجارب. وفقًا للاختبارات والنتائج ، فإن عشوائي الخوارزمية راضٍ للغاية ، لذلك من الصعب مهاجمة ملفات stego الناتجة. أيضا عملية التضمين للخوارزمية ينتج عنها ملفات stego بجودة عالية ، لذلك لا يثير أي شك.

الكلمات الرئيسية- إخفاء المعلومات ، إخفاء المعلومات ، أمن المعلومات ، تحليل المعلومات ، العشوائية.

جامعة ديالى – كلية العلوم
قسم علوم الحاسبات

نظام إخفاء المعلومات الرقمية

بحث مقدم الى مجلس كلية العلوم – جامعة ديالى – قسم الحاسبات كجزء من متطلبات الحصول على شهادة البكالوريوس في علوم الحاسوب

إعداد الطلاب
حسن هيثم محمد علي
ابراهيم احمد محمود
رقية حسن إبراهيم

اشرف على البحث
د. بشار طالب النعيمي

1441هـ 2020م