T.C

ISTANBUL AYDIN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

# IMAGE STEGANOGRAPHY

M.Sc. THESIS

Waleed TUZA

Department of Electrical and Electronics Engineering

Electrical and Electronics Engineering Program

July 2018

T.C
ISTANBUL AYDIN UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY

# IMAGE STEGANOGRAPHY

M.Sc. THESIS

Waleed TUZA

(Y1613.300002)

Department of Electrical and Electronics Engineering
Electrical and Electronics Engineering Program

Advisor: Assist. Prof. Dr. Necip Gökhan KASAPOĞLU

July 2018

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜ

**Yüksek Lisans Tez Onay Belgesi**

Elektrik-Elektronik Mühendisliği Ana Bilim Dalı Elektrik-Elektronik Tezli Yüksek Lisans Programı **Y1613.300002** numaralı öğrencisi **WALEED TUZA**'nın **"IMAGE STEGANOGRAPHY"** adlı tez çalışması Enstitümüz Yönetim Kurulunun 21.06.2018 tarih ve 2018/11 sayılı kararıyla oluşturulan jüri tarafından .basarı. ile Tezli Yüksek Lisans tezi olarak kabul..edilmiştir.

| **Öğretim Üyesi Adı Soyadı** | **İmzası** |
|---|---|

**Tez Savunma Tarihi : 05/07/2018**

**1)Tez Danışmanı:** Dr. Öğr. Üyesi Necip Gökhan KASAPOĞLU N. Gökhan ......................

**2) Jüri Üyesi :** Prof. Dr. Sedef KENT Sedef Kent ......................

**3) Jüri Üyesi :** Dr. Öğr. Üyesi Evrim TETİK E. Tetik ......................

Not: Öğrencinin Tez savunmasında **Başarılı** olması halinde bu form **imzalanacaktır**. Aksi halde geçersizdir.

I declare that the information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Waleed Tuza

## FOREWORD

After thanks to Allah our creator, I would like to thank my mother and my father who raised me to become a good person. They were patient during my mistakes and my bad times and helped me in all times and everything I have accomplished is because of their effort. I hope I can make them happy and return even some of what they gave me during their whole lives.

I would like to thank my thesis advisor Dr. Necip Gökhan Kasapoğlu for his guidance, support, and help during my work in the thesis. I thank him for everything I learned from him.

I thank all my teachers starting from my school time until today as they had great influence on me and made me love education and I hope I can become one day a good teacher as they were

**July 2018** **Waleed Tuza**

## TABLE OF CONTENT

Page

## LIST OF ABBREVIATIONS

| | |
|---|---|
| **LSB** | Least Significant Bit |
| **LSBG** | Least Significant Bit Gaped |
| **bpp** | bit per pixel |
| **HVS** | Human Visual System |
| **MSE** | Mean Square Error |
| **PSNR** | Peak Signal to Noise Ratio |
| **MAXi** | Maximum pixel intensity value |
| **RQP** | Raw Quick Pair method |
| **PoVs** | Pair of Values statistical analysis |
| **EDC** | Error Detection and Correction |
| **DFT** | Discrete Fourier Transform |
| **DCT** | Discrete Cosine Transform |
| **DWT** | Discrete Wavelet Transform |
| **PRNG** | Pseudo Random Number Generator |
| **SSIS** | Spread Spectrum Image Steganography |
| **LSBM** | Least Significant Bit Matched |
| **COM** | Center of Mass method |
| **LSBMR** | Least Significant Bit Matched Revisited |
| **POC** | Point of Care system |
| **FWHT** | Fast Walsh-Hadamard Transform |
| **2D LSBG** | Two-Dimensional Least Significant Bit Gaped |
| **ALSBG** | Adaptive Least Significant Bit Gaped |

## LIST OF TABLES

**Page**

## LIST OF FIGURES Page

## GÖRÜNTÜ STEGANOGRAFİSİ

## ÖZET

Steganografi, bilgi gizliliği çözümü sağladığından bilgi güvenliği için çok önemli bir tekniktir. Şifrelemeyle karşılaştırıldığında, bilgilerin gizli olarak saklandığı durumlarda, kapak verilerinin bilgi verisi olarak görülmesi avantajına sahiptir.

Böyle bir metodun, steganografi yöntemlerinin kullanıldığı filigran ve parmak izi gibi çeşitli uygulamaları vardır, ancak bu uygulamaların amacı farklıdır. Bu tezde steganografi uygulaması gizli haberleşme sistemleridir. Steganografide metin, görüntü ya da ses gibi farklı türde veriler kullanılabilir.. Deneylerde önerilen LSBG yöntemi görüntülerin kapak verisi olarak seçildiği görüntü stenografisi için uygulanmıştır.

Temel steganografi yöntemlerinden biri olan en az anlamlı bit (LSB) yönteminde bazı sınırlamaların olması dolayısıyla LSB yönteminigeliştirmek için birçok yaklaşım yapılmıştır. LSBM ve LSBMR gibi LSB' ye bazı iyileştirme yöntemleri uygulanmıştır [7]. Bu tezde önerilen iyileştirme yöntemi, gizli verinin LSB' ye göre daha az algılanabilir olduğu ve gizli veriyi güvence altına alacak yeni bir anahtar yapısının önerildiği en az anlamlı bit açıklığıdır (LSBG).

**Anahtar kelimeler:** *Steganografi, En az anlamlı bit, En az anlamlı bit açıklığı*

# IMAGE STEGANOGRAPHY

## ABSTRACT

Steganography is very important technique for information security as it provides a solution of hiding information. Compared to encryption it has the advantage of considering the cover data as the information data itself while actually the secret information is hidden inside it.

Such a method has several applications like watermarking and fingerprinting where steganography methods are used but the objective of those applications are different. The application that we are concerned in for this project is the covert communication systems.

There are different types of cover mediums that can be used in steganography, such as text, image, or audio steganography. We chose image steganography where the images are used as cover object to be our base of experiments for the proposed LSBG method.

One of main steganography methods is least significant bit (LSB) but it has some limitations therefore many approaches have been proposed to improve it. Some improvement methods have been applied to LSB such as LSBM and LSBMR [7]. The new proposed improvement method is least significant bit gaped (LSBG) where the aim is to improve its imperceptibility compared to LSB and to improve a new key structure that will increase the level of securing the information.

**Keywords:** *Steganography, Least Significant Bit, Least Significant Bit Gaped*

# 1 INTRODUCTION

Steganography is the art of hiding data. It is an information security method that can be applied to secure the information by hiding it in a medium where the secret information cannot be observed. In recent years steganography methods have been applied in the digital world where we deal with different digital media images, audio, or video data. Digital steganography is working inside the digital signal systems by using those digital mediums as cover mediums and also the secret message is required to be in a digital data form. The application of steganography simply is to embed secret information data in a selected cover medium to produce a stego medium where it holds the hidden data.

There are wide applications where steganography can be used. Secret and covert communications systems like military communication systems need to have a high level of information security during transmission where steganography take a place as one of the possible solutions [2]. Some widely used application for steganography are watermarking and fingerprinting which they are used for protecting the copyrights and data property for owners.

Another possible application of steganography is secure storage of information [12]. Steganography can be considered as great method to safe information data in undetectable way, which is an important element for securing the data.

Steganography as mentioned can use different types of cover mediums like text medium, image medium, audio medium, and other types of mediums [7]. Also for secret messages, it could be any kind of data like a text or an image etc. In this thesis we are concerned with image steganography where images are used as the cover domain.

## 1.1 History

The word is originally a Greek word, which means “covered writing “[2]. The word is actually composed of two separate words. The first word is stego, which means cover, and the second word is grafia which means writing in Greek [7]. The method itself is very old technique and there are proofs of using steganography even before BC.

Here are some examples of steganography methods that have been used in the history. In 440 BC, Histiæus wanted to send a secret message and found a way to hide the message by using one of his trusted slaves. He shaved the slave's head and tattooed the secret message on its head then waited for the slave's hair to grow back so the tattooed message cannot be seen. After that, he sent his slave to his friends without giving any suspicion to his enemies of sending any useful information [2]. In 17th century, Schott has made a method of hiding information in music scores where each music note does represent a specific letter [2]. It also known that secret messages have been hidden on the backside of paintings. Another way was using invisible ink as an example of hiding messages. For some invisible inks, it can be only observed or detected by a specific light at a specific operating frequency band [2]. Micro writing can be considered as one of the steganography methods where the message can be only observed by magnifying the size of the writing with some optical magnification tools [2]. And still there are a lot of examples of using steganography methods in history and probably still used until today.

## 1.2 Steganography and Encryption

Cryptography is a process that converts the information data to a data form that cannot be read or analyzed unless that form is decrypted where the aim is not to give the capability for third parties to analyze the information. While Steganography is an information security process that hides the information (secret message) in carrier medium (cover) without being able to observe the secret information in the cover medium. The only way to read the secret information is applying the inverse operation of steganography to extract the secret message data from the cover medium. The beauty of such a method that it will give the 3rd party an impression that the information is the cover medium while in reality the real information is hidden inside cover medium but cannot be noticed [1].

Steganography is preferred over encryption by some parties in many applications, as it gives less suspicion to other sides [2]. Both of these techniques are considered as information security techniques. As mentioned before, there are differences in the way of securing the information for both methods. As we have discussed steganography secures the information by hiding it in cover medium in a way that not to be visible. While encryption method is securing the data by processing or converting the information signal to an encrypted signal, by simply scattering the bit stream sequence or rearrange the bit stream to have data form that cannot be read.

One of the elements that both methods share is the fact of having a key to secure both of them. In the absence of the key in both methods will cause the incapability of extracting or reconstructing the message even if other elements are known such as the type of method used. This shows the importance of having a key in the method as it provides additional security to the information.

What is interesting of having both techniques is that you apply both of these techniques to the information signal simultaneously as a two level information security method. The combination of applying both techniques will give a result with a higher level of security and also additional complexity to the system. In other words if the information signal was received by third party, they need to deal with both steganography and encryption analysis to extract the information which will be extremely difficult and complicated to do.

## 1.3 Watermarking and Fingerprinting

As discussed, steganography could be used for different applications such as a copyright mark for property protection (watermarking), a covert communication to transmit the message in a secure way, or a serial number of a product to give the capability to trace that specific product (fingerprinting) [1]. In recent years the main focus of scientific researches and articles was in watermarking and fingerprinting applications because the need of finding some strong solutions for a very important and widely used application which is property and ownership protection. The importance to give the ability to protect the property of the digital mediums products, and that is where watermarking and fingerprinting took a place as possible and applicable solutions. Although the focus of this project is in the covert communication application, it was important to explain the concept of watermarking and fingerprinting for their wide use and importance as some famous steganography applications.

Watermarking is process of marking an object with invisible mark (or in sometimes with visible mark). All objects are marked in the same way with same mark. The aim is prove ownership of these objects by having the owner mark [1]. Watermarking: can be defined with 3 main stages. The first stage is generating the watermark. The second stage is embedding the watermark in the medium. And the third and the final stage is the process of detecting the watermark in the medium to ensure the success of the whole watermarking process [4]. The goal of having digital watermarking is to have the signature of the owner to ensure the protection of his product property [3].

Fingerprinting is a marking process of product but with unique mark or fingerprint for each product. The aim is distinguish each product separately as it has a mark that is different from others [1]. Fingerprints can be defined also as labels by some authors [2]. The main objective of having a unique mark or fingerprint is that the ability to define specifically who did share the product and did not commit to the copyrights laws and by that the copyright owner can take action against him in the law.

For watermarking and fingerprinting, they should have a strong resistance of its removal or from being edited by image processing methods such as compression, filtering, cropping or rotation [3]. That is why the main focus in improving these methods is in their robustness even on the account of their capacity or imperceptibility.

### 1.4 Steganography Types

Applying steganography to digital data can be categorized based on the type of cover medium (also known as host medium as by some authors [6]) used in the method [7]. The cover medium can be any type of data file such as message file, image file, audio file, or video file. There is also the ability to use other data forms as data protocols where the free space of data can be used for embedding the secret message [1]. Regardless of the type of secret message data weather the secret was a text, image, or other data file types, if it was embedded in a cover image then it will be considered as image steganography. Here are the main types as listed below:

- Text steganography: All steganography methods that has been applied in text files (such as .txt, .docx, or .pdf) as a cover medium. Some methods have been used in text steganography such as the process of slight changes of the selected letter (in the form or font size) where it is difficult to distinguish those changes. Another possible method is by using the spacing between words to embed the data as white space method [6].

- Image steganography: The steganography methods that are applied on image files (such as .png, .tiff, .bmp, .gif, etc) as cover medium. Image steganography is the most used type out of others. That is why many methods have been proposed in image steganography such as improved methods of basic LSB.
- Audio steganography: The steganography methods that use audio files (.mp3, .wav, etc) as cover mediums. Some methods used in audio steganography such as parity coding, phase coding, echo hiding, and spread spectrum [7].
- Video steganography are the methods that apply steganography on video files such as (.mp4, .mkv etc) as cover mediums. Basic methods can be applied in video steganography like LSB method.
- Protocol steganography that embed the data in unused of free parts of protocols' frames. Hiding in TCP/IP protocol as an example, in the header specifically in the free and not used parts [7].

## 1.5 Steganography Elements

In steganography method, as a system it can be divided into four main elements as listed below:

- Secret message: it the message that will be embedded in the cover medium. It is actually the crucial element that all other are used to protect it by hiding it so it cannot be detected. It can be any type of data such as simple text message or an image.
- Cover object: is the medium that will be used as a carrier of the embedded secret data. Selecting a suitable cover medium is very important for concealing the secret information based on the steganography method requirements. The cover medium selection process will be explained in details in the following chapter topics.
- Steganography key: The Key can be considered the control data part that you need when you want to apply the inverse operation of steganography

method and extract the secret message. Without the knowledge of the key, secret message could not be extracted from the cover medium. This key adds the level of information security of steganography [1]. The power of having a key is that even if the third party did identify the steganography method but does not knows the key, they will not be able to extract the secret message [2].

- Stego object is the result carrier medium after embedding that secret message inside it. It is very important when we try to compare the stego image with the original cover image, you will not be able to observe any difference visually and that where we can evaluate the process of steganography method to be successful or not.

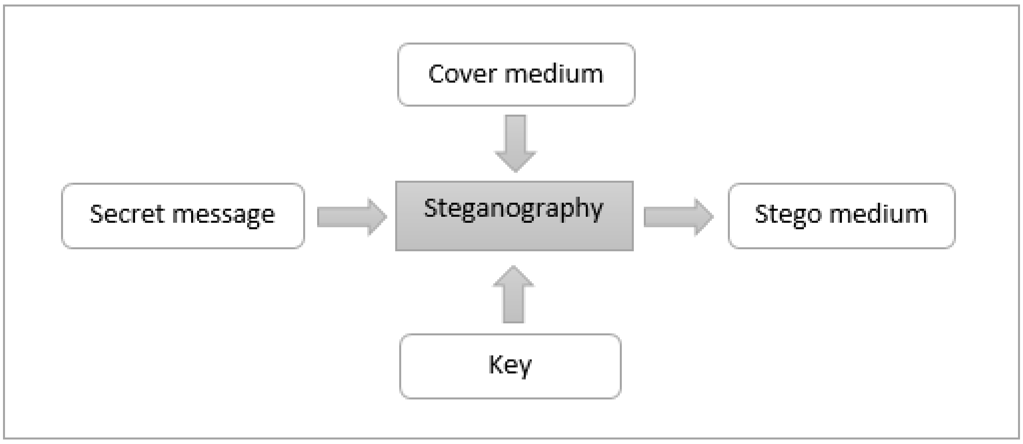


**Figure 1.1:** Steganography System Block Diagram.

## 1.6 Cover Medium Selection

The cover image must be carefully chosen for any specific steganography method. Studying the cover image properties to be suitable where the aim is to find a specific cover medium that after the data embedding, it will not produce any observable difference for observers to be detected by the human visual systems [5].

One of the important parameters to be studied is the capacity of the cover image and the needed capacity of embedding the secret message. It is needed to be sure that the cover image will be able to carry completely the secret message capacity.

Cover image selection process is very important and it is done by studying the characteristics of the image and find the suitable candidates of cover images based on the type of steganography method used and its requirements to ensure a high level of performance and security [14]. As for some steganography methods, specific cover images will be suitable for them but it will not be suitable for other steganography methods that require different cover image properties or requirements. For example some steganography algorithms use embedding in dark areas so the suitable image for the method, is the image that have mainly dark areas. Another example for some algorithms do use the skin-detected areas in cover images for data embedding, so definitely they will require different cover images that contain body skin parts in it.

One on the main characteristics of images is the image histogram. Analyzing the histogram explain the intensities usage in the cover image and the dynamic range or the cover image. From studying the histogram of the images, specific conditions of the histogram could be specified to be suitable for the embedding method. For example for some methods, a wide dynamic range of the cover image histogram is needed to have an appropriate cover image. Histogram analysis is used in our experiments study as evaluation parameter for proper cover images selection and finding the proper histogram conditions for LSBG method.

In general, experts recommend using grayscale images as cover images, they recommend to use images captured from digital cameras or uncompressed scan of photographs. It is also recommended for the cover image to have high variations of colors, which will ensure better results as stego images and will make it for the attackers more difficult to be capable of detecting the secret message [14].

## 1.7 Steganography Main Parameters

Steganography method is controlled by three main parameters, which are capacity, invisibility, and robustness [3]. Based on the application and its objectives or requirements steganography parameter levels can vary where the designer can increase the level of main parameter that he needs to ensure its high performance while it will affect the levels of other consequently.

1. Capacity: can be defined as the maximum number of bits that can be embedded in the cover medium [3]. Another used definition of capacity is the number of embedded bits in the pixel, which can be denoted as bit per pixel (bpp) [7]. The less capacity for secret message to be embedded in

the cover medium, the less probability to detect the existence of a secret message [14], and that is why it is recommended to have a smaller capacity of secret message to be embedded.

2. Imperceptibility: can be also defined as invisibility or undetectability, which means the inability to observe or detect the difference between the stego image and original cover image by the human visual system (HVS) and its sensitivity to the changes of luminance in the image [6]. It is highly recommended spatially for covert communication, is to increase the level of imperceptibility to ensure the difficulty of detecting the modification of the cover medium.

3. Robustness: is the resistance of modification or removal of the embedded data, from any image processing methods such as compression, cropping, filtering, or even noise reduction [6]. Robustness can be measured by testing its resistance of a kind of jamming process to stego medium [3]. Robustness can be increased by redundancy but it will cause the need to have a higher capacity [6]. The improvement of robustness is very important and recommended for watermarking and fingerprinting applications to ensure the immunity of the watermark or fingerprint from being edited or removed.

## 1.8 Some Important Evaluation Parameters in Steganography

Mean Square Error (MSE) is the measurement to calculate the average of summation of squared errors of the difference between the edited image with the original image. The better performance of steganography system is by having as MSE value which indicates that error deviation in the stego image is small compared to the original cover image.

Peak Signal to Noise Ratio (PSNR) is another measurement that can be used to evaluate some steganography method performance [12]. It measure the peak value of signal to noise ratio by calculating the logarithm operation  to the ratio between the max pixel intensity value (MAXi)  divided by the MSE. If MSE value is small then PSNR value is large [13], which indicates a better performance of the steganography method. In other words, the higher value of PSNR shows the better performance of the steganography technique to be considered.

## 1.9 Steganalysis

Steganalysis: are the methods and processes that are used to break and analyze steganography techniques [7]. The concept of Steganalysis is to analyze the steganography technique to detect the modification done by the method to the medium, which can be divided into two main stages:

First stage: Steganography analysis only to detect if it is present in the medium and by that there is a hidden information in the medium. Once the Steganography technique is detected in the medium, different possible signal processing techniques can be applied to destroy the hiding data. As the aim is not be able to extract the hidden information by the true receiver.

Second stage: Steganography analysis to detect the Steganography technique in the medium and also and analyzing the technique to find out the inverse operation of the technique and by that extract the hidden information.

Attacks in Steganalysis can be categorized into, Visual attack which is visual analyses of stego image. The process can be accurately performed by separating some parts of the stego image to be analyzed separately which will help the attacker to find the noise part (edited part) [7]. Another type of attack is Histogram attack, which is done by analyzing the histogram of the stego image to detect the differences in the comparison with the histogram of the original cover image. Detecting the difference will indicate the presence of a secret message in the medium [7].

One of Steganalysis methods for LSB steganography analysis is known as raw quick pair method (RQP) which is done by Fridrich et al. It analyses the close pairs of colors created by LSB. The method can provide a rough estimation of the secret message size. The method works with reasonable result as long the number of unique colors limitation does not exceed 30% of the total number of pixels. RQP can be only applied to color images [14]. Another Steganalysis method which performs statistical analysis of pair of values (PoVs) introduced by Pfitzmann and Westfeld. The method can give reliable results when the placement of the message in the stego image is known [14]. In addition, there is method called RS Steganalysis method, which estimates the length of secret message. It is more accurate for random embedding of LSB than the concentrated LSB in specific area of image [14].

## 1.10 Steganography Limitations

In general, steganography methods are limited by the three main parameters known as capacity, robustness, and imperceptibility. All these parameters do affect each other. For example if imperceptibility need to be improved, one of the solutions it to reduce the capacity of the secret message. Another example if robustness needs to be improved, it can be done by redundancy of the embedded data, which will cause the enlargement of the capacity of the embedded data in the medium.

However, the main limitation of steganography is robustness especially in watermarking application. The method need to be resistant to many image processing techniques which could cause the distortion or even removal of the embedded data. Image processing methods like filtering will cause the change of values of least significant bits in pixels, which will destroy the embedded data [1]. Process such as cropping which may delete a part of the image that could have a portion of the embedded data. Such process may cause the loss of a part of the embedded data and could cause the incapability of retrieving the hidden data [1]. Many steganography methods do not recommend JPEG image because it is one of the file types that supports lossy compression, which will cause the modification of the embedded once decompression is done. The reason of this, lossy decompression does not produce the exact estimate of the original file but it produces an approximate estimate of the original file. It is recommended to use file types that support lossless compression such as GIF, and BMP image files to solve the problem of editing the embedded data [5]. Another solution for compression processes is to find areas for embedding where compression process does not edit or change [6].

## 1.11 Steganography in Communication Systems

In covert communication systems, the most important parameter for steganography to be considered is imperceptibility. The main objective for the hidden data is not to raise any suspicion that the cover medium has been edited [4]. That is why the aim of the designer is to maximize the level of imperceptibility on the expense of having reduced levels of capacity and robustness [3].

One known example for covert communication is military communication systems they use spread spectrum modulation or frequency scattering transmission as some of the possible solutions to protect the transmission from being detected by enemies and their attacks on the signal by jamming [2].

Steganography as an information security technique can be very important role in communication systems. Especially some communication systems that are used for military applications where communication of information is considered as classified and it is very important not to received and analyzed by a third party. That is why it is important to send the information in a secured way where if it was received by third party. They cannot analyze it and that is where Encryption and steganography can take place. Those information security techniques can be used as pre stages of the communication system to secure the information signal from being used by the third party.

Using lossless compression as pre-stage to steganography in the communication system will give the capability to extract and reconstruct the secret message 100% correctly [5]. Moreover, it will be a great solution to reduce the capacity of secret message. Error detection and correction (EDC) coding is used to ensure correct reconstructing of the embedded data. The figure below shows a block diagram of the communication system that has steganography method as pre stage:

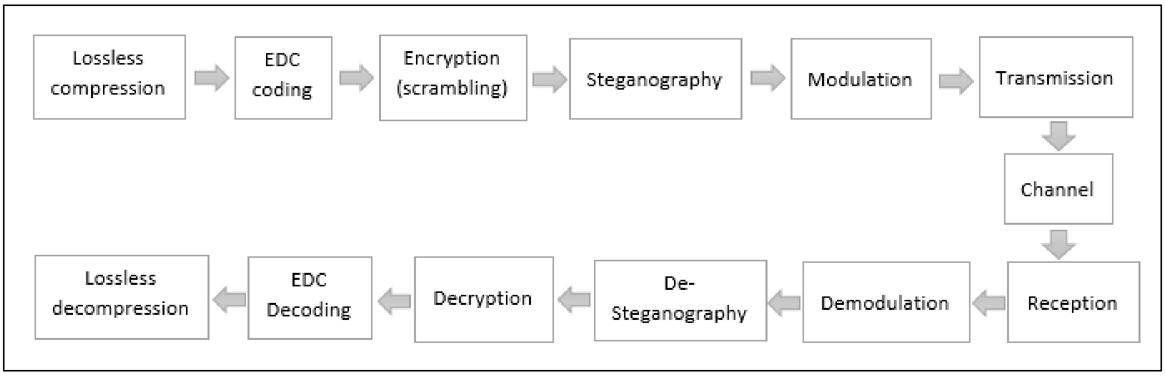


**Figure 1.2:** Covert Communication System Block Diagram.

### 1.12 Steganography Extraction Schemes

For the inverse operation of steganography, the extraction process can be done mainly in two different methods. Some techniques need to have the original cover signal in the extraction process of the hidden data from the stego signal; such methods are known as cover escrow. Then comparison between the original cover medium with the stego medium can be done by subtracting the stego signal from the cover signal and from the result, the embedded data can be identified [3]. While the other method known as blind schemes. In this method to extract the hidden data from the stego image is performed without the need of having the original cover signal. Knowing the steganography method used in addition of knowing the key used in the method is enough to extract the hidden embedded data [3].

### 1.13 Steganography Domains

For steganography methods mainly work into two domains as shown below:

In spatial domain basic steganography methods like LSB and other improvement approaches to LSB method like LSBM operate in the spatial domain of the medium where the secret data is directly embedded in pixels of the spatial domain.

Transform domain: Some steganography methods use transform function like Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), and Discrete Wavelet Transform (DWT) before data embedding [2]where actually the data is embedded in the transform domain such as the frequency domain when for example DFT is applied. Using transform function to apply steganography methods in the transform domain can increase the robustness against image processing like compression, filtering, and even noises.

# 2 STEGANOGRAPHY METHODS

In this chapter, we will study some steganography methods. Starting from LSB method as a base line method. In addition, some of methods that are proposed as improvement methods to LSB method. Also some methods that use different techniques for data embedding will be studied in this chapter. The main concept of these methods will be explained briefly with some examples if possible to demonstrate how the methods work.

## 2.1. Least Significant Bit (LSB)

LSB is a steganography method of impending hidden data in the least significant bits of the cover data bytes as in image pixels. It is one most famous, basic, and simple embedding method used in steganography. Many improvement approaches have been applied to LSB method to improve the parameters of imperceptibility and robustness of the method. For any interested researcher in digital steganography start by studying LSB as the oldest and base concept of embedding for many other steganography methods. In basic LSB method, the pixels are used for embedding in a sequential form [5]. LSB hiding can be done also in a randomized way, which will be explained later [7].

To understand simply how to embed the data in LSB method, LSB is located as the last bit from the right side of any binary value [7]. For example, if we have a byte binary value of (11110101), the first LSB is (1), and the first two LSB bits are (01), also if we want to know the last three LSB bits are equal to (101). From the example, we need to mention that for LSM embedding, it is possible to use only the last bit, last two bits, or last three bits in the binary value of the byte or pixel for data embedding. Therefore, if we have a 1-byte pixel size, the minimum LSB capacity is 1 bit per pixel (bpp) [7].

Let have three pixels of the cover image with binary values of (11110001 11110000 11110011) and we want to embed a message of (110) in those three pixels. So 1 bit is needed for LSB embedding in each pixel. By that, the result stego pixels are (11110001 11110001 11110010). As you can see for the first pixel, the secret bit is equal to last bit in the pixel and that is why the pixel LSB value is still the same. While in the second and the third pixels, the LSB bits were changed to the value of the second and third bits of the message as highlighted in red. Another point that to mention that probability of changing for each bit is 0.5 as we have half to half change that the secret and cover bits are equal [7].

### 2.2. LSB with Pseudorandom Coding

One of the improvement methods applied to LSB is using pseudorandom coding for random selection of the cover medium parts (pixels) for LSB embedding. The random embedding can be done by using pseudo random number generator (PRNG) [7]. The specific pseudorandom unique code used in the LSB steganography is actually the key of the method. Without having the key of pseudorandom code, the message cannot be extracted in the receiver side [2]. The randomization method has the same characteristics of as random noise in the cover medium. So the embedded data could be considered as noise by third party, and that is the aim of using pseudorandom coding [2].

### 2.3. Spread Spectrum Image Steganography

Spread spectrum image steganography is denoted as SSIS. It is also using pseudorandom code generator for selection. Selection pairs of image regions based on the number generated by pseudorandom generator. In one region, the secret data embedding is done as follows; the intensities are increased by constant while the other region are decreased by the same constant value [3]. SSIS is one of the blind scheme methods where there is no need to have cover image for the extraction of the hidden information from the stego image in the receiver part. Only there is the need of the pseudorandom key that has been used in the steganography method [3].

In SSIS error correcting technique to solve the error probability of the extracted embedded data and image restoration technique is used to produce an approximate of the cover image in the receiver part for processing the extraction of the hidden data and by that having the blind scheme stego decoding system [3].

Spread spectrum signal (spreading the bandwidth of a narrow band signal) has the same characteristics of additive white Gaussian noise. Spreading the narrow band signal to larger bandwidth will cause to decrease the energy level of the signal to be similar to energy level of noise. And such a level is not in the detection range of communication systems since it is considered as noise energy level [3]. Therefore in SSIS, the embedding of secret data is embedded to act as a noise random behavior in the cover medium [3].

### 2.4. LSB with Selection Algorithms

Some algorithms study the characteristics and properties of the cover medium and then select specific suitable region in the cover medium based on parameter valued required in the algorithm. For example is an image is used a cover medium, the algorithm select specific pixels for LSB embedding based on analyzing the pixel and its surrounding pixels variances. The selection of the pixels is used if their variance is within the acceptable levels for the algorithm. Other algorithms could use other measured parameter for selection method of LSB embedding [2].

Some improvement selection methods, do select the noisy area in images for embedding [5]. Other improvement methods perform the selection of dark areas for LSB embedding [9]. Most of these selection methods improve the imperceptibility of the steganography, spatially when it is compared to the basic LSB steganography method.

### 2.5. Least Significant Bit Matched (LSBM)

Least significant Bit matched algorithm (LSBM) differs in the way of data embedding of basic LSB. The embedding method is applied by adding +1 or -1 to each pixel instead of changing last bits in the pixel [7]. Addition of one could be considered as equivalent to binary value of zero, while subtraction of one is considered as equivalent to binary value one. Therefore, the LSBM pixel capacity is one bpp.

For example, if we have a pixel value of 31 in decimal which is equivalent to (00011111) in binary. If we add 1 to the pixel it will become 32 in decimal and (00100000) in binary. And if we subtract one from the pixel intensity value, it will become 30 in decimal and (00011110) in binary.

The improvement will solve the asymmetry problem of LSB, as the probability of increasing or decreasing the modified pixels are approximately the same [7]. However, Center of mass (COM) method of histogram characteristic function by Harmsen and Perlman can detect LSBM method [7].

### 2.6. Least Significant Bit Matched Revisited (LSBMR)

LSB Matched Revisited is proposed by Jarno Mielikainen. The method uses two pixels for embedding two units of the secret data. Based on the relation between the comparison process for both pixels bit values and secret bit values a result of 4 possible cases is in the output of stego image pair [7]. As LSBM, the maximum pixel capacity rate for LSBMR is one bpp [7].

One of the improvement fields for LSBMR is the reducing the probability of bit change in data embedding. The bit change probability is reduced because of the maximum change that could happen for the pair of pixels is one pixel to be modified. This improves the imperceptibility level of LSBMR compared to LSB method [7].

### 2.7. LSB with B+trees

B tree is a fast indexing method used to compress the secret message. It is a data structure that can be used to locate file in a database within a short period of time. The proposed method is to combine LSB method with B+trees compression method. The process can be applied by firstly compress the secret message using B+trees and produce nodes that will contain the value of key to be used in steganography. Then secondly, LSB steganography is applied to embed the compressed data in the cover image [8].

### 2.8. Quadtree Partition

Quadtree partition is image partitioning method that will divide the image into guardable parts, which can be presented as a tree structure. The partitioning is applied based on the differences between image parts intensities [9].

The process can be applied by having different threshold levels for intensity differences, which provides different number of blocks with sizes that starts from 2x2, 4x4, 8x8, 16x16, 32x32, and 64x64 as a result. Then the selection of fine-grained areas is applied for the secret message embedding. For example, if (2x2) block was chosen for message embedding. The block has four pixels and if (one bit) LSB is embedded then you will have the ability to hide four bits in the (2x2) sub-block [9].

### 2.9. Walsh-Hadamard 3D Steganography

Walsh-Hadamard 3D steganography is used as one of the medical covert communication application to ensure the privacy of the data transfer in remote point of care (POC) application. In the beginning, to explain the POC system which is a medical care system where the patients are monitored remotely from their homes by using a devices and sensor to send the important medical signals every certain period of time. The aim to use steganography for POC system is the need to protect the privacy of the medical information of the patients as it considered being a low on hospitals to secure the patient information in some countries [10].

Fast Walsh-Hadamard transform (FWHT) is used for transforming the medical signals to coefficients and the less significant values of the coefficients will be used. FWHT transforms the signal from spatial domain to frequency domain and there a group of values as a result of the transform which are identified as coefficients. Least significant values of the coefficient are used to embed the secret information. The key to be used is three dimensional coefficients (3 layers key) which increases the level of securing the hidden information. One of the advantages of applying this steganography method over cryptography is the ability of being secure the medical information is less data capacity and power consumption [10].

### 2.10. Raspberry Pi Steganography

Raspberry Pi is a microcomputer which can be programmed to perform system required operations and one of its applications is home automation processes. Raspberry Pi is used as a sensor web node, which is programmed to perform two optional steganography techniques based on the secret message type (text or image). The Raspberry Pireceives the secret message, then process and applies steganography technique then send the stego image to the specified destination [11].

For secret text embedding in the cover medium option, a combination of 3 bits per 2 pixels LSB method and Super-knight’s tour algorithm are used. Super-knight’s tour algorithm is required for producing a location map that will be used for text embedding. For the option of secret image embedding, Discrete Cosine Transform (DCT) based steganography method is used [11].

### 2.11. Skintone Detection Based Steganography

The method can be only applicable for cover images that have body skin as an object in them. The method is composed of two stages. The first stage is to apply skintone detection on hue and saturation values then the color model to detect specifically the skin area in the cover image. The second stage is applying transformation of the image into frequency domain using Discrete Wavelet Transform (DWT). In addition, choose the sub band low-low (LL) in the frequency domain for the secret data embedding in the selected subband. For adding an additional complexity to the system, cropping process is used for securing the method where the cropped area of the image is used as a key for the steganography method [13].

# 3 METHODOLOGY OF PROPOSED LSBG METHOD

In this chapter, the proposed Least Significant Bit Gaped will be explained in details. Starting from explaining the Gapping concept and how it can be used to apply the LSBG method. Examples will be given to demonstrate the concept cases clearly. In addition, some ideas are proposed of how to increase the LSBG method flexibility capabilities.

## 3.1. Least Significant Bit Gaped (LSBG) Methodology

The concept of the proposed LSBG method is to embed the secret message using LSB method in certain pixels with a certain rate. In addition, have a gap pixels with a certain rate also where no embedding of data in these pixels. Therefore in the stego medium, the pixels will be divided into two parts. First LSB pixels and those are the selected pixels for LSB embedding. The second part is the gap pixels which can be defined as the pixels that are not selected for LSB embedding. The reason for naming gap pixels is the fact that those pixels have not been used to contain the real data which is the secret message and that is why it can be considered as gaps. The LSB pixels to gap pixels ratio can vary from one LSB pixel to one gap pixel (1LSB to 1G), one LSB pixel to two gap pixels (1LSB to 2G), one LSB pixel to three gap pixels (1LSB to 3G), and so on. The main idea of applying such a method is to have the ability to distribute the secret embedded LSB pixels in the majority of the cover medium area (image pixels area) as much as possible. The LSB pixels to gap pixels ratio will depend on two main factors:

1. The capacity of the secret message to be hidden. The smaller the capacity of the secret message, the more possible ratios can be applied in the cover medium.
2. The capacity of the cover medium (cover image). The larger the capacity of the cover image will give the capability to apply high variety of LSB to G ratios.

From studying the previous two factors, we can select a suitable LSB to gab ratio to be chosen based on the capacity of the secret message and the capacity of the chosen cover image to apply LSBG method.

In LSBG method, LSB pixels to gap pixels ratio can be performed in either horizontal axis or vertical axis in the cover image. The selection of the axis can be chosen by the designer. One important thing that is needed to be mentioned is that when we compare the use of both axis' it will result a completely different selection of LSB pixels for each axis.

To explain LSBG method with different possible ratios, let us assume that we have a cover image with a size of (8x8) pixels as shown in Figure ***3.1***below:

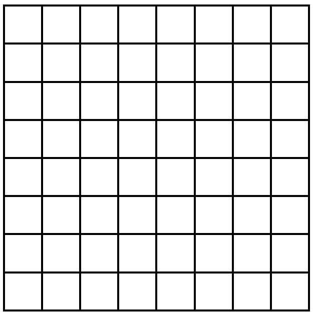


**Figure 3.1:** Cover image before applying LSB or LSBG method.

As seen in Figure ***3.1*** above, the cover image is demonstrated with white pixels where still neither basic LSB method nor LSBG is applied. For LSB or LSBG embedding method the LSB pixels are highlighted in gray, which demonstrate that the grey pixels have been modified and used for LSB embedding. Let us assume that we have a secret message with the capacity of 24 bits (3 byte) is needed to be hidden. The last 2 bits of LSB (2 bit LSB) will be used for data embedding in each pixel. Therefore, we need a sum of 12 LSB pixels for data embedding. In Figure 3.2 a basic LSB method is applied for secret message embedding and highlighted pixels in gray are the pixels used for LSB method.

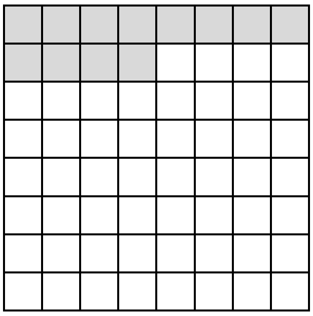


**Figure 3.2:** Stego image of LSB method.

Once we apply LSBG method with the ratio of one LSB pixel to one gap pixel in the horizontal axis. The LSBG embedding is shown in Figure **3.3** where highlighted pixels in gray are the ones used for LSBG method.

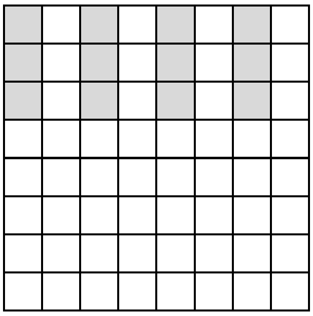


**Figure 3.3:** Stego image of LSBG method with 1LSB /1G ratio in horizontal axis method.

In case of vertical axis is chosen for applying one LSB pixel to one gap pixel ratio of LSBG method the result stego image is shown in Figure **3.4** where highlighted pixels in gray are the ones used for LSBG method.

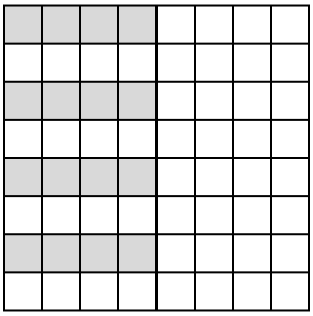


**Figure 3.4:** Stego image of LSBG method with (1LSB /1G) ratio in vertical axis.

If one LSB pixel to three Gap pixels ratio is used in LSBG method in horizontal axis the distribution of LSB pixels can be depicted in Figure **3.5** where highlighted pixels in gray are the ones used for LSBG method.

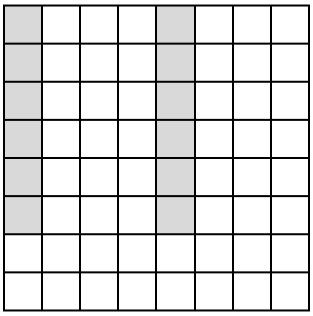


**Figure 3.5:** **S**tego image of LSBG method with (1LSB /3G) ratio in horizontal axis.

## 3.2. Two-dimensional LSBG (2D LSBG)

The LSB pixels to gap pixels ratio can be applied in the vertical axis (y axis) or in the horizontal axis (x axis) as explained before. However, there is also the capability to apply (two dimensional) LSB pixels to gap pixels ratio in both vertical and horizontal axis. The 2D LSBG provides a greater LSB pixels distribution in the cover medium.

If we have the same cover image with a size of 8x8 pixels and in addition, the same secret message with the capacity of 24 bits (3 bytes) then finally, 2 bit LSB is used where 2 bits of secret information is embedded in each LSB pixel and by that, a total of 12 pixels are needed. The 2D LSBG method with the ratio of one LSB pixel to one Gap pixel can be applied as in Figure **3.6** where highlighted pixels in gray are the ones used for 2D LSBG method with 1LSB /1G ratio.

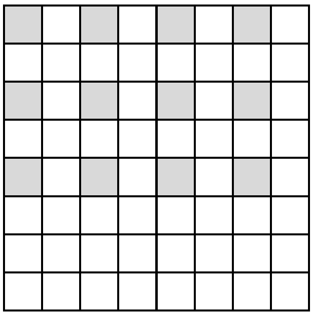


**Figure 3.6:** Stego image of 2D LSBG method with (1LSB /1G) ratio.

## 3.3. Shifting Property for LSBG

Shifting property can be applied to either LSB or LSBG method where the aim is not to apply the steganography technique from the beginning of the cover medium but to shift the starting point (starting LSB pixel) with a certain number that is chosen by the designer. Applying this method will increase the difficulty of extracting the secret information from the LSB pixels by third parties since the starting point is only known by the designer. The shifting property must be added as an additional element to the LSBG key.

In Figure 3.7, a demonstration of applying the shifting property to a 2D LSBG method where the shifting of embedding made the starting point in the 3rd column of the cover image. In the figure highlighted pixels in gray are the ones used for 2D LSBG method.

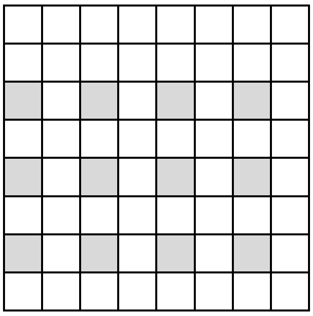


**Figure 3.7:** **S**tego image of 2D LSBG method with 1LSB /1G ratio and with the shifting property.

### 3.4. Applying Flexible Ratio in LSBG

As discussed before, fixed the LSB pixels to gap pixels ratio is used. But a flexible ratio can be applied instead of fixed ratio. For example, we can use the ratio of 1LSB to 2G for the odd LSB pixels and another ratio of 1LSB to 4G for the even LSB pixels. We can have even more than two ratios to be used in the same LSBG. We can arrange the sequence of these ratios in different possible ways. But actually using different and flexible ratio for LSBG method will add the complexity to be applied even for the designer whether it was in the encoding part of the decoding and extraction part.

### 3.5. Adaptive LSBG

The selection of LSB pixels to gap pixels ratio is both the needed capacity of the secret message and the capacity of the cover image. The aim is to apply an adaptive LSBG method will give the optimized LSB pixels to gap pixels ratio based on calculating the needed capacity to hide the secret message and calculation the cover image capacity with different possible ratios that can hide the total secret message.

### 3.6. Ability of Using LSBG for Multiplexing

Let us assume that three secret messages are wanted to hide in the same cover medium. The aim is to hide each message separately. LSBG can be used in a way if we assume that 1LSB to 3Gaps ratio is selected. Therefore we each four pixels can be used as the first pixels of LSB. The idea is to use the set of four pixels where the first pixel is assigned for the first secret message, the 2nd pixel for the 2nd secret message, and the same process for the 3rd secret message. By that, 3LSB to 1Gap ratio can be used. But keep in mind that if we want to extract the 1st secret message the 1st pixel out of each 4 pixels sets is needed to be extracted with the same ratio of 1LSB to 3Gaps, because we are only concerned with the 1st secret message and by that the other LSB pixels for the other secret messages are considered as gap pixels.

The same extraction process must be applied if we want to extract the 2nd message, we will extract the 2nd LSB pixel out of each 4 pixels set in the cover medium with the same1LSB to 3Gaps ratio. And the same process can be for the extraction of the 3rd secret message.

In Figure **3.8**, a demonstration of applying three secret messages using LSBG method can be seen. Each secret message is highlighted with a different color to be distinguished clearly from the other messages Highlighted pixels in gray, yellow and red for 1$^{st}$ , 2$^{nd}$ and 3$^{rd}$ secret messages for LSBG method with 1LSB /3G ratio, respectively.

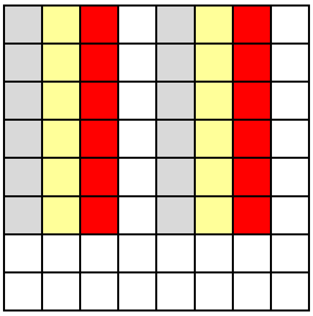


**Figure 3.8:** Stego image of LSBG method with 1LSB /3G ratio for three multiplexed secret messages.

### 3.7. Scrambling as Pre-stage

We have discussed before the capability of combining both encryption and steganography and how it improves the level of information security by increasing the complexity of the key used in the method. Simple encryption method like scrambling which can be defined as changing the order of the secret binary bitstream and rearrange it in a way that it cannot be read. Applying scrambling as a pre-stage before applying LSBG method adds the use of the key of scrambling. Without having this key, even if secret message was extracted successfully it cannot be analyzed since secret message is still scrambled and decryption is still needed.

### 3.8. LSBG Key Elements

One of the main improvement points of LSBG over LSB is the complexity of the key elements of LSBG method. Having a multi levels or elements key in steganography is very powerful tool that will strengthen the security of the steganography method. That is available in LSBG method as you do not only need to know the length of the secret message and the number of LSB bits used in the pixels embedding as in LSB method. You actually need to have all the elements of the LSBG key to be able to extract the embedded secret message successfully. The LSBG key elements can be defined as follows:

- The capacity of the secret message (Total number of bits of the secret bitstream).
- The selection of number of LSB bits to be embedded in each pixel by the designer.
- The selection of LSB pixels to Gap pixels ratio to be used for data embedding.
- The selection of the start point of LSBG embedding. An additional applicable element by using shifting property.
- The LSB sequence number in the LSB to G ratio. (If multiple messages multiplexing is used).
- The possible usage of encryption (scrambling) before applying LSBG method as pre-stage. Adding the key of encryption will cause the embedding the secret bits not in a sequential way in the selected LSB pixels.
- The possible usage of Lossless compression method to the secret message before applying LSBG method. Adding the key of compression as a pre-stage, will help in reducing the capacity of the secret message and also while

extraction from LSBG, the secret cannot be analyzed unless it is decompressed.

- Band selection for LSBG method. This is applicable certainly; if multiple bands color images are used like RGB images. Where one band would be selected while the other band will remain the same.

### 3.9. Possible Transfer Solutions for the Key

There are two approached for the key transfer to the receiver side. The first approach is to send the key data separately in another independent communication channel with defining a coding method to show the key information is a coded form to be only understood by the two concerned parties.

The other approach is to transmit the key data in the same stego medium by embedding the key in specific region in the image that can be agreed on previously and fixed for all steganography communications. The key elements can organized as a header control part of the secret bit steam to be embedded but it will be actually separated and embedded separate, specific, fixed, and agreed on region in the cover medium. The key data element can be presented approximately within a frame with size. The following figure (Figure **3.9**) is demonstration of possible frame structure of key control data.

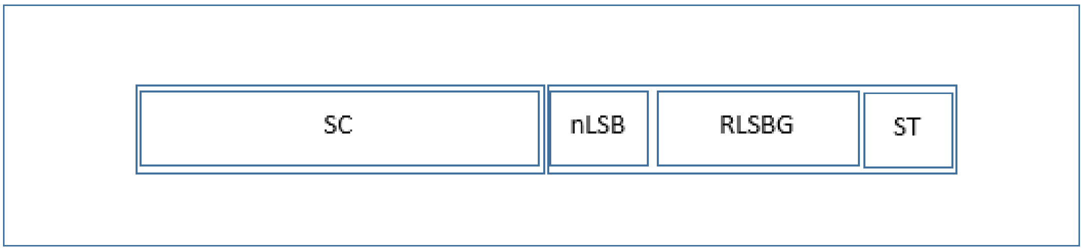


**Figure 3.9:** Possible frame structure of Key data elements.

As shown in the figure above, the key frame structure should have the following sub-frames:

- SC: the total capacity of the secret message bit stream.
- nLSB: number of LSB bits to be embedded in each selected pixel.
- RLSBG: LSB to G ratio number.
- ST: starting point of LSB method (if shifting property is applied).

The bit length required for each sub-frame part can be defined based on the secret message requirements and the type of LSBG method applied. Other sub-frames could be added for the key frame according to the addition of extra possible key elements as explained before.

### 3.10. LSBG Decoding

LSBG decoding can be considered as a blind scheme decoder since there is no requirement to have the original cover medium to perform the extraction of the embedded secret message from the stego medium. But instead it is only required to the all the key elements and details to perform successful extraction and reconstruct the message data correctly. The decoding LSBG procedure are explained in details as the followings.

### 3.11. MATLAB code procedure for LSBG Encoding

MATLAB codes have been written to perform steganography methods of LSB and proposed LSBG. The procedure of encoding part where secret data embedding is applied is as shown below:

1. Read the secret message using the function `imread`. The secret message used is an image.
2. Convert the secret message to a vector, and then convert the vector to a binary bitstream that represents the secret message using function `dec2bin`.
3. Find the total number of binary bits of the secret message to be embedded using function `size`.
4. Read the cover image (it is used to carry the embedded data) using function `imread`.
5. Find the number of rows and columns of the cover image using function `size`. This information is needed for the for-loops.
6. Initially make the stego image equal to cover image.
7. Make a variable to be used for counting the secret bits to be embedded in the for-loops
8. Produce a two dimensional for-loops to move within the cover image pixel by pixel. For LSBG embedding the increment will be edited instead of one as for LSB to become as selected by the designer as either an increment of two or

more.

9. The selected pixel of the cover image will be edited first by making the used LSB bits zeroes using function (bitshift).
10. Two bit of secret bitstream is taken for the selected pixel embedding.
11. Insert secret 2bits in the selected and edited cover pixel by simply adding the decimal value.
12. If the counter reached to the last bit in the secret bitstream (last $w = n - 1$ and , $w + 1 = n\ w + 2 = n + 1$) then end the first for loop
13. If the counter reached to the last bit in the secret bitstream (last $w = n - 1$ and $+1 = n\ w + 2 = n + 1$) then end the second for loop.
14. Once the loops are ended, then the secret message have been embedded completely in the cover image
15. Show the figures of the cover image and the stego image to make the comparison using function `figure`.

MATLAB code for LSBG Encoding can be found in Appendix.

### 3.12. MATLAB code procedure for LSBG Decoding

In the decoding part, having the original cover image is not required. Having the exact values of length of secret bitstream and the LSB to Gap ratio used in the transmitter side is required to apply correct extraction. The procedure of decoding part where secret data extracting is applied is demonstrated step by step as follows:

1. Read the stego image using function `imread`.
2. Find the number of rows and columns of stego image to be used in for-loops.
3. Define an empty matrix for the secret image convert it to a vector.
4. From the vector create initially an empty bitstream to be filled after decoding with the extracted secret bits.
5. Define the total number of binary bits in secret message. This is needed for the counter of the bitstream.
6. Produce a counter for secret bitstream to be extracted.
7. Produce a two dimensional for-loops to move within the cover image pixel by

pixel. For LSBG embedding the increment will be edited instead of one as for LSB to become as selected by the designer as either an increment of two or more. It is important to use the same increment in the encoding (the same LSB to G ratio) to apply correct extraction.

8. Start with of selected stego pixel and converted to binary value using `dec2bin`.
9. Count how many bits is converted in the binary value of the selected pixel.
10. Only select the last 2 bits (LSB)
11. The selected bits are inserted in the empty bitstream
12. The counter is incremented by 2 for extracting the next 2 bit form the next LSB pixel.
13. If the counter reached to the total number of secret binary bits, end the first loop.
14. If the counter reached to the total number of secret binary bits, end the first loop.
15. Convert the extracted binary bitstream into a pixel matrix to be reconstructed in it is original form as a secret message.
16. Show the Figure of secret image using the function `figure`.

MATLAB code for LSBG Decoding can be found in Appendix.

### 3.13. LSB and LSBG capacity

For finding the maximum possible capacity of LSB method in a cover image is shown in equation (3.1) below:

$$Max\ capacity\ for\ LSB = n_P.n_b \tag{31}$$

where $n_P$ is total number of pixels and $n_b$ is total number of bits used. While the maximum possible capacity of LSBG method in a cover image is shown in equation (3.2) below:

$$Max\ capacity\ for\ LSBG = \frac{n_P.n_b}{(n_g + 1)} \tag{3.2}$$

Where $n_g$ is number of gaps. Finally, the maximum possible capacity of 2D LSBG method in a cover image is given in equation (3.3).

$$Max\ capacity\ for\ 2D\ LSBG = \frac{n_P.n_b}{(n_g + 1)^2} \tag{3.3}$$

The following tables show in details all the possible maximum capacities based on the cover image size, the number of LSB bits to be used, and the LSB to Gap ratio. Those capacities have been calculated for standard cover images with the size of 256x256, 512x512, and 1024x1024.

**Table 3.1:** Max capacity of LSB (1 bpp) and LSBG (1 bpp) for a cover image with the size of 256x256 and pixel capacity =1 byte.

| Method | Max Capacity |
|---|---|
| LSB (1 bpp) | 65,536 bits (8,192 byte) |
| LSBG (1 bpp) (1LSB to 1G) ratio | 32,768 bits (4,096 byte) |
| LSBG (1 bpp) (1LSB to 3G) ratio | 16,384 bits (2,048 byte) |
| LSBG (1 bpp) (1LSB to 7G) ratio | 8,192 bits (1,024 byte) |
| LSBG (1 bpp) (1LSB to 15G) ratio | 4,096 bits (512 byte) |
| LSBG (1 bpp) (1LSB to 31G) ratio | 2,048 bits (256 byte) |
| LSBG (1 bpp) (1LSB to 63G) ratio | 1,024 bits (128 byte) |
| LSBG (1 bpp) (1LSB to 127G) ratio | 512 bits (64 byte) |

**Table 3.2:** Max capacity of LSB (1 bpp) and 2D LSBG (1 bpp) for a cover image with the size of 256x256 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (1 bpp) | 65,536 bits (8,192 byte) |

| | |
|---|---|
| 2D LSBG (1 bpp) (1LSB to 1G) ratio | 16,384 bits (2,048 byte) |
| 2D LSBG (1 bpp) (1LSB to 3G) ratio | 4,096 bits (512 byte) |
| 2D LSBG (1 bpp) (1LSB to 7G) ratio | 1,024 bits (128 byte) |
| 2D LSBG (1 bpp) (1LSB to 15G) ratio | 256 bits (32 byte) |
| 2D LSBG (1 bpp) (1LSB to 31G) ratio | 64 bits (8 byte) |
| 2D LSBG (1 bpp) (1LSB to 63G) ratio | 16 bits (2 byte) |
| 2D LSBG (1 bpp) (1LSB to 127G) ratio | 4 bits |

**Table 3.3:** Max capacity of LSB (2 bpp) and LSBG (2 bpp) for a cover image with the size of 256x256 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (2 bpp) | 131,072 bits (16,384 byte) |
| LSBG (2 bpp) (1LSB to 1G) ratio | 65,536 bits (8,192 byte) |
| LSBG (2 bpp) (1LSB to 3G) ratio | 32,768 bits (4,096 byte) |
| LSBG (2 bpp) (1LSB to 7G) ratio | 16,384 bits (2,048 byte) |
| LSBG (2 bpp) (1LSB to 15G) ratio | 8,192 bits (1,024 byte) |
| LSBG (2 bpp) (1LSB to 31G) ratio | 4,096 bits (512 byte) |
| LSBG (2 bpp) (1LSB to 63G) ratio | 2,048 bits (256 byte) |
| LSBG (2 bpp) (1LSB to 127G) ratio | 1,024 bits (128 byte) |

**Table 3.4:** Max capacity of LSB (2 bpp) and 2D LSBG (2 bpp) for a cover image with the size of 256x256 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (2 bpp) | 131,072 bits (16,384 byte) |
| 2D LSBG (2 bpp) (1LSB to 1G) ratio | 32,768 bits (4,096 byte) |

| | |
|---|---|
| 2D LSBG (2 bpp) (1LSB to 3G) ratio | 8,192 bits (1,024 byte) |
| 2D LSBG (2 bpp) (1LSB to 7G) ratio | 2,048 bits (256 byte) |
| 2D LSBG (2 bpp) (1LSB to 15G) ratio | 512 bits (64 byte) |
| 2D LSBG (2 bpp) (1LSB to 31G) ratio | 128 bits (16 byte) |
| 2D LSBG (2 bpp) (1LSB to 63G) ratio | 32 bits (4 byte) |
| 2D LSBG (2 bpp) (1LSB to 127G) ratio | 8 bits (1 byte) |

**Table 3.5:** Max capacity of LSB (3 bpp) and LSBG (3 bpp) for a cover image with the size of 256x256 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (3 bpp) | 196,608 bits (24,567 byte) |
| LSBG (3 bpp) (1LSB to 1G) ratio | 98,304 bits (12,288 byte) |
| LSBG (3 bpp) (1LSB to 3G) ratio | 49,152 bits (6,144 byte) |
| LSBG (3 bpp) (1LSB to 7G) ratio | 24,576 bits (3,072 byte) |
| LSBG (3 bpp) (1LSB to 15G) ratio | 12,288 bits (1,536 byte) |
| LSBG (3 bpp) (1LSB to 31G) ratio | 6,144 bits (768 byte) |
| LSBG (3 bpp) (1LSB to 63G) ratio | 3,072 bits (384 byte) |
| LSBG (3 bpp) (1LSB to 127G) ratio | 1,536 bits(192 byte) |

**Table 3.6:** Max capacity of LSB (3 bpp) and 2D LSBG (3 bpp) for a cover image with the size of 256x256 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (3 bpp) | 196,608 bits (24,567 byte) |
| 2D LSBG (3 bpp) (1LSB to 1G) ratio | 49,152 bits (6,144 byte) |
| 2D LSBG (3 bpp) (1LSB to 3G) ratio | 12,228 bits (1,536 byte) |

| | |
|---|---|
| 2D LSBG (3 bpp) (1LSB to 7G) ratio | 3,072 bits (384 byte) |
| 2D LSBG (3 bpp) (1LSB to 15G) ratio | 768 bits (96 byte) |
| 2D LSBG (3 bpp) (1LSB to 31G) ratio | 192 bits (24 byte) |
| 2D LSBG (3 bpp) (1LSB to 63G) ratio | 48 bits (6 byte) |
| 2D LSBG (3 bpp) (1LSB to 127G) ratio | 12 bits (1.5 byte) |

**Table 3.7:** Max capacity of LSB (1 bpp) and LSBG (1 bpp) for a cover image with the size of 512x512 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (1 bpp) | 262,144 bits (32,768 byte) |
| LSBG (1 bpp) (1LSB to 1G) ratio | 131,072 bits (16,384 byte) |
| LSBG (1 bpp) (1LSB to 3G) ratio | 65,536 bits (8,192 byte) |
| LSBG (1 bpp) (1LSB to 7G) ratio | 32,768 bits (4,096 byte) |
| LSBG (1 bpp) (1LSB to 15G) ratio | 16,384 bits (2,048 byte) |
| LSBG (1 bpp) (1LSB to 31G) ratio | 8,192 bits (1,024 byte) |
| LSBG (1 bpp) (1LSB to 63G) ratio | 4,096 bits (512 byte) |
| LSBG (1 bpp) (1LSB to 127G) ratio | 2,048 bits (256 byte) |
| LSBG (1 bpp) (1LSB to 255G) ratio | 1,024 bits (128 byte) |

**Table 3.8:** Max capacity of LSB (1 bpp) and 2D LSBG (1 bpp) for a cover image with the size of 512x512 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (1 bpp) | 262,144 bits (32,768 byte) |
| 2D LSBG (1 bpp) (1LSB to 1G) ratio | 65,536 bits (8,192 byte) |
| 2D LSBG (1 bpp) (1LSB to 3G) ratio | 16,384 bits (2,048 byte) |

| | |
|---|---|
| 2D LSBG (1 bpp) (1LSB to 7G) ratio | 4,096 bits (512 byte) |
| 2D LSBG (1 bpp) (1LSB to 15G) ratio | 1,024 bits (128 byte) |
| 2D LSBG (1 bpp) (1LSB to 31G) ratio | 256 bits (32 byte) |
| 2D LSBG (1 bpp) (1LSB to 63G) ratio | 64 bits (8 byte) |
| 2D LSBG (1 bpp) (1LSB to 127G) ratio | 16 bits (2 byte) |
| 2D LSBG (1 bpp) (1LSB to 255G) ratio | 4 bits |

**Table 3.9:** Max capacity of LSB (2 bpp) and LSBG (2 bpp) for a cover image with the size of 512x512 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (2 bpp) | 524,288 bits (65,536 byte) |
| LSBG (2 bpp) (1LSB to 1G) ratio | 262,144 bits (32,768 byte) |
| LSBG (2 bpp) (1LSB to 3G) ratio | 131,072 bits (16,384 byte) |
| LSBG (2 bpp) (1LSB to 7G) ratio | 65,536 bits (8,192 byte) |
| LSBG (2 bpp) (1LSB to 15G) ratio | 32,768 bits (4,096 byte) |
| LSBG (2 bpp) (1LSB to 31G) ratio | 16,384 bits (2,048 byte) |
| LSBG (2 bpp) (1LSB to 63G) ratio | 8,192 bits (1,024 byte) |
| LSBG (2 bpp) (1LSB to 127G) ratio | 4,096 bits (512 byte) |
| LSBG (2 bpp) (1LSB to 255G) ratio | 2,048 bits (256 byte) |

**Table 3.10:** Max capacity of LSB (2 bpp) and 2D LSBG (2 bpp) for a cover image with the size of 512x512 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (2 bpp) | 524,288 bits (65,536 byte) |
| 2D LSBG (2 bpp) (1LSB to 1G) ratio | 131,072 bits (16,384 byte) |

| | |
|---|---|
| 2D LSBG (2 bpp) (1LSB to 3G) ratio | 32,768 bits (4,096 byte) |
| 2D LSBG (2 bpp) (1LSB to 7G) ratio | 8,192 bits (1,024 byte) |
| 2D LSBG (2 bpp) (1LSB to 15G) ratio | 2,048 bits (256 byte) |
| 2D LSBG (2 bpp) (1LSB to 31G) ratio | 512 bits (64 byte) |
| 2D LSBG (2 bpp) (1LSB to 63G) ratio | 128 bits (16 byte) |
| 2D LSBG (2 bpp) (1LSB to 127G) ratio | 32 bits (4 byte) |
| 2D LSBG (2 bpp) (1LSB to 255G) ratio | 8 bits (1 byte) |

**Table 3.11:** Max capacity of LSB (3 bpp) and LSBG (3 bpp) for a cover image with the size of 512x512 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (3 bpp) | 786,432 bits (98,304 byte) |
| LSBG (3 bpp) (1LSB to 1G) ratio | 393,216 bits (49,152 byte) |
| LSBG (3 bpp) (1LSB to 3G) ratio | 196,608 bits (24,567 byte) |
| LSBG (3 bpp) (1LSB to 7G) ratio | 98,304 bits (12,288 byte) |
| LSBG (3 bpp) (1LSB to 15G) ratio | 49,152 bits (6,144 byte) |
| LSBG (3 bpp) (1LSB to 31G) ratio | 24,576 bits (3,072 byte) |
| LSBG (3 bpp) (1LSB to 63G) ratio | 12,288 bits (1,536 byte) |
| LSBG (3 bpp) (1LSB to 127G) ratio | 6,144 bits (768 byte) |
| LSBG (3 bpp) (1LSB to 255G) ratio | 3,072 bits (384 byte) |

**Table 3.12:** Max capacity of LSB (3 bpp) and 2D LSBG (3 bpp) for a cover image with the size of 512x512 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (3 bpp) | 786,432 bits (98,304 byte) |

| | |
|---|---|
| 2D LSBG (3 bpp) (1LSB to 1G) ratio | 196,608 bits (24,567 byte) |
| 2D LSBG (3 bpp) (1LSB to 3G) ratio | 49,152 bits (6,144 byte) |
| 2D LSBG (3 bpp) (1LSB to 7G) ratio | 12,228 bits (1,536 byte) |
| 2D LSBG (3 bpp) (1LSB to 15G) ratio | 3,072 bits (384 byte) |
| 2D LSBG (3 bpp) (1LSB to 31G) ratio | 768 bits (96 byte) |
| 2D LSBG (3 bpp) (1LSB to 63G) ratio | 192 bits (24 byte) |
| 2D LSBG (3 bpp) (1LSB to 127G) ratio | 48 bits (6 byte) |
| 2D LSBG (3 bpp) (1LSB to 255G) ratio | 12 bits (1.5 byte) |

**Table 3.13:** Max capacity of LSB (1 bpp) and LSBG (1 bpp) for a cover image with the size of 1024x1024 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (1 bpp) | 1,048,576 bits (131,072 byte) |
| LSBG (1 bpp) (1LSB to 1G) ratio | 524,288 bits (65,536 byte) |
| LSBG (1 bpp) (1LSB to 3G) ratio | 262,144 bits (32,768 byte) |
| LSBG (1 bpp) (1LSB to 7G) ratio | 131,072 bits (16,384 byte) |
| LSBG (1 bpp) (1LSB to 15G) ratio | 65,536 bits (8,192 byte) |
| LSBG (1 bpp) (1LSB to 31G) ratio | 32,768 bits (4,096 byte) |
| LSBG (1 bpp) (1LSB to 63G) ratio | 16,384 bits (2,048 byte) |
| LSBG (1 bpp) (1LSB to 127G) ratio | 8,192 bits (1,024 byte) |
| LSBG (1 bpp) (1LSB to 255G) ratio | 4,096 bits (512 byte) |
| LSBG (1 bpp) (1LSB to 511G) ratio | 2,048 bits (256 byte) |

**Table 3.14:** Max capacity of LSB (1 bpp) and 2D LSBG (1 bpp) for a cover image with the size of 1024x1024 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (1 bpp) | 1,048,576 bits (131,072 byte) |
| 2D LSBG (1 bpp) (1LSB to 1G) ratio | 262,144 bits (32,768 byte) |
| 2D LSBG (1 bpp) (1LSB to 3G) ratio | 65,536 bits (8,192 byte) |
| 2D LSBG (1 bpp) (1LSB to 7G) ratio | 16,384 bits (2,048 byte) |
| 2D LSBG (1 bpp) (1LSB to 15G) ratio | 4,096 bits (512 byte) |
| 2D LSBG (1 bpp) (1LSB to 31G) ratio | 1,024 bits (128 byte) |
| 2D LSBG (1 bpp) (1LSB to 63G) ratio | 256 bits (32 byte) |
| 2D LSBG (1 bpp) (1LSB to 127G) ratio | 64 bits (8 byte) |
| 2D LSBG (1 bpp) (1LSB to 255G) ratio | 16 bits (2 byte) |
| 2D LSBG (1 bpp) (1LSB to 511G) ratio | 4 bits |

**Table 3.15:** Max capacity of LSB (2 bpp) and LSBG (2 bpp) for a cover image with the size of 1024x1024 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (2 bpp) | 2,097,152 bits (262,144 byte) |
| LSBG (2 bpp) (1LSB to 1G) ratio | 1,048,576 bits (131,072 byte) |
| LSBG (2 bpp) (1LSB to 3G) ratio | 524,288 bits (65,536 byte) |
| LSBG (2 bpp) (1LSB to 7G) ratio | 262,144 bits (32,768 byte) |
| LSBG (2 bpp) (1LSB to 15G) ratio | 131,072 bits (16,384 byte) |
| LSBG (2 bpp) (1LSB to 31G) ratio | 65,536 bits (8,192 byte) |
| LSBG (2 bpp) (1LSB to 63G) ratio | 32,768 bits (4,096 byte) |
| LSBG (2 bpp) (1LSB to 127G) ratio | 16,384 bits (2,048 byte) |

| | |
|---|---|
| LSBG (2 bpp) (1LSB to 255G) ratio | 8,192 bits (1,024 byte) |
| LSBG (2 bpp) (1LSB to 511G) ratio | 4,096 bits (512 byte) |

**Table 3.16:** Max capacity of LSB (2 bpp) and 2D LSBG (2 bpp) for a cover image with the size of 1024x1024 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (2 bpp) | 2,097,152 bits (262,144 byte) |
| 2D LSBG (2 bpp) (1LSB to 1G) ratio | 524,288 bits (65,536 byte) |
| 2D LSBG (2 bpp) (1LSB to 3G) ratio | 131,072 bits (16,384 byte) |
| 2D LSBG (2 bpp) (1LSB to 7G) ratio | 32,768 bits (4,096 byte) |
| 2D LSBG (2 bpp) (1LSB to 15G) ratio | 8,192 bits (1,024 byte) |
| 2D LSBG (2 bpp) (1LSB to 31G) ratio | 2,048 bits (256 byte) |
| 2D LSBG (2 bpp) (1LSB to 63G) ratio | 512 bits (64 byte) |
| 2D LSBG (2 bpp) (1LSB to 127G) ratio | 128 bits (16 byte) |
| 2D LSBG (2 bpp) (1LSB to 255G) ratio | 32 bits (4 byte) |
| 2D LSBG (2 bpp) (1LSB to 511G) ratio | 8 bits (1 byte) |

**Table 3.17:** Max capacity of LSB (3 bpp) and LSBG (3 bpp) for a cover image with the size of 1024x1024 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (3 bpp) | 3,145,728 bits (393,216 byte) |
| LSBG (3 bpp) (1LSB to 1G) ratio | 1,572,864 bits (196,608 byte) |
| LSBG (3 bpp) (1LSB to 3G) ratio | 786,432 bits (98,304 byte) |
| LSBG (3 bpp) (1LSB to 7G) ratio | 393,216 bits (49,152 byte) |
| LSBG (3 bpp) (1LSB to 15G) ratio | 196,608 bits (24,567 byte) |

| | |
|---|---|
| LSBG (3 bpp) (1LSB to 31G) ratio | 98,304 bits (12,288 byte) |
| LSBG (3 bpp) (1LSB to 63G) ratio | 49,152 bits (6,144 byte) |
| LSBG (3 bpp) (1LSB to 127G) ratio | 24,576 bits (3,072 byte) |
| LSBG (3 bpp) (1LSB to 255G) ratio | 12,288 bits (1,536 byte) |
| LSBG (3 bpp) (1LSB to 511G) ratio | 6,144 bits (768 byte) |

**Table 3.18:** Max capacity of LSB (3 bpp) and 2D LSBG (3 bpp) for a cover image with the size of 1024x1024 and pixel capacity =1 byte.

| Method | Max capacity |
|---|---|
| LSB (3 bpp) | 3,145,728 bits (393,216 byte) |
| 2D LSBG (3 bpp) (1LSB to 1G) ratio | 786,432 bits (98,304 byte) |
| 2D LSBG (3 bpp) (1LSB to 3G) ratio | 196,608 bits (24,567 byte) |
| 2D LSBG (3 bpp) (1LSB to 7G) ratio | 49,152 bits (6,144 byte) |
| 2D LSBG (3 bpp) (1LSB to 15G) ratio | 12,228 bits (1,536 byte) |
| 2D LSBG (3 bpp) (1LSB to 31G) ratio | 3,072 bits (384 byte) |
| 2D LSBG (3 bpp) (1LSB to 63G) ratio | 768 bits (96 byte) |
| 2D LSBG (3 bpp) (1LSB to 127G) ratio | 192 bits (24 byte) |
| 2D LSBG (3 bpp) (1LSB to 255G) ratio | 48 bits (6 byte) |
| 2D LSBG (3 bpp) (1LSB to 511G) ratio | 12 bits (1.5 byte) |

# 4 EXPERIMENTS RESULTS

Thirteen experiments were conducted for the study of LSB method cases, different cases of LSBG and 2D LSBG methods used with different gapping rates (6 different rates), three cases selected cover image, and two cases of selected secret image. All of these cases are explained in details in the experiment data sets section.

## 4.1. Experiments introduction

The experiments were designed to be composed of three main parts for each single experiment. Each experiment is actually a two-sided experiment for all the experiments parts where two data sets are generated in each experiment since there are two different secret messages that were selected for the experiment or two different embedding methods were selected in a single experiment. The first part of the experiment presents the selected cover image, the selected secret images (one message of two), and the embedding methods used in the experiments (one method or two).

The Second part of each experiment is the figures part where it is composed of two main figures. The first main figure is composed of eight subfigures that show the input images and result images (stego image). The first two subfigures show the selected secret images. The second two subfigures show the cover images before steganography embedding. The third two subfigures present a demonstration figures with the same size of the stego images but pixels are with only to colors (white and black) to demonstrate clearly the all pixels that are used for the steganography embedding and how they are distributed in the image. The white pixels are actually the pixels that are selected for data embedding by the method used and the black pixels are the pixels that were not used for data embedding and can be considered as gap pixels in the LSBG method. The fourth and last two subfigures will present the resulted stego images after the secret data embedding.

The second main figure in the second part of experiments is composed of four subfigures. The first two subfigures are the histogram figures of the cover images before the data embedding. And the second two subfigures are the histogram figures of the stego images after the data embedding. The histogram figures are very important since they are used to evaluate initially the amount of difference between the cover images and the stego images. However, the main evaluation is to compare the histogram figures of the stego images of a group of experiments that have the cover image and secret image as inputs but used different embedding methods starting from basic LSB then to LSBG with different possible gapping rates. Studying and analyzing that group of histogram figure provide an indication of how LSBG method with different possible rate can improve the performance compared to LSB method by reducing the deviation in the stego histogram caused by the secret data embedding and distribute it in larger pixels values band of the histogram.

The third part of the Experiments will show the secret message capacity details and the amount of capacity needed from the cover image to carry the secret message completely after the data embedding. Selecting the amount of bits to be embedded in each pixel (bpp) will affect the number of pixels needed for data embedding. In addition, the last pixels location will be specified in this part and it is affected by the embedding method used and the gaping rate in the LSBG method.

The last part of this chapter contains the MSE-PSNR parameters table for comparison. The MSE and PSNR parameters were measured for all the experiment cases and the results can be found in Table 4.1. MSE & PSNR parameters are used as evaluation parameters of the amount of changes (errors) and their weight between the cover images and the stego images. In addition, the comparison between applying LSB and LSBG methods with different gapping rates will give the capability to evaluate the amount of possible improvement in the performance of the system by having MSE value to be reduced and PSNR value to be increased.

The main discussion, evaluation, and comparison between the experiments results and the different methods and cases used will take place in the following chapter in the discussion and conclusion part.

## 4.2. Data Sets Used In The Experiments

Two secret images were used in the experiments. These images are:

- Grayscale pepper image with size of 50x50 pixels.

- 50x50 white background image with fixed pixel value of 255 (the binary value of 11111111). This secret image were chosen as severe case as it causes  a high amount of deviation in the stego histogram which is important since applying different methods used in the experiments will show the great difference clearly in the stego histogram between each embedding method used.

Three cover images were used in the experiments. These images are:

- Grayscale watch image with the size of 256x256 pixels (TIFF).
- Cameraman image, grayscale image with size of 512x512 pixels (BMP).
- Baboon greyscale image, grayscale image with size of 512x512 pixels (PNG).

Steganography Embedding methods used in the Experiments:

- Least Significant Bit method (LSB) as the Baseline method (1 bpp), (2 bpp), & (3 bpp).
- Least Significant Bit Gaped method LSBG (2 bpp) (1LSB to 2G).
- LSBG (2 bpp) (1LSB to 3G).
- LSBG (2 bpp) (1LSB to 7G).
- LSBG (2 bpp) (1LSB to 15G).
- Shifted LSBG (2 bpp) (1LSB to 2G).
- 2D LSBG (2 bpp) (1LSB to 1G).
- 2D LSBG (2 bpp) (1LSB to 3G).
- LSBG (1 bpp) (1LSB to 1G).
- 2D LSBG (1 bpp) (1LSB to 2G).
- LSBG (3 bpp) (1LSB to 3G).
- LSBG (3 bpp) (1LSB to 7G).
- 2D LSBG (3 bpp) (1LSB to 2G).
- 2D LSBG (3 bpp) (1LSB to 3G).

### 4.3. Experiments 1

Cover image: grayscale watch image with the size of 256x256.

Secret Message: a) grayscale pepper image with size of 50x50.

b) 50x50 white background image with fixed pixel value of 255.

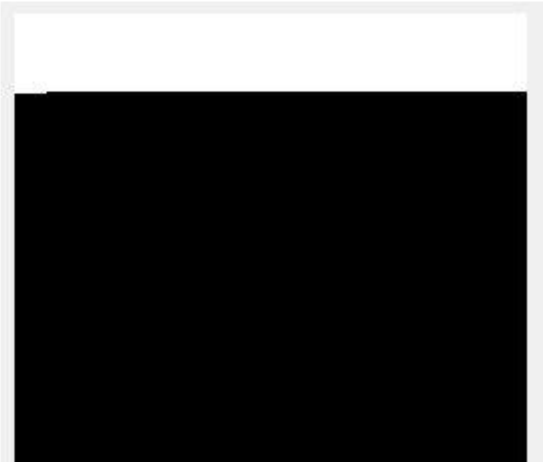
Method:  LSB (2 bpp).

(a)

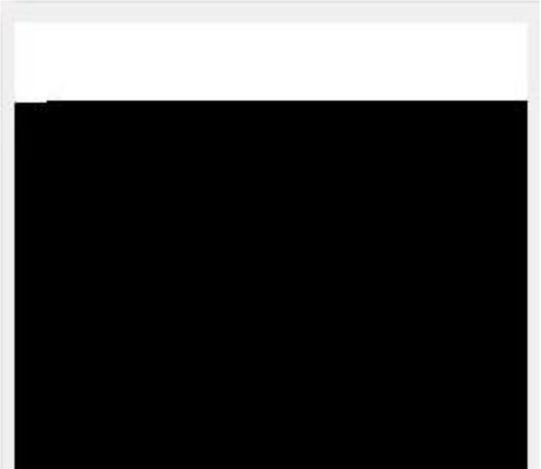
(b)

(c)

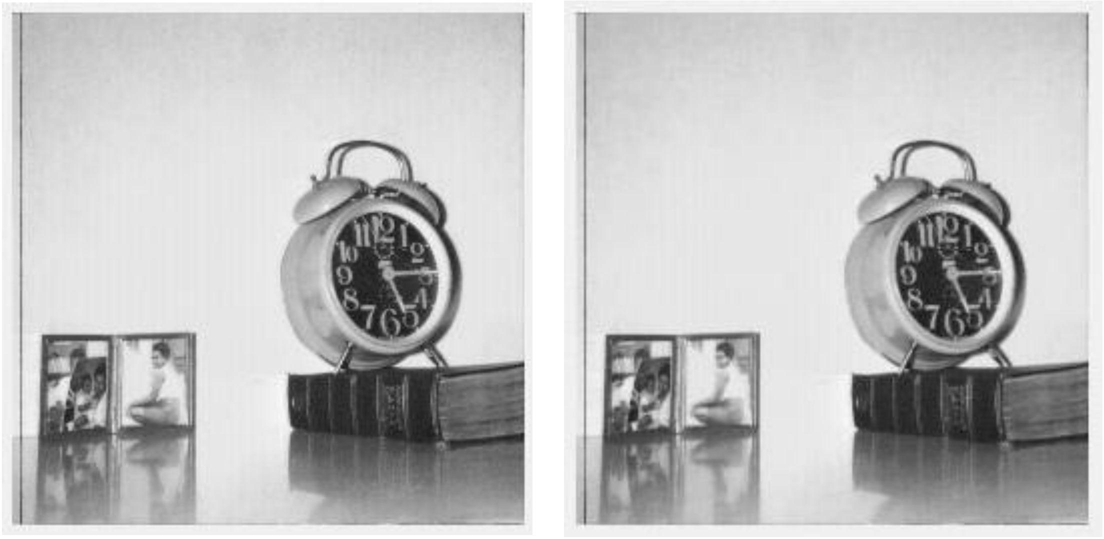
(d)



(e)

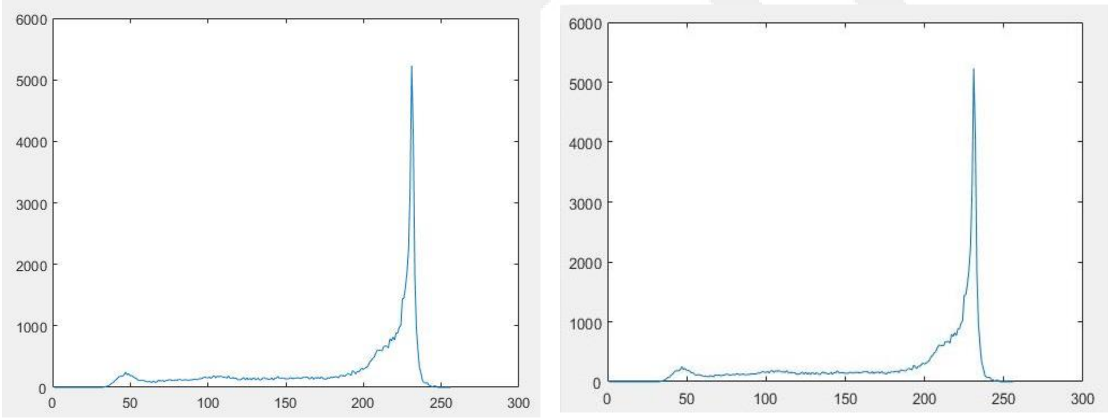


(f)

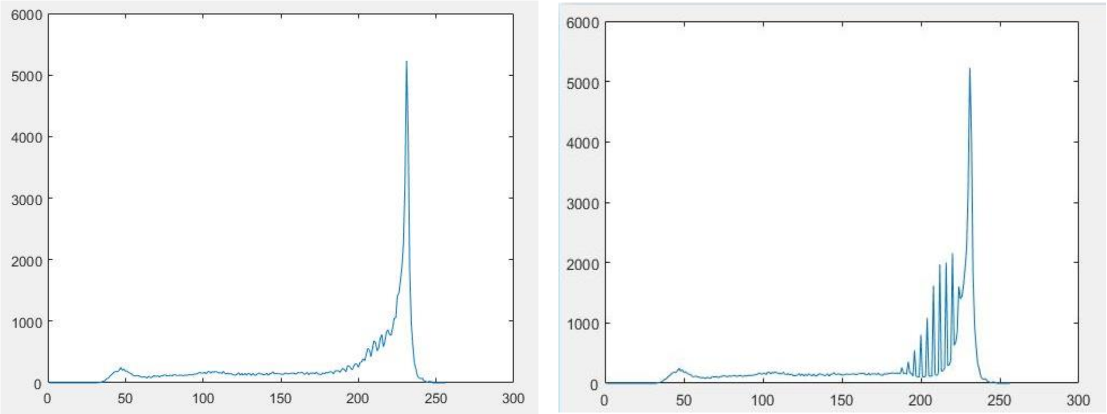


(g) (h)

**Figure 4.1:** (a) The secret image 1; (b) secret image 2; (c) and(d) cover images before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a) (b)



(c) (d)

**Figure 4.2:** Histogram of the images for(a) the cover image before LSB embedding of secret message 1; (b) cover image before LSB embedding of secret message 2; (c) stego image after LSB embedding of secret message 1; (d) stego image after LSB embedding of secret message 2.

The Secret message size is 50x50x8 bits, which requires a total of 20000 bits capacity in the cover image. We used LSB (2 bpp) embedding where 2 bits of secret is embedded inside each pixel of the cover image. By that, we need to have 10000 pixels to cover the capacity of the secret message. The Last pixel used for embedding is pixel the number 16 in row 40.

### 4.4. Experiments 2

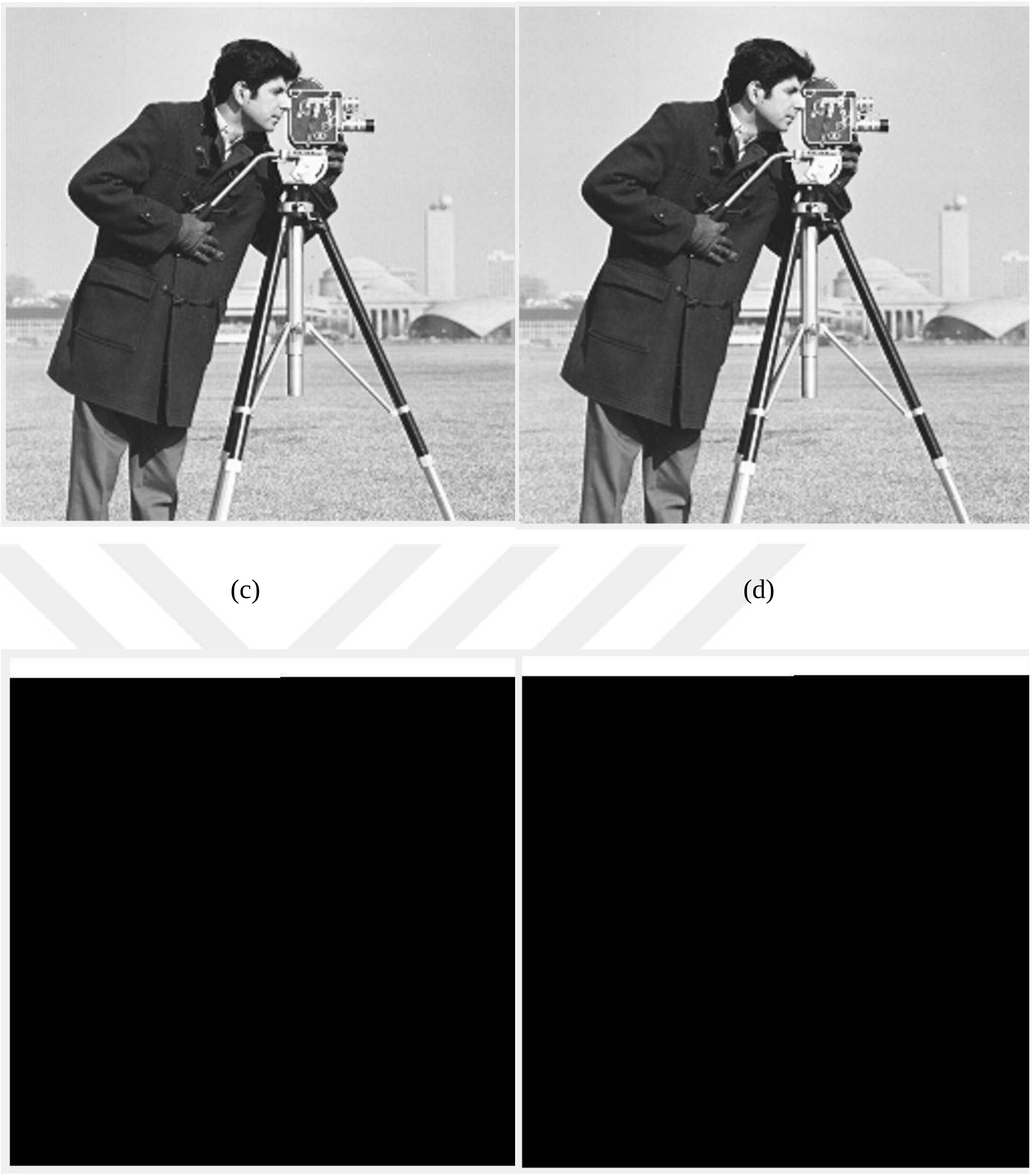
Cover image: Cameraman image, grayscale image with size of 512x512.

Secret Message: a) grayscale pepper image with size of 50x50.

b) 50x50 white background image with fixed pixel value of 255.

Method:  LSB (2 bpp).

(a) (b)



(c)

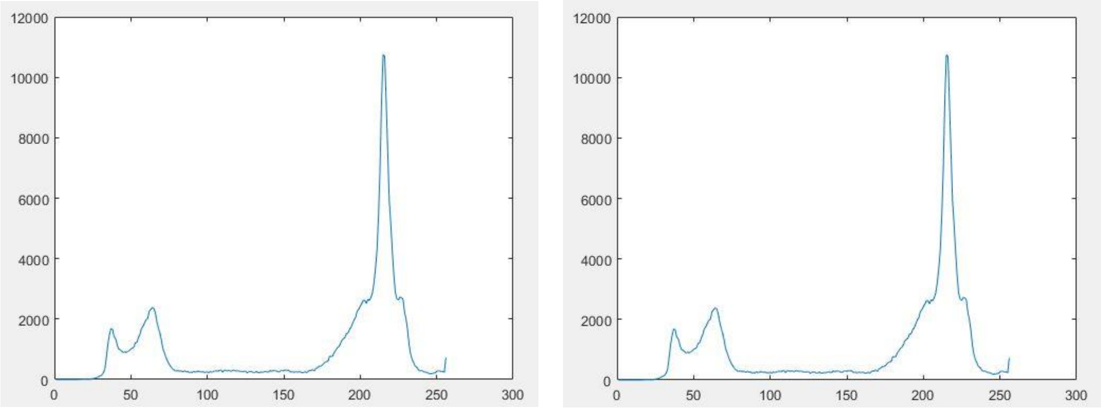
(d)

(e)

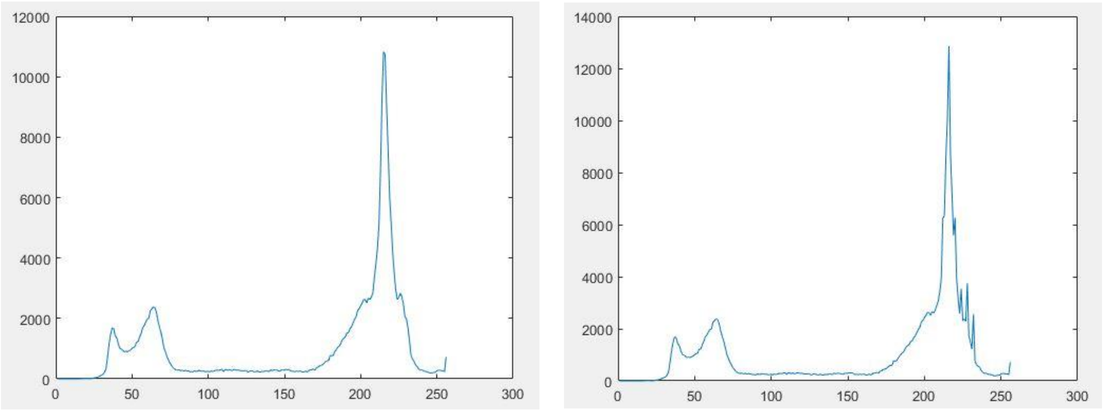
(f)

(g) (h)

**Figure 4.3:** (a) The secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a) (b)



(c) (d)

**Figure 4.4:** Histogram for (a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

The Secret message size is 50x50x8 bits, which is equal totally 20000 bits. Using LSB (2 bpp) will insert 2 bits inside each pixel. We need to have 10000 pixels to cover the secret message capacity. Last pixel to be used for embedding is pixel number 272 in row 20.

## 4.5. Experiments 3

Cover image: watch image, grayscale image with size of 256x256.

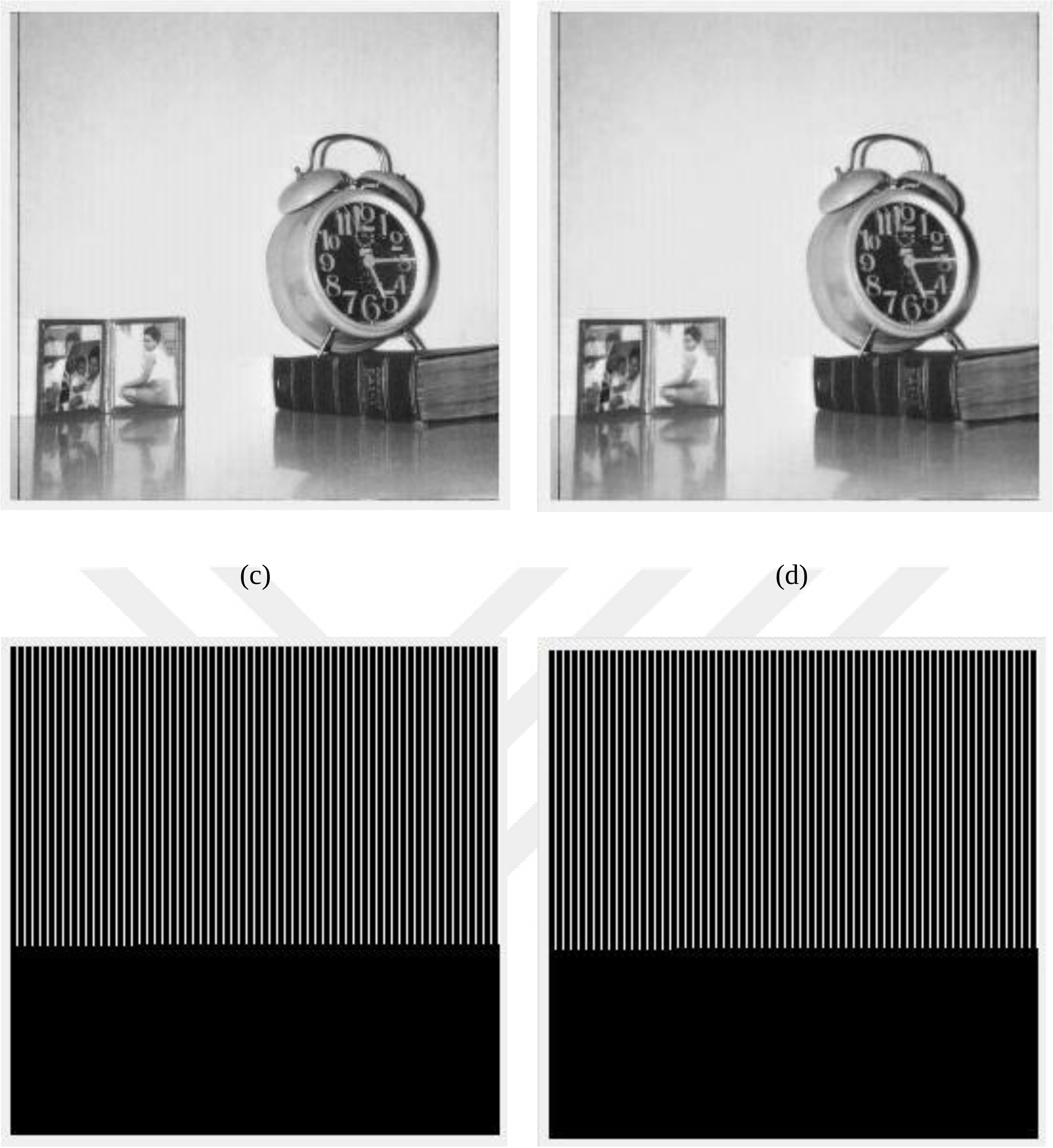
Secret Message: a) grayscale pepper image with size of 50x50.

b) (50x50) white background image with fixed pixel value of 255.

Method: LSBG (2 bits) (1LSB to 3G) ratio.

(a)

(b)

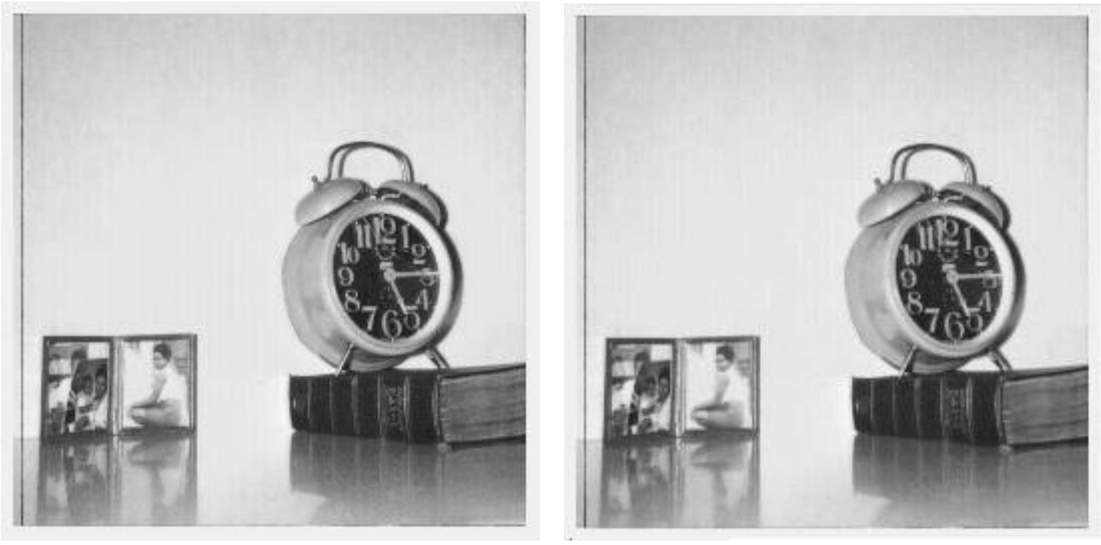


(c)

(d)

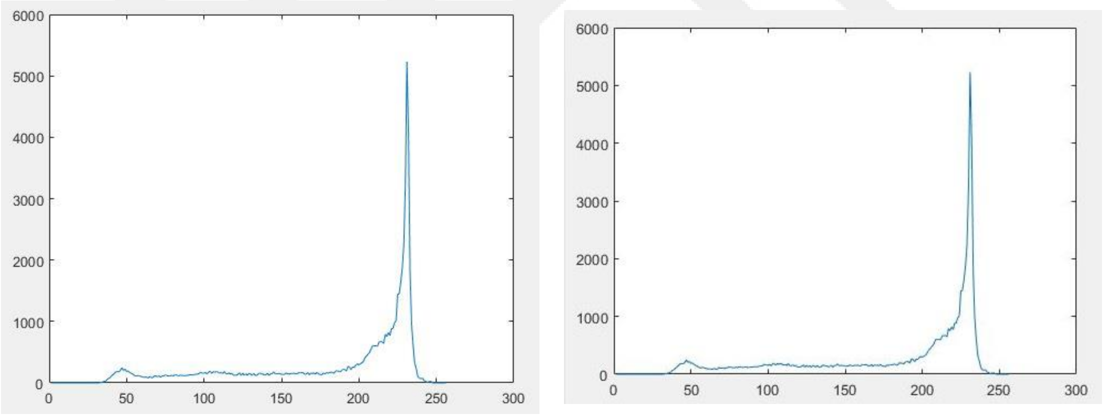
(e)

(f)

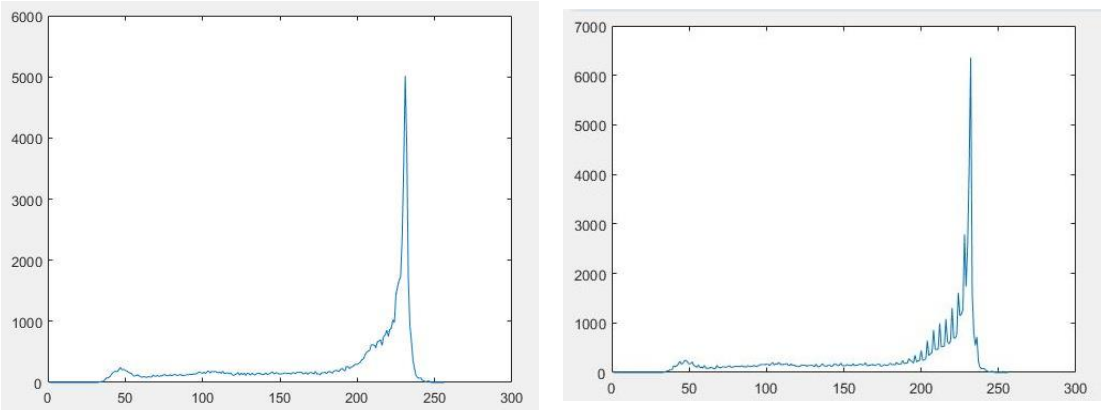


(g) (h)

**Figure 4.5:** (a) the secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a) (b)



(c) (d)

**Figure 4.6:** histogram for. (a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

The Secret message size is 50x50x8 bits, which is equal totally 20000 bits. Using LSB (2 bits) will insert 2 bits inside each pixel. We need to have 10000 pixels to cover the secret message capacity. Last pixel to be used for embedding is pixel number 64 row 157. In each row 64 pixels are used in such a sequence of pixel (4, 8, 12, 16, …. ,256 ).

## 4.6. Experiments 4

Cover image: Cameraman image, grayscale image with size of 512x512.

Secret Message: a) grayscale pepper image with size of 50x50.

b) 50x50 white background image with fixed pixel value of 255.

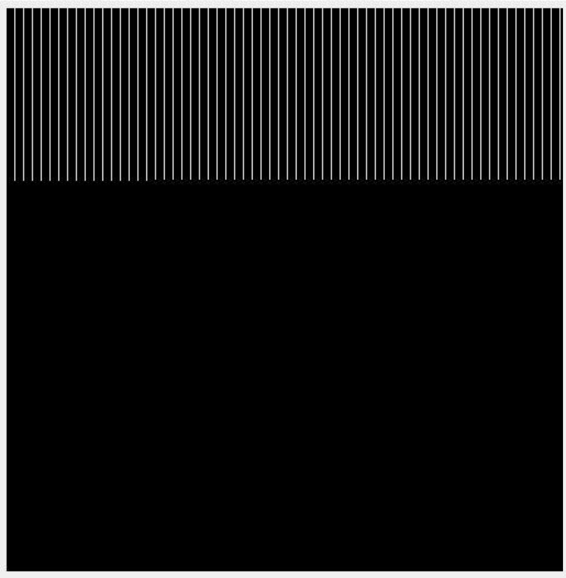
Method: LSBG (2 bpp) (1LSB to 7G).

(a)

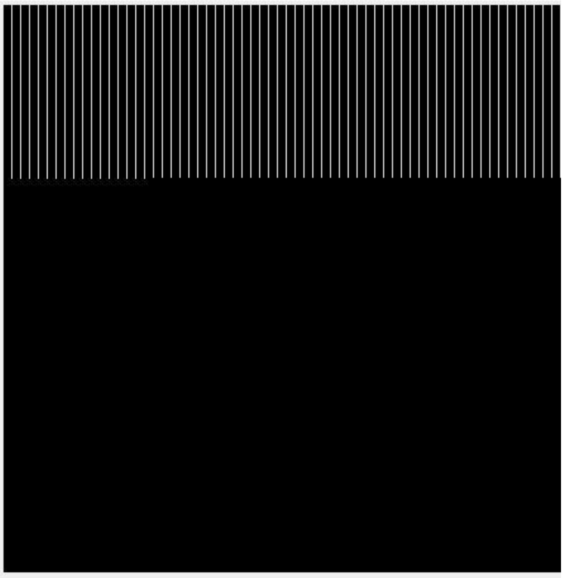
(b)

(c)

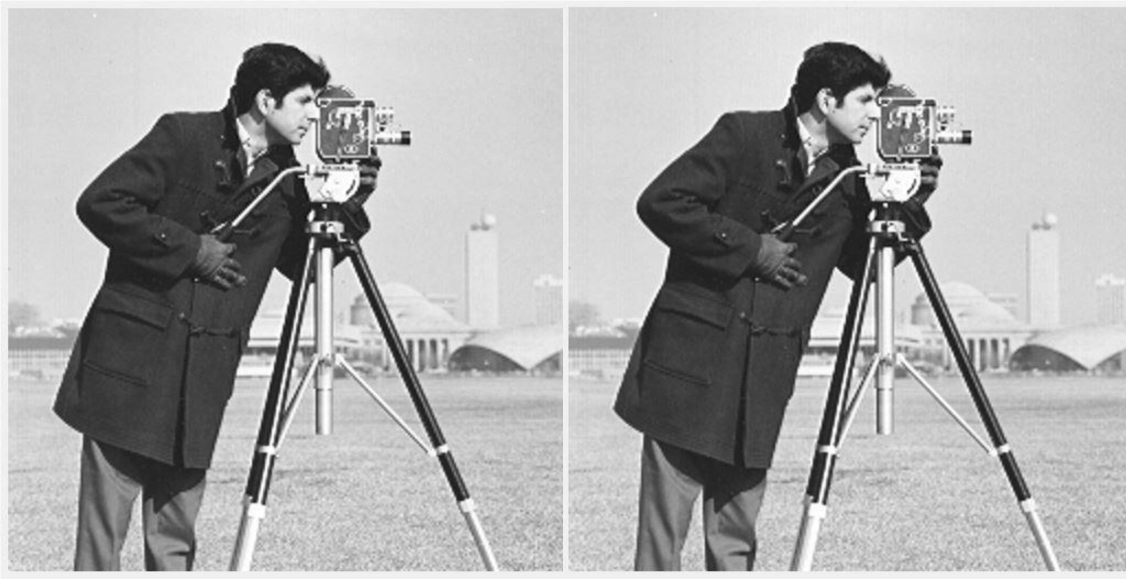
(d)



(e)

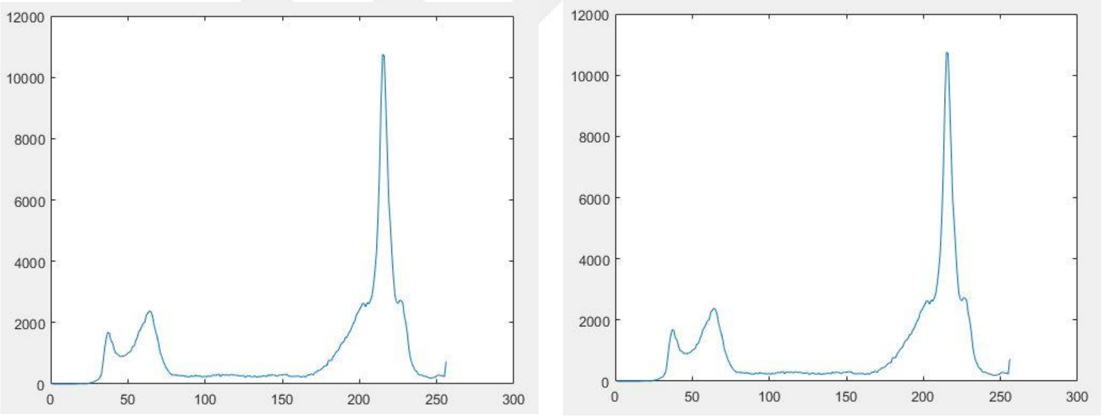


(f)

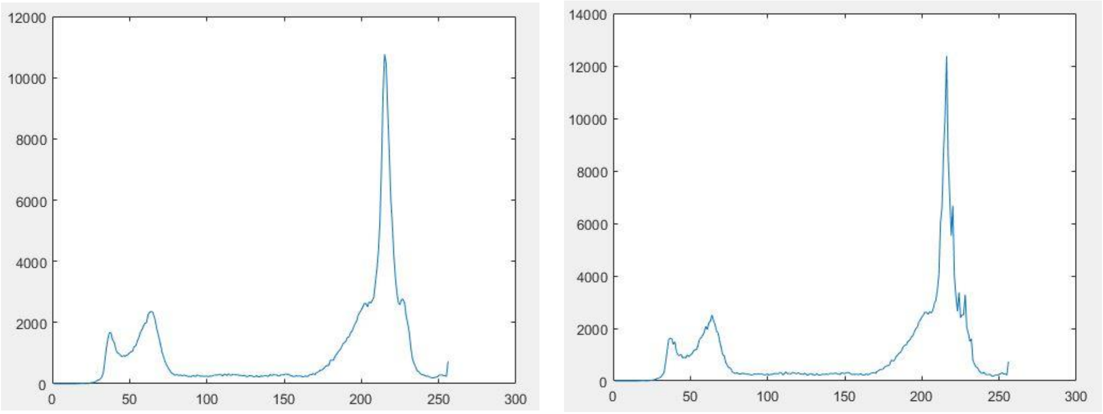


(g) (h)

**Figure 4.7:** (a) The secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a) (b)



(c) (d)

**Figure 4.8:** Histogram for (a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

The Secret message size is 50x50x8 bits, which is equal totally 20000 bits. Using LSB (2 bpp) will insert 2 bits inside each pixel. We need to have 10000 pixels to cover the secret message capacity. Last pixel to be used for embedding is pixel number 64 in row 316. In each row 32 pixels are used in such a sequence of pixel (8, 16, 24, 32, …. ,256 ).

### 4.7. Experiments 5

Cover image: watch image, grayscale image with size of 256x256.

Secret Message: a) grayscale pepper image with size of 50x50.

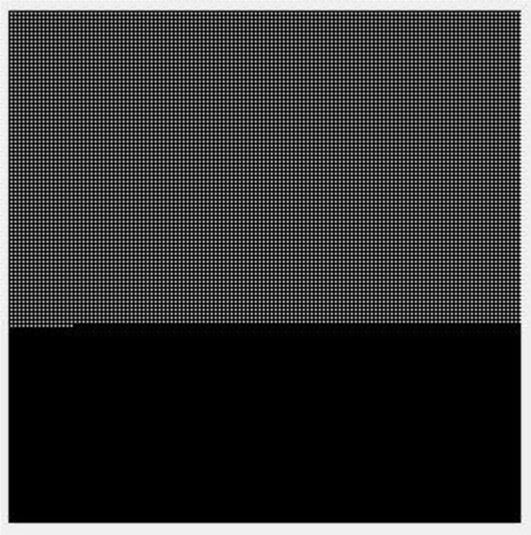
b) (50x50) white background image with fixed pixel value of 255.

Method: 2D LSBG (2 bpp) (1LSB to 1G) ratio.

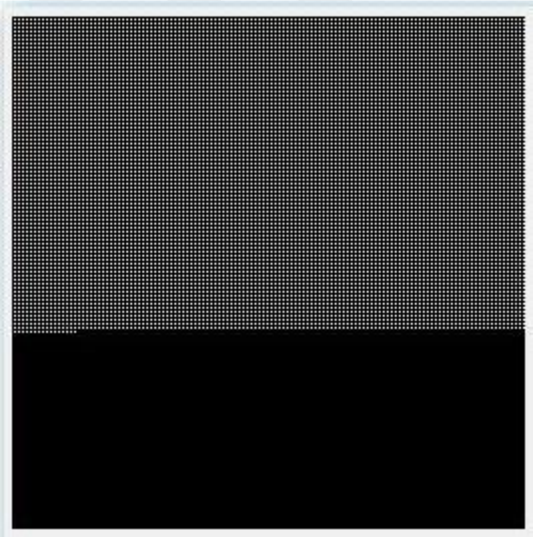
(a) (b)

(c)

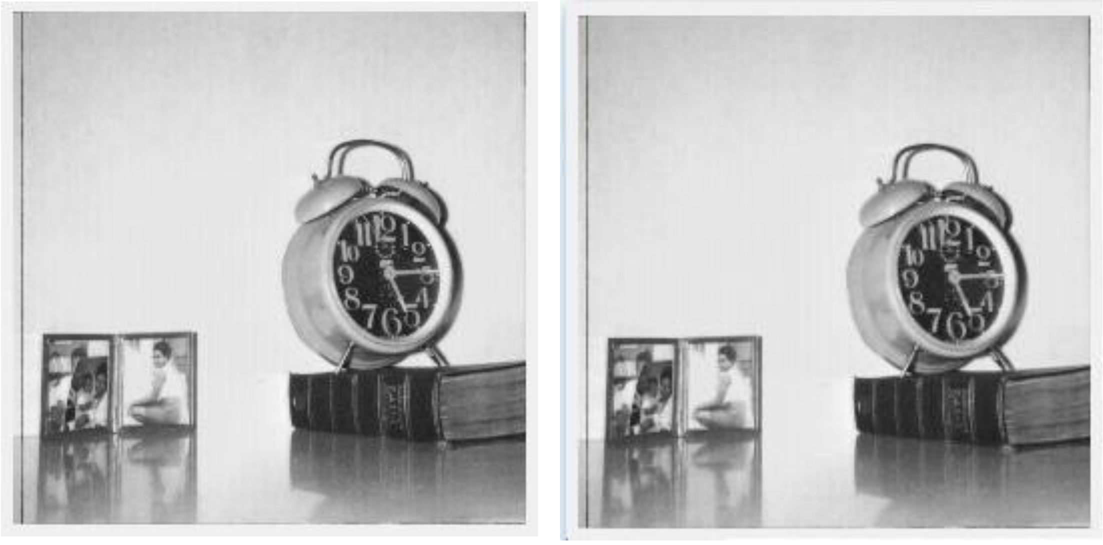
(d)



(e)

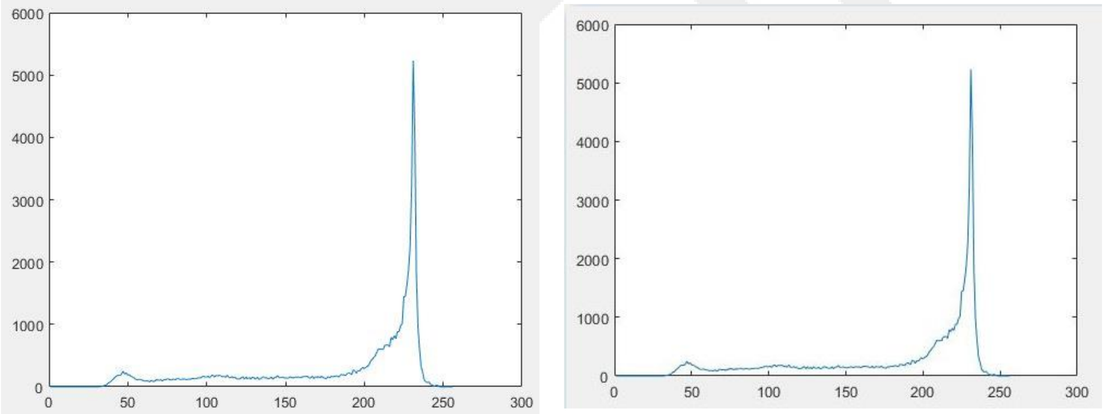


(f)

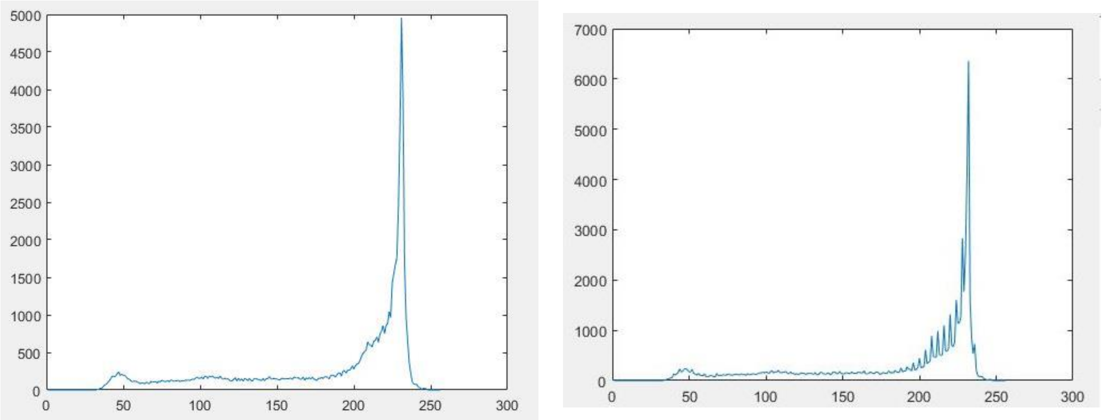


(g) (h)

**Figure 4.9:** (a) The secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a) (b)



(c) (d)

**Figure 4.10:** Histogram for(a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

The Secret message size is 50x50x8 bits, which is equal totally 20000 bits. Using LSB (2 bpp) will insert 2 bits inside each pixel. We need to have 10000 pixels to cover the secret message capacity. Last pixel to be used for embedding is pixel number 32 row 158. In each row and column 128 pixels can be used in a sequence as (2, 4, 6, 8, ... ,256).

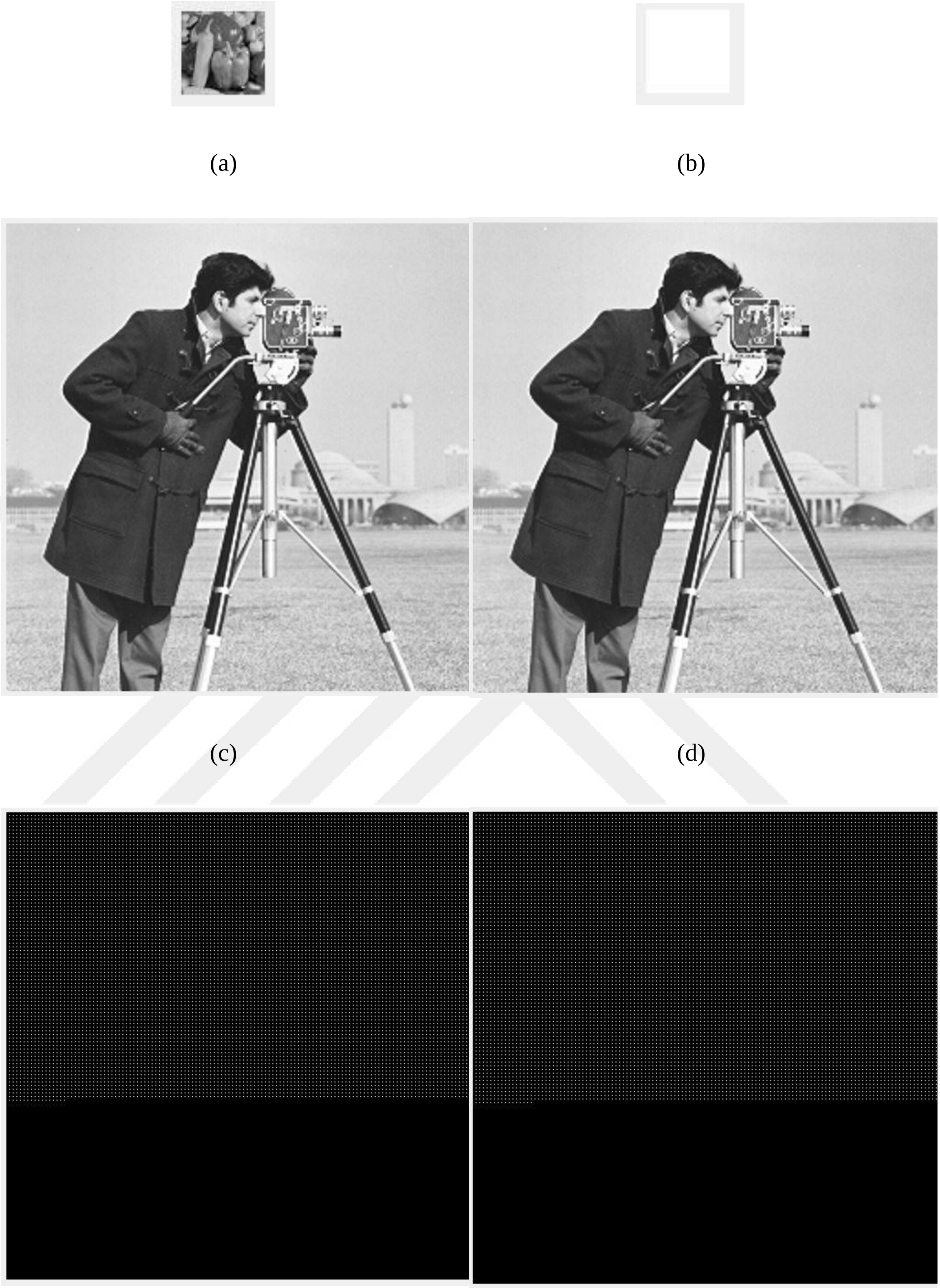
### 4.8. Experiments 6

Cover image: Cameraman image, grayscale image with size of 512x512.

Secret Message: a) grayscale pepper image with size of 50x50.

b) 50x50 white background image with fixed pixel value of 255.

Method: 2D LSBG (2 bpp) (1LSB to 3G) ratio.



(a)

(b)

(c)

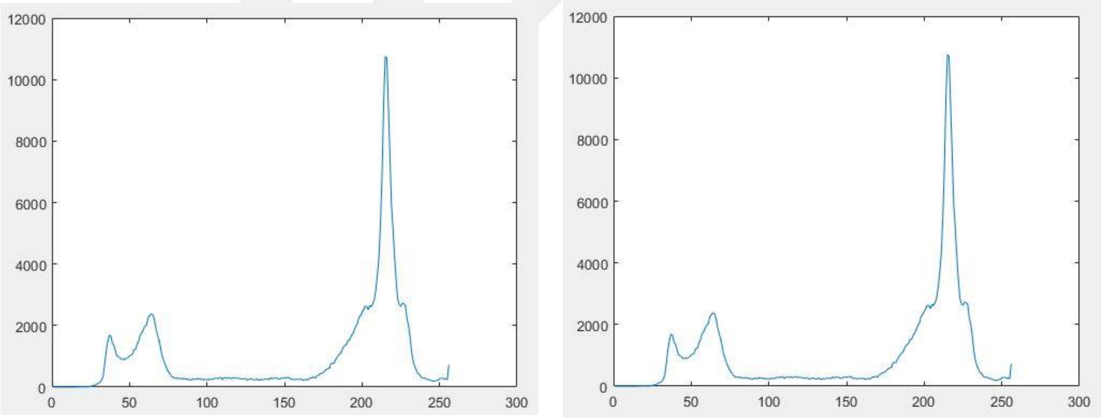
(d)

(e)

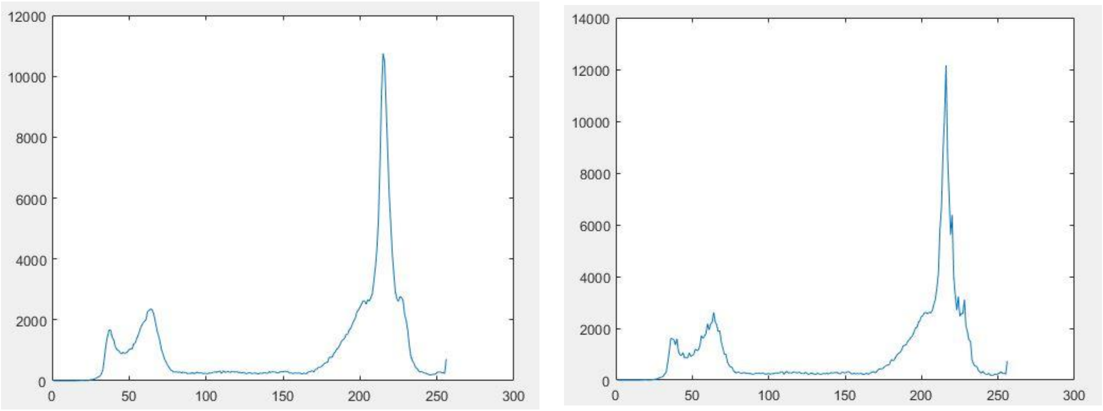
(f)

(g) (h)

**Figure 4.11:** (a) the secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a) (b)



(c) (d)

**Figure 4.12:** Histogram for(a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

The Secret message size is 50x50x8 bits, which is equal totally 20000 bits. Using LSB (2 bpp) will insert 2 bits inside each pixel. We need to have 10000 pixels to cover the secret message capacity. Last pixel to be used for embedding is pixel number 64 row 314. In each row and column 64 pixels are used in a sequence of (4, 8, 12, 16, …. ,256 ).

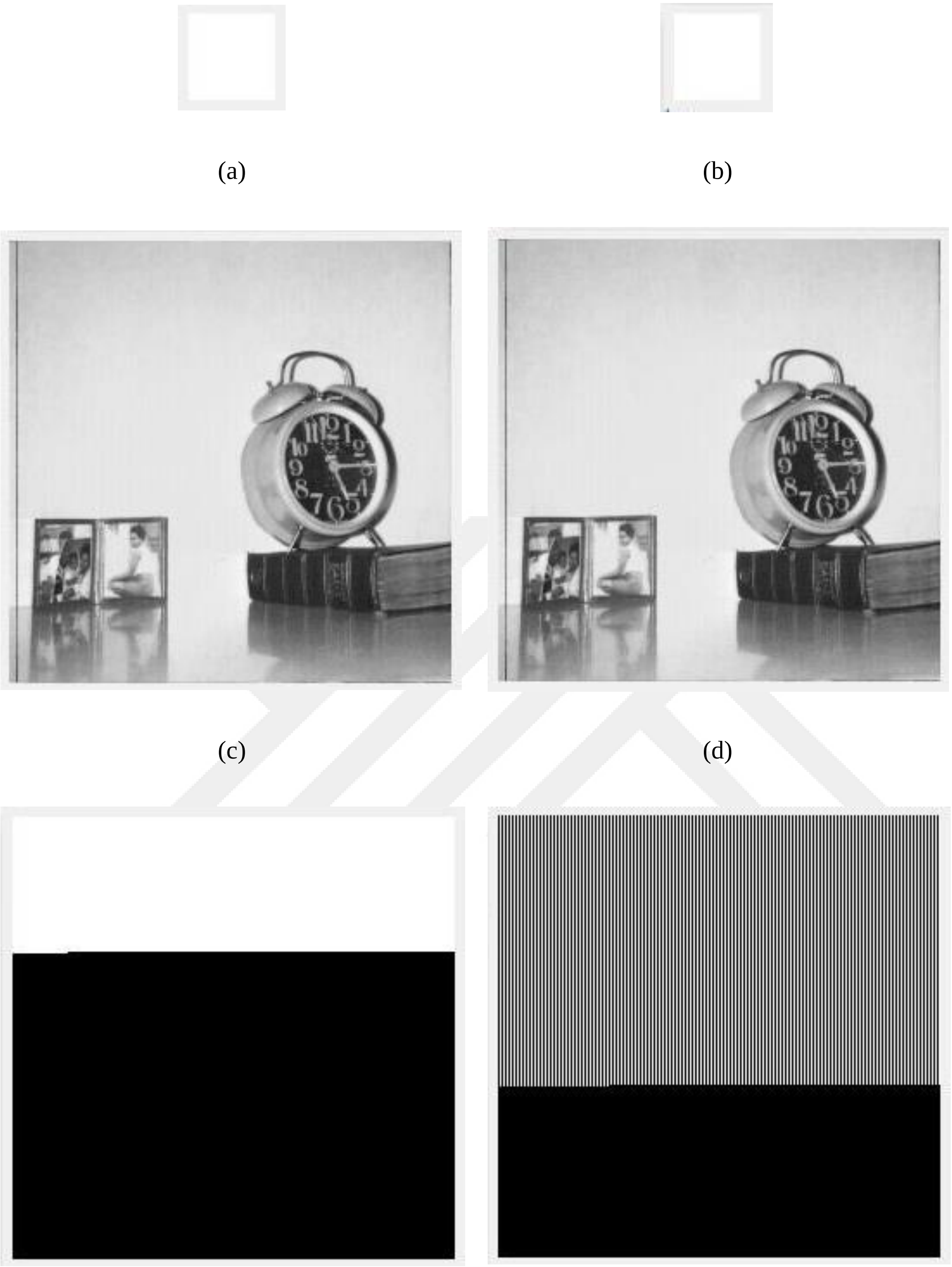
## 4.9. Experiments 7

Cover image: watch image, grayscale image with size of 256x256.

Secret Message: (50x50) white background image with fixed pixel value of 255.

Method: a) LSB (1 bpp).

b) LSBG (1 bpp) (1LSB to 1G).



(a)

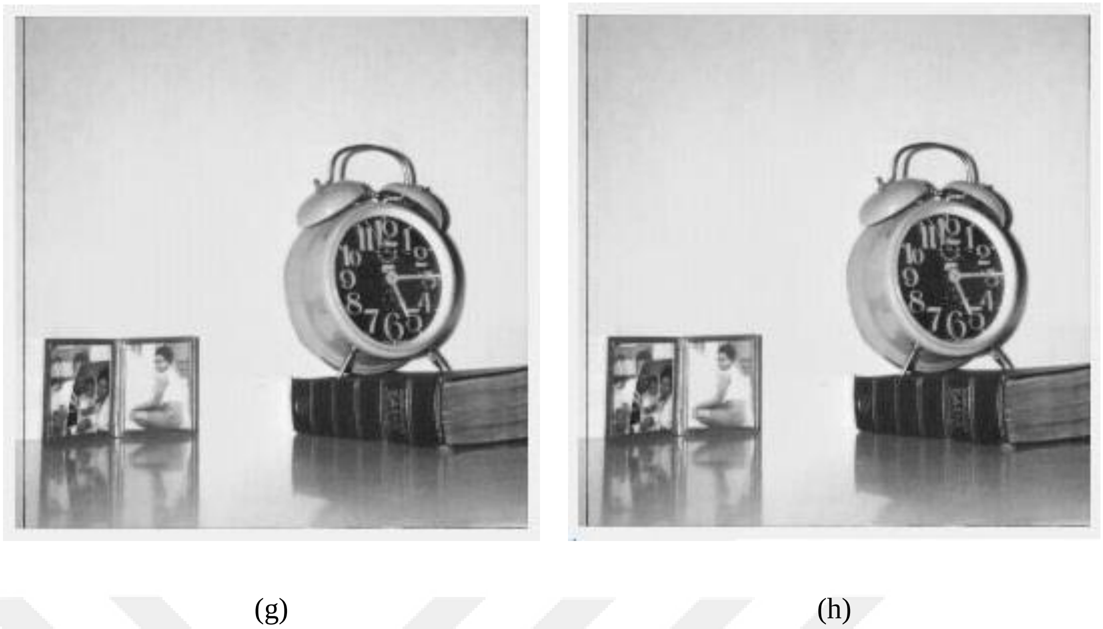
(b)

(c)

(d)

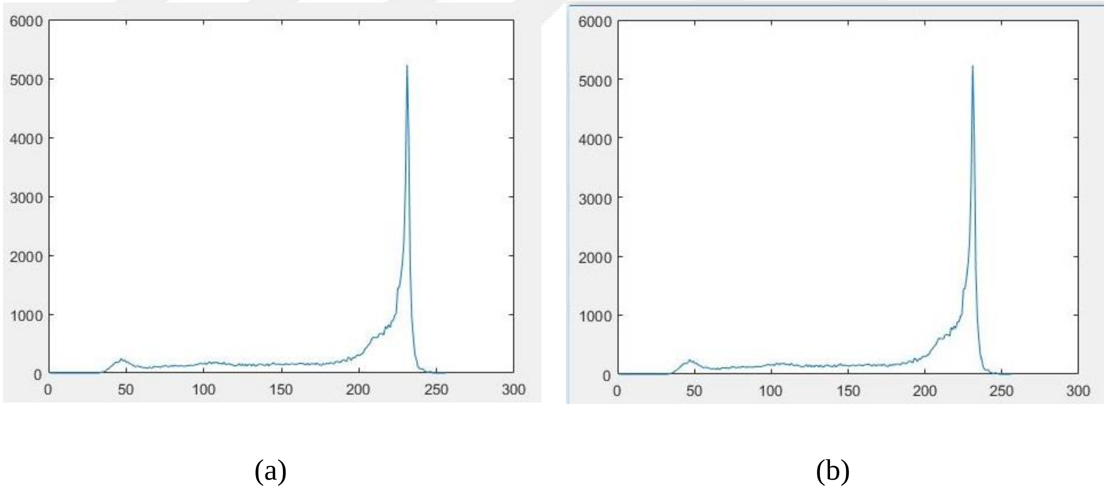
(e)

(f)

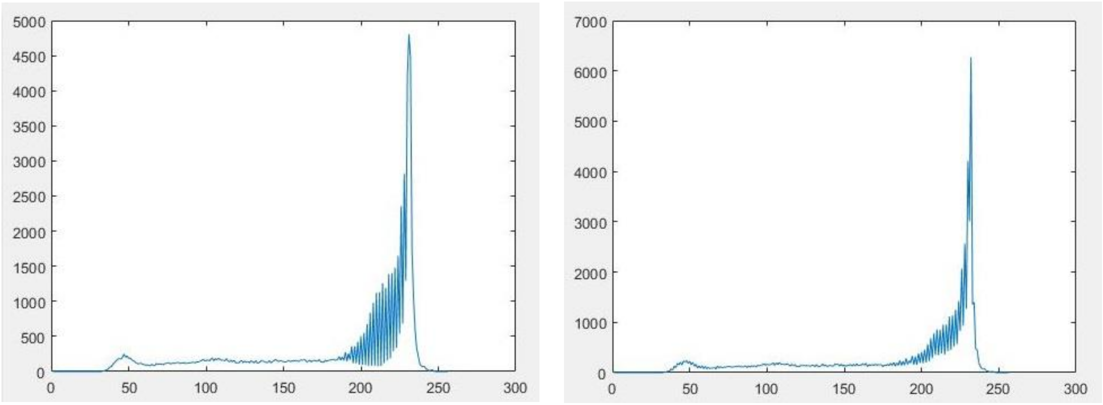


(g) (h)

**Figure 4.13:** (a) the secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a) (b)



(c) (d)

**Figure 4.14:** Histogram for(a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

The Secret message size is 50x50x8 bits, which is totally equal to 20000 bits. For embedding LSB (1 bit) is used where 1 bit is embedded inside each pixel. We need to have 20000 pixels for the embedding process. For LSB (1 bpp) method, the last pixel used is pixel number 32 in row 79. For LSBG (1 bpp) (1LSB to 1G) method, the last pixel used is pixel number 64 in row 157. (In each 256 size row 128 are used as a sequence of 2, 4, 6 ,8, ...,256.

### 4.10. Experiments 8

Cover image: Cameraman image, grayscale image with size of 512x512.

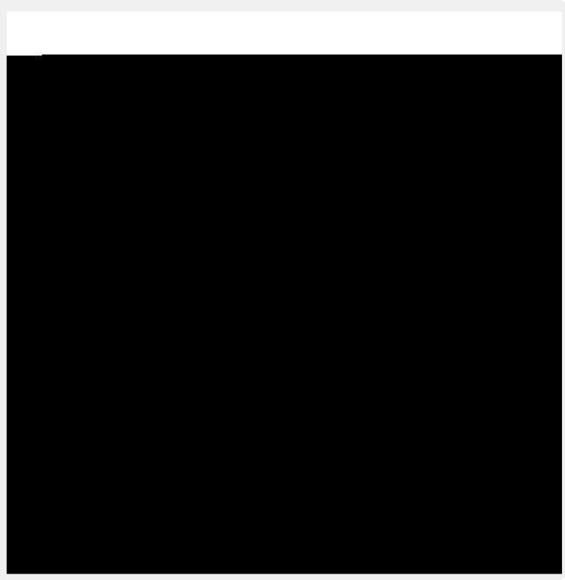
Secret Message: grayscale pepper image with size of 50x50.

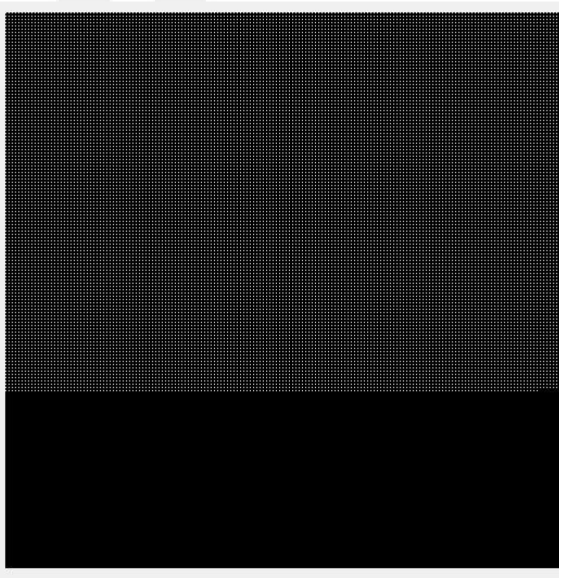
Method: a) LSB (1 bpp).

b) 2D LSBG (1 bpp) (1LSB to 2G) ratio

(a) (b)

(c) (d)



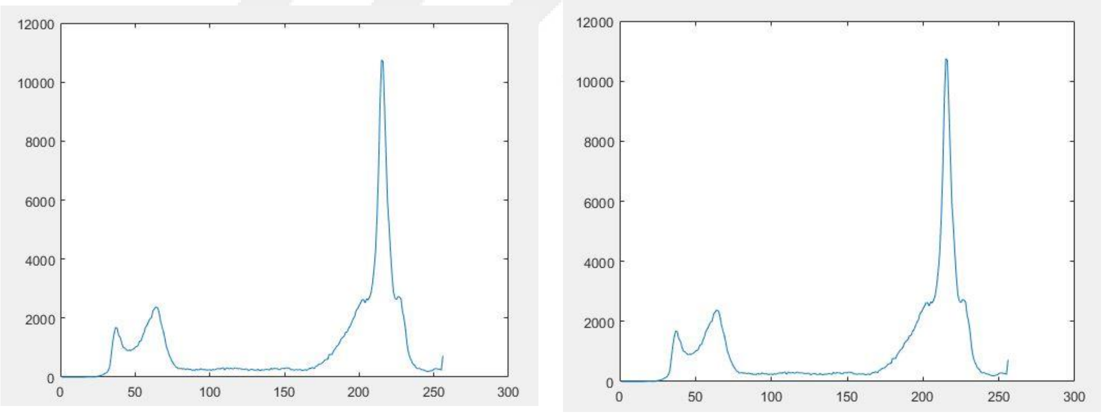


(e) (f)

(g)

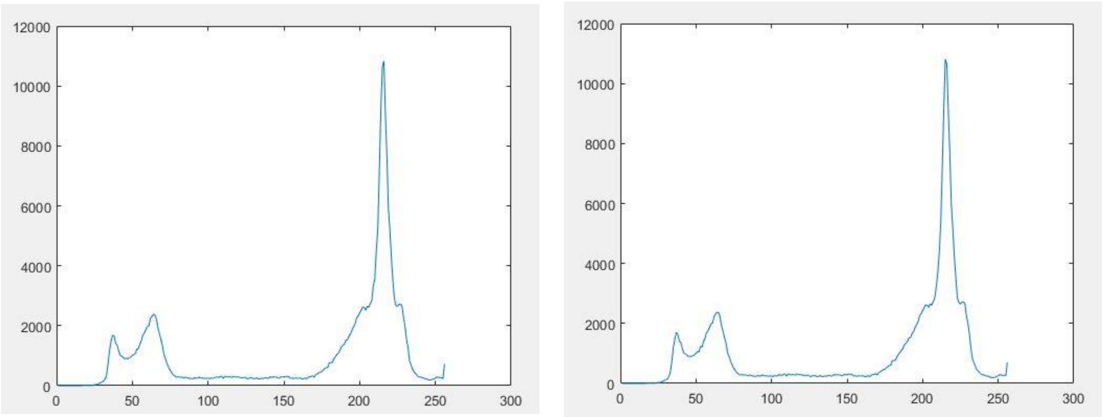
(h)

**Figure 4.15:** (a) the secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a)

(b)



(c) (d)

**Figure 4.16:** Histogram for(a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

The Secret message size is 50x50x8 bits, which is totally equal to 20000 bits. For embedding LSB (1 bit) is used where 1 bit is embedded inside each pixel. We need to have 20000 pixels for embedding process. For LSB (1 bpp) method, the last pixel used is pixel number 32 in row 40. For 2D LSBG (1 bpp) (1LSB to 2G) method, the last pixel used is pixel number 490 in row 349 (in each row or column 171 pixels can be used as a sequence of 1, 4, 7, 10, …. ,511.

## 4.11. Experiments 9

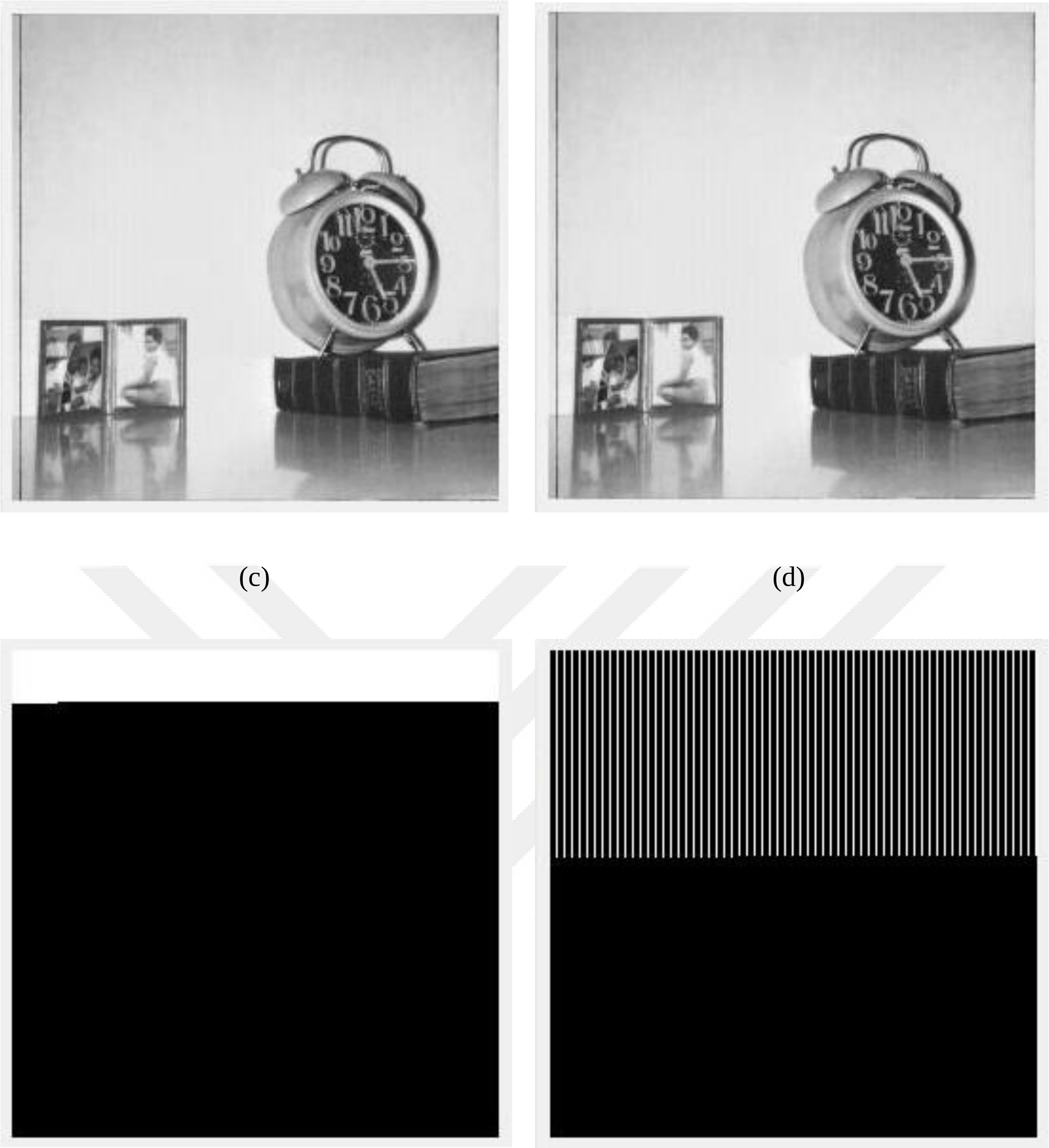
Cover image: watch image, grayscale image with size of 256x256.

Secret Message: 51x51 matrix with fixed pixel value of 255.

Method: a) LSB (3 bpp).

b) LSBG (3 bpp) (1LSB to 3G).

(a) (b)

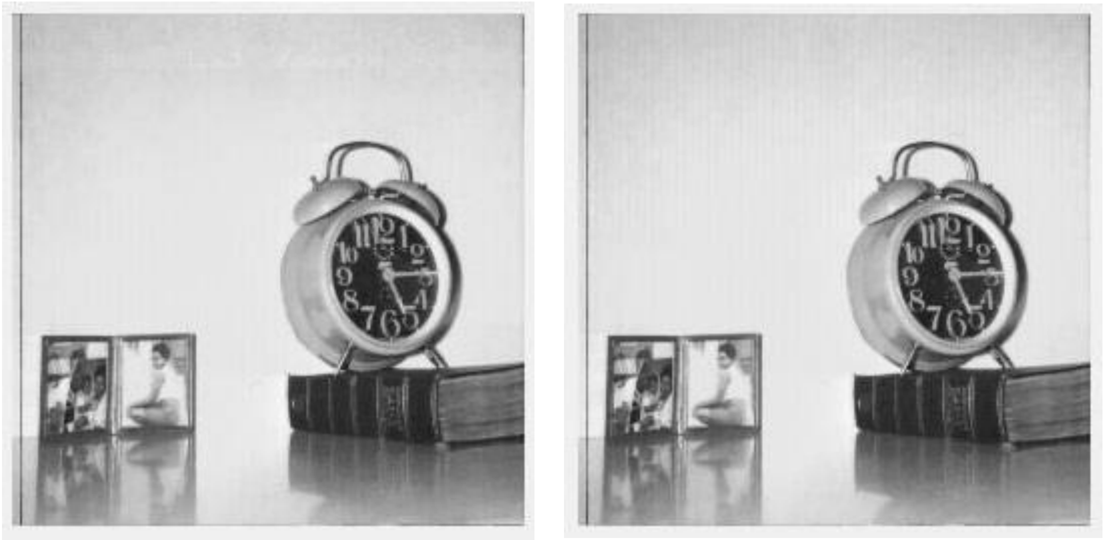


(c)

(d)

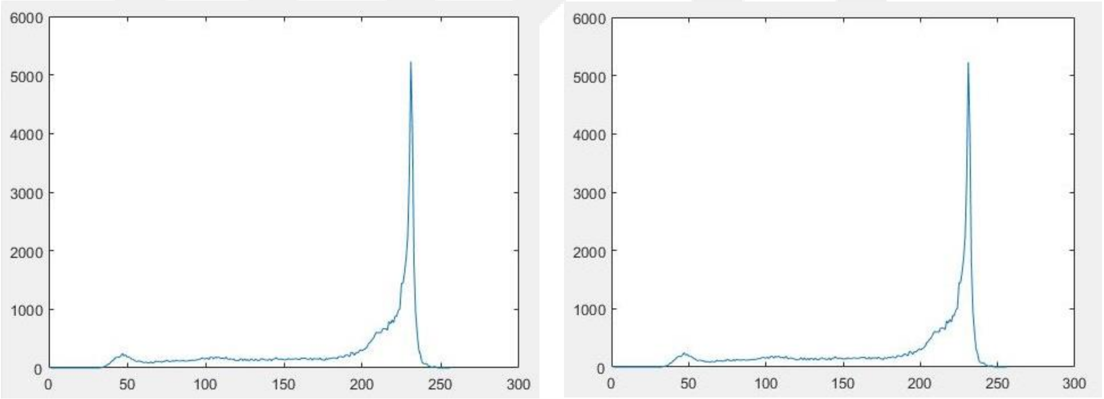
(e)

(f)

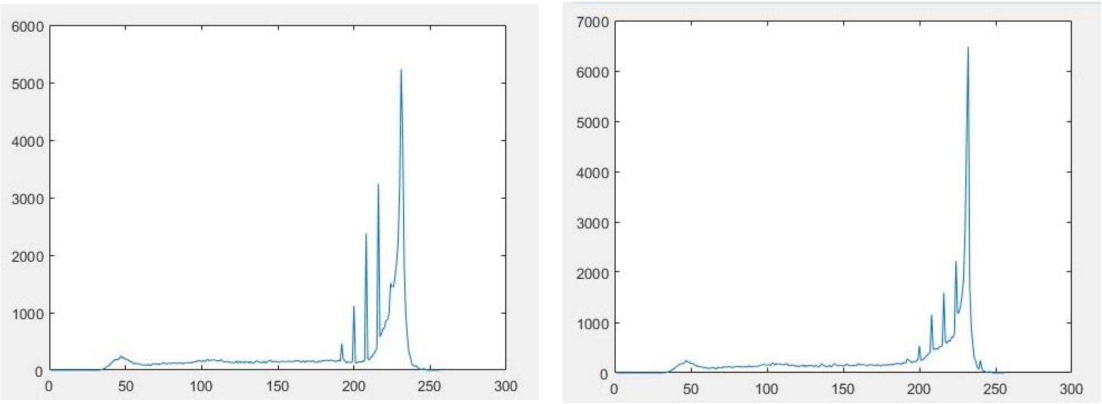


(g) (h)

**Figure 4.17:** (a) The secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a) (b)



(c) (d)

**Figure 4.18:** Histogram for (a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

Secret message size is 51x51x8 bits, which equal to 20808 bits. LSB (1 bpp) will insert 3 bits inside each pixel and by that; we need to have 6936 pixels. For LSB (3 bpp), Last pixel used is pixel number 24 in row 28. For LSBG (3 bpp) (1LSB to 3G), Last pixel used is pixel number 96 in row 109 (in each 256 size row 64 are used like 4, 8, 12, 16, ...,256.

### 4.12. Experiments 10

Cover image: watch image, grayscale image with size of 256x256.

Secret Message: 51x51 matrix with fixed pixel value of 255.

Method: a) LSBG (3 bpp) (1LSB to 7G).

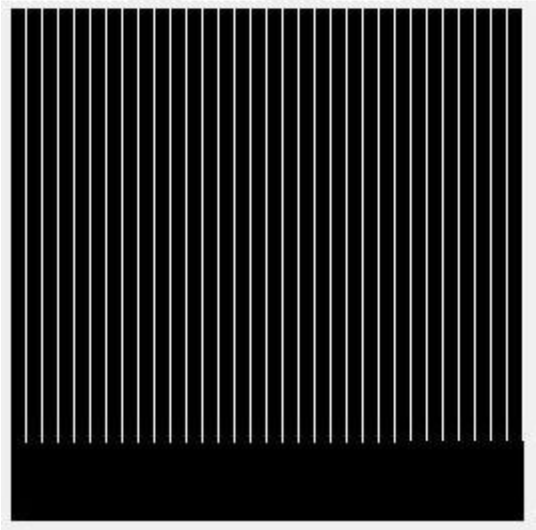
b) 2D LSBG (3 bpp) (1LSB to 2G).

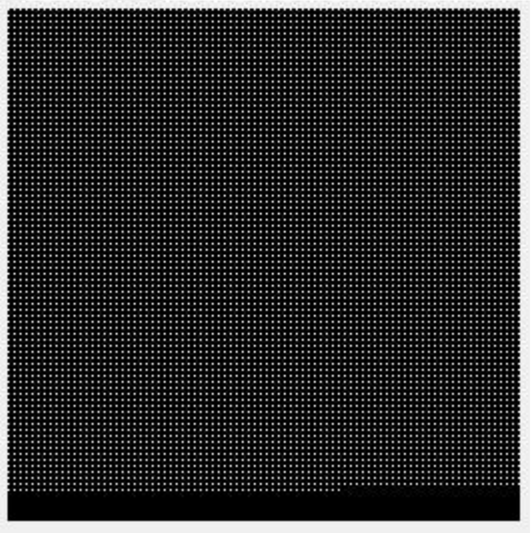
(a)

(b)

(c)

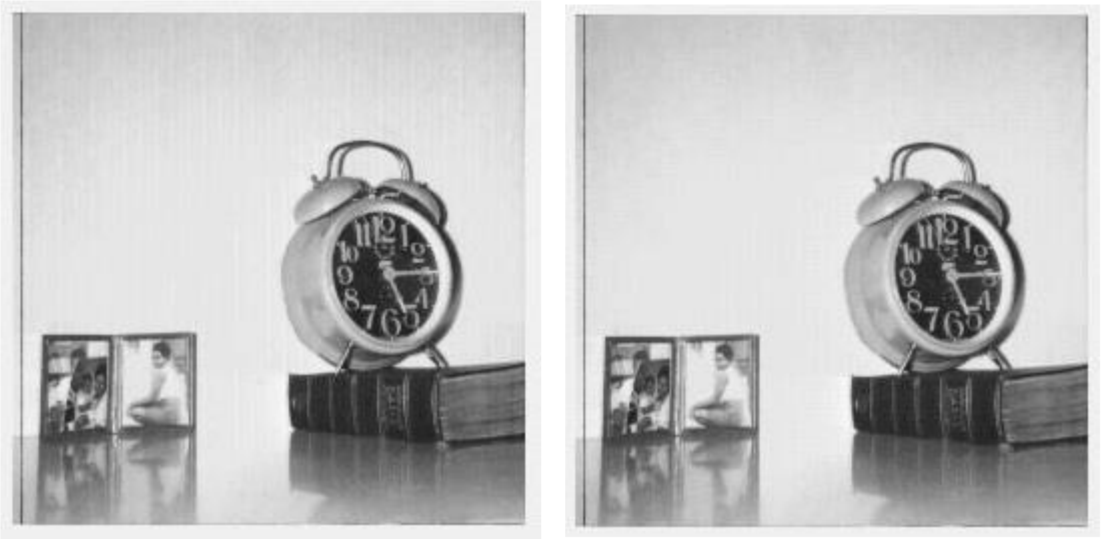
(d)





(e)

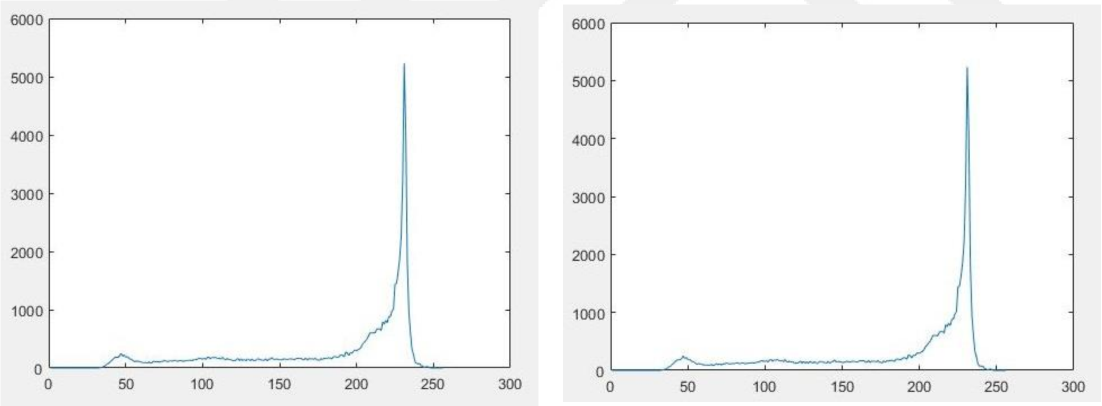
(f)



(g)

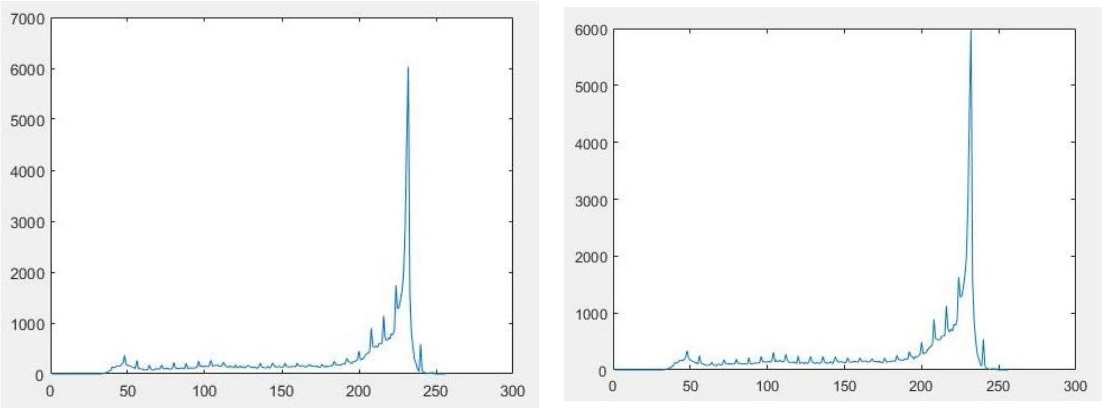
(h)

**Figure 4.19:** (a) the secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a)

(b)



(c) (d)

**Figure 4.20:** Histogram for (a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

Secret message size is (51x51x8) bits, which equal to 20808 bits. LSB (3 bpp) will insert 3 bits inside each pixel and by that; we need to have 6936 pixels. For LSBG (3 bpp) (1LSB to 7G), Last pixel used is pixel number 192 in row 217 (in each 256 size row 32 are used like 8,16,24,32 ...256). For 2D LSBG (3 bpp) (1LSB to 2G), Last pixel used is pixel number 166 in row 241 (in each 256 size row 86 are used like 1, 3, 6, 9, ... ,256.

### 4.13. Experiments 11

Cover image: Cameraman image, grayscale image with size of 512x512.

Secret Message: a grayscale pepper image with size of 51x51.

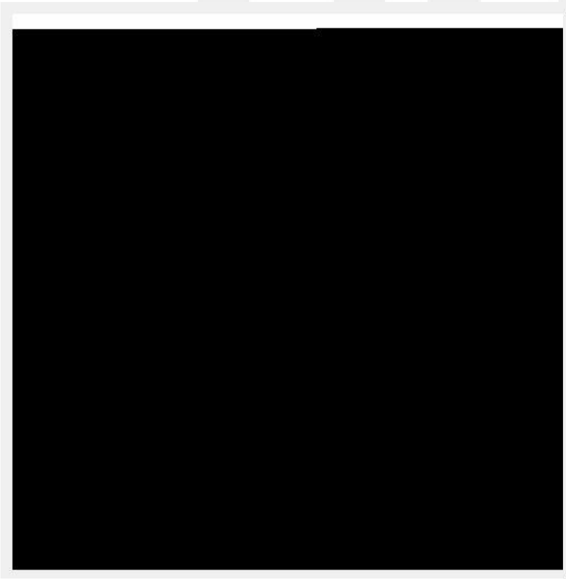
Method: a) LSB (3 bpp).

b) 2D LSBG (3 bpp) (1LSB to 3G) ratio.

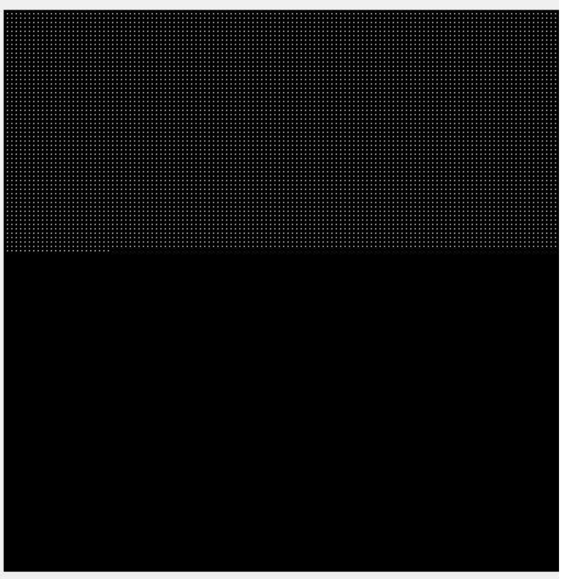
(a) (b)

(c)

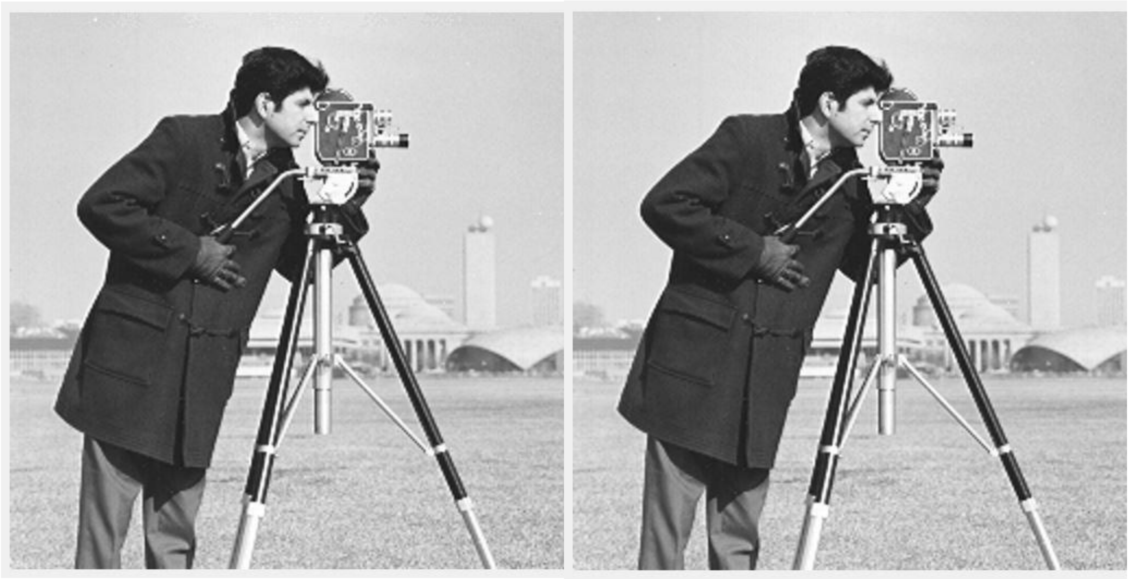
(d)



(e)

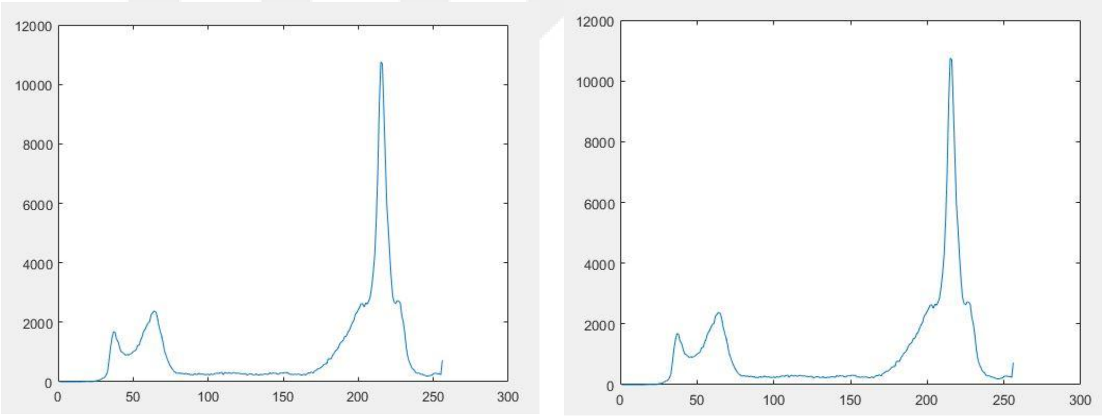


(f)

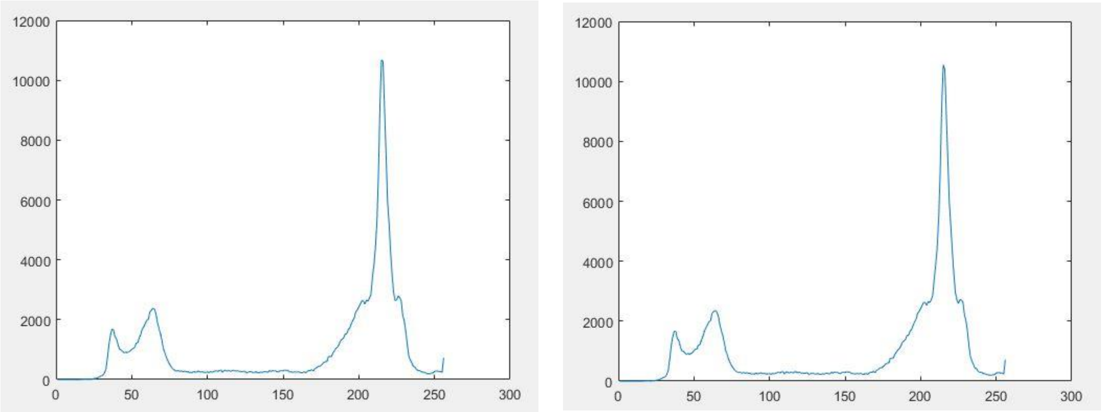


(g) (h)

**Figure 4.21:** (a) The secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a) (b)



(c) (d)

**Figure 4.22:** Histogram for (a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

Secret message size is 51x51x8 bits, which equal to 20808 bits. LSB (3 bpp) will insert 3 bits inside each pixel and by that; we need to have 6936 pixels. For LSB (3 bpp), Last pixel used is pixel number 280 in row 14. For 2D LSBG (3 bpp) (1LSB to 3G), Last pixel used is pixel number 96 in row 220. In each row or column 128 pixels can be used like 4, 8, 12, 15 …. 512.

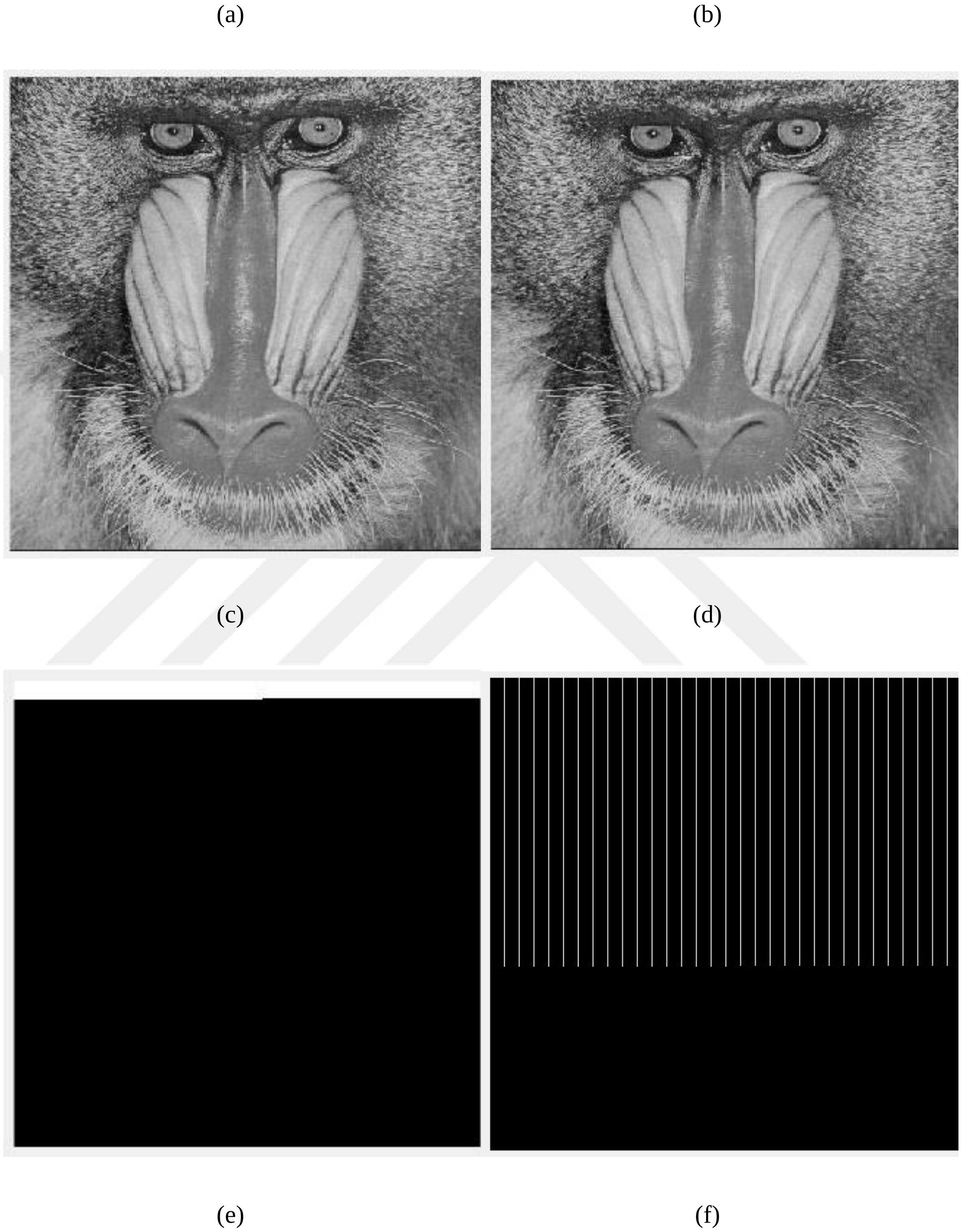
### 4.14. Experiments 12

Cover image: Baboon greyscale image, grayscale image with size of 512x512.

Secret Message: 50x50 matrix with fixed pixel value of 255.

Method: a) LSB (2 bits).

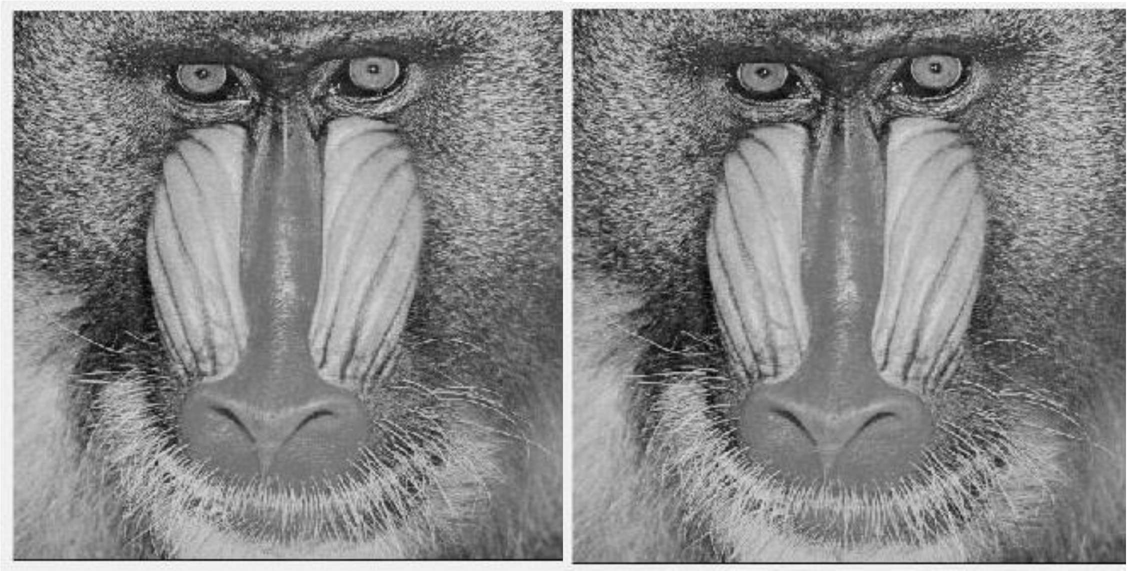
b) LSBG (2 bits) (1LSB to 15G).



(a) (b)

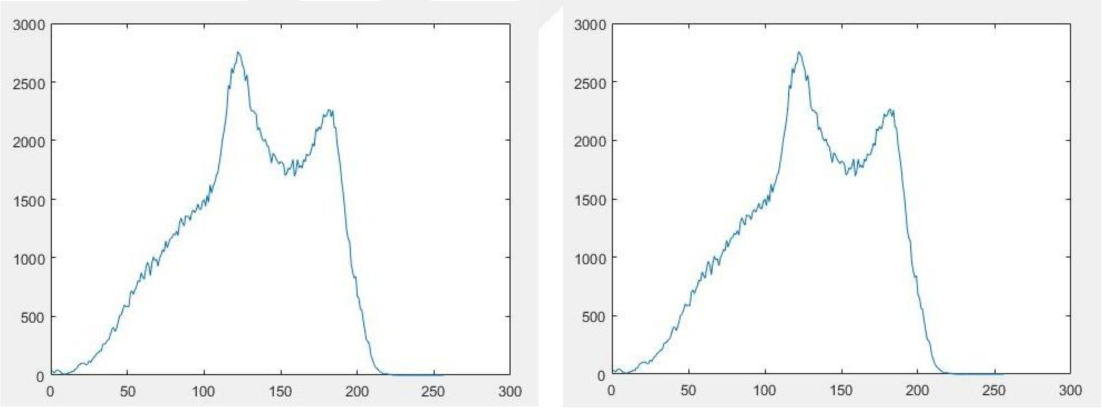
(c) (d)

(e) (f)

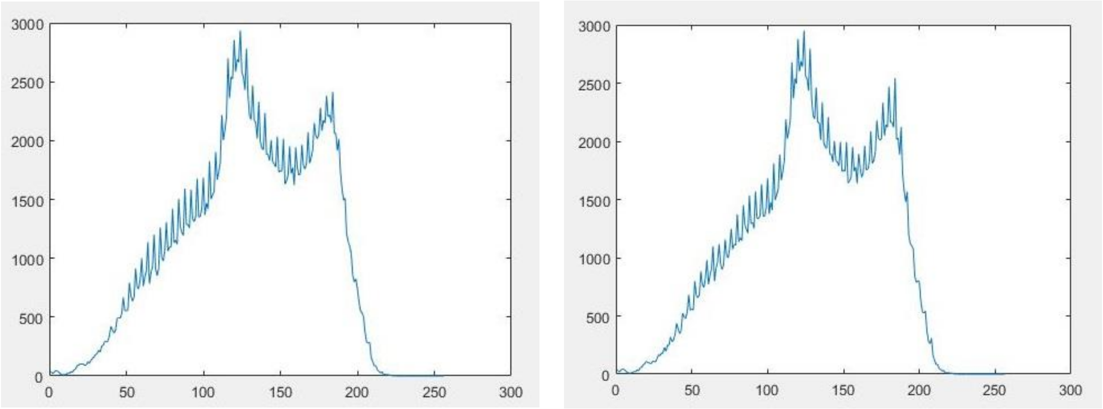


(g) (h)

**Figure 4.23:** (a) The secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a) (b)



(c) (d)

**Figure 4.24:** Histogram for (a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

Secret message size is 50x50x8 bits, which equal to 20000 bits. LSB (2 bits) will insert 2 bits inside each pixel and by that; we need to have 10000 pixels. For LSB (2 bits), Last pixel used is pixel number 272 in row 20. For LSBG (2 bits) (1LSB to 15G), Last pixel used is pixel number 256 in row 313 (in each 512 size row 32 are used like 16,32,48,64 ...512).

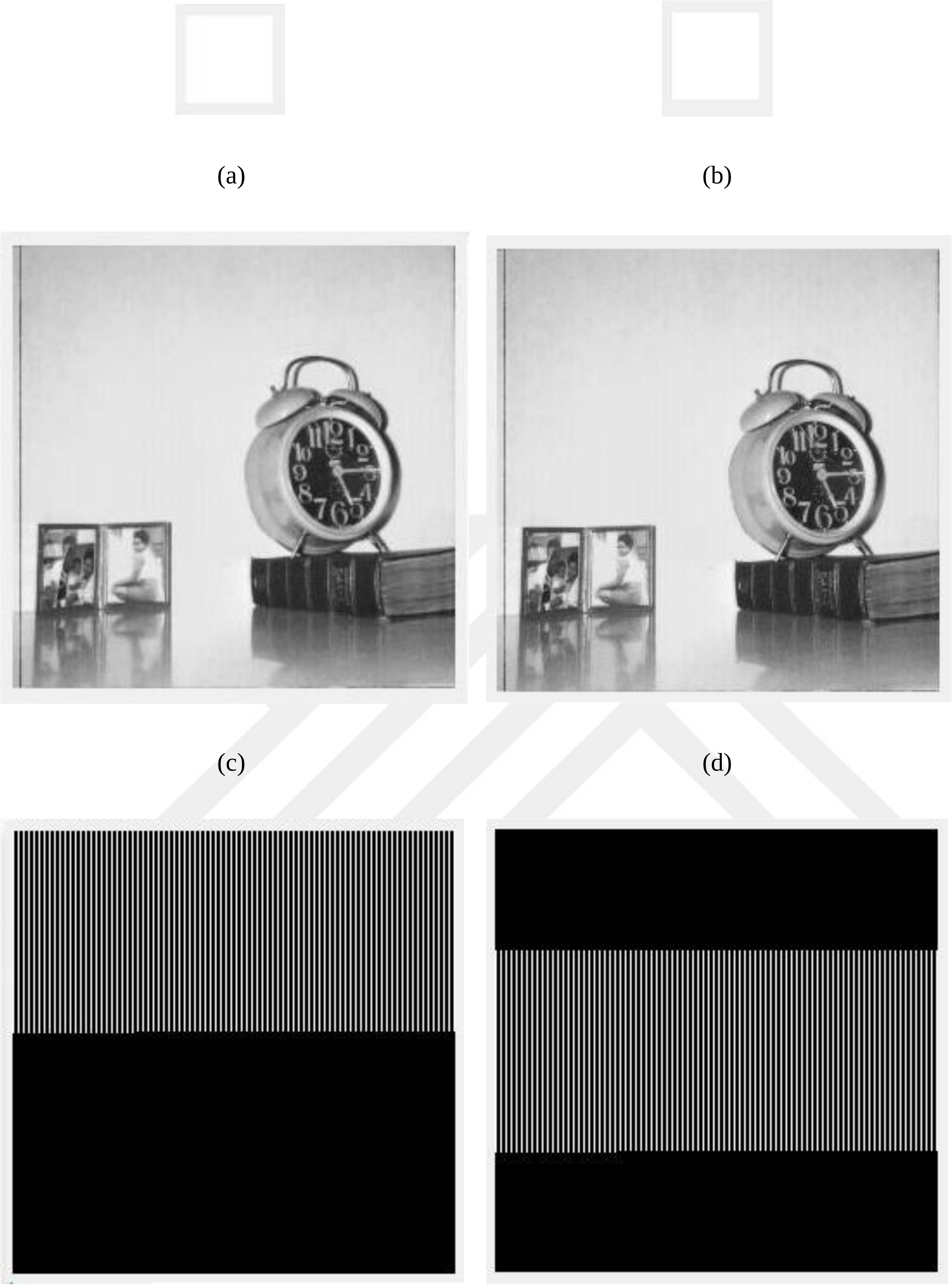
### 4.15. Experiments 13

Cover image: watch image, grayscale image with size of 256x256.

Secret Message: 50x50 matrix with fixed pixel value of 255.

Method: a) LSBG (2 bits) (1LSB to 2G).

b) Shifted LSBG (2 bits) (1LSB to 2G).



(a)

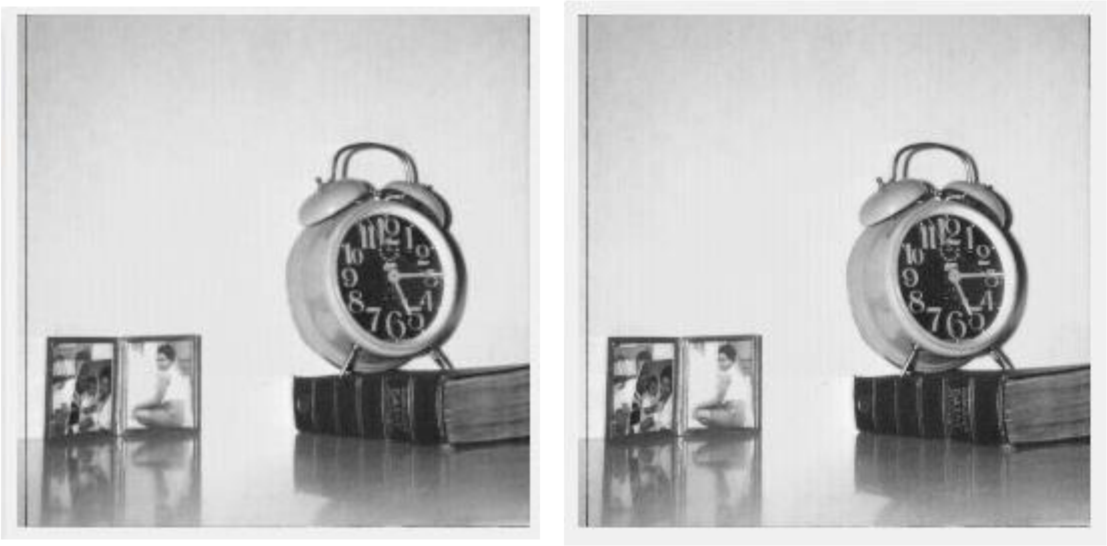
(b)

(c)

(d)

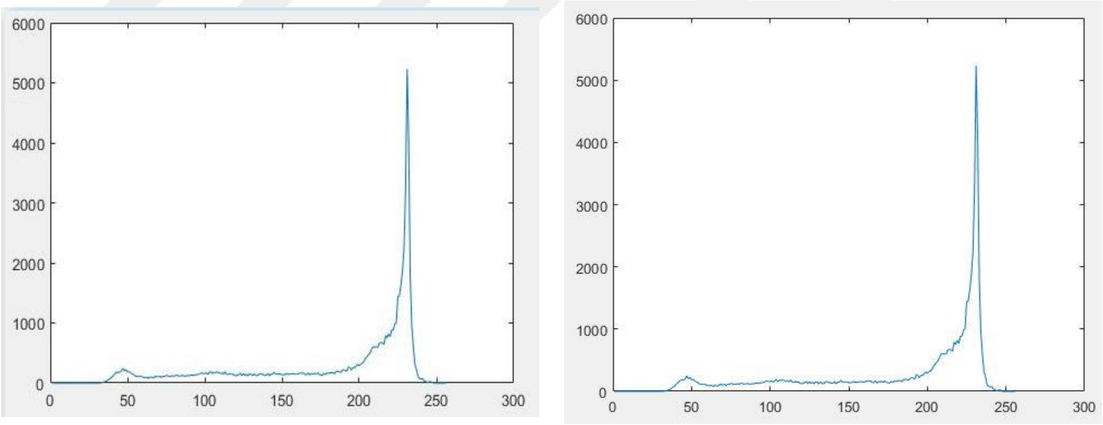
(e)

(f)

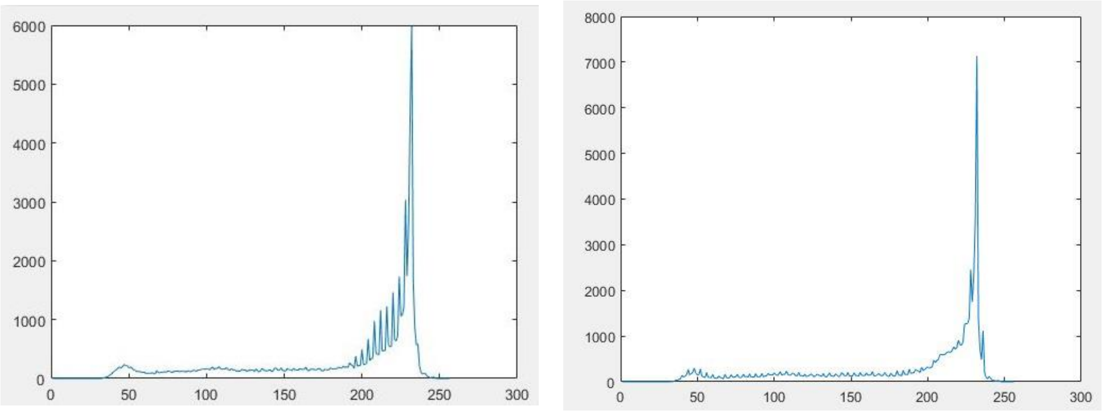


(g) (h)

**Figure 4.25:** (a) The secret image 1; (b) secret image 2; (c) and (d) cover image before embedding; (e) and (f) the demonstration of the selected pixels for embedding in white and the unchanged pixels in black; (g) the stego image of secret image 1 (h) the stego image of secret image 2.



(a) (b)



(c) (d)

**Figure 4.26:** Histogram for (a) the cover image before LSB embedding of secret message 1; (b) the cover image before LSB embedding of secret message 2; (c) the stego image after LSB embedding of secret message 1; (d) the stego image after LSB embedding of secret message 2.

Secret message size is 50x50x8 bits, which equal to 20000 bits. LSB (2 bits) will insert 2 bits inside each pixel and by that; we need to have 10000 pixels. For LSBG (2 bits) (1LSB to 2G), Last pixel used is pixel number 70 in row 117 (in each 256 size row 86 are used like 1, 4, 7, 10, …. ,256 ). For Shifted LSBG (2 bits) (1LSB to 2G), start from row 71. Last pixel used is pixel number 70 in row 187 (in each 256 size row 86 are used like 1, 4, 7, 10, …. ,256 ).

## 4.16 Quantative Analysis

**Table 4.1:** MSE & PSNR results for the experiments.

| Exp # | Secret | Cover image | Method | MSE | PSNR |
|---|---|---|---|---|---|
| 1a | Pepper image | Watch image | LSB (2 bpp) | 0.3586 | 52.5845 |
| 1b | White image | Watch image | LSB (2 bpp) | 0.5467 | 50.7528 |
| 2a | Pepper image | Cameraman image | LSB (2 bpp) | 0.0878 | 58.6926 |
| 2b | White image | Cameraman image | LSB (2 bpp) | 0.1340 | 56.8583 |
| 3a | Pepper image | Watch image | LSBG (2 bpp) (1LSB to 3G) | 0.3486 | 52.7069 |

| | | | | | |
|---|---|---|---|---|---|
| 3b | White image | Watch image | LSBG (2 bpp) (1LSB to 3G) | 0.4998 | 51.1428 |
| 4a | Pepper image | Cameraman image | LSBG (2 bpp) (1LSB to 7G) | 0.0896 | 58.6047 |
| 4b | White image | Cameraman image | LSBG (2 bpp) (1LSB to 7G) | 0.1337 | 56.8663 |
| 5a | Pepper image | Watch image | 2D LSBG (2 bpp) (1LSB to 1G) | 0.3472 | 52.7246 |
| 5b | White image | Watch image | 2D LSBG (2 bpp) (1LSB to 1G) | 0.5011 | 51.1307 |
| 6a | Pepper image | Cameraman image | 2D LSBG (2 bpp) (1LSB to 3G) | 0.0894 | 58.6151 |
| 6b | White image | Cameraman image | 2D LSBG (2 bpp) (1LSB to 3G) | 0.1333 | 56.8792 |
| 7a | White image | Watch image | LSB (1 bpp) | 0.1554 | 56.2142 |
| 7b | White image | Watch image | LSBG (1 bpp) (1LSB to 1G) | 0.1561 | 56.1951 |
| 8a | Pepper image | Cameraman image | LSB (1 bpp) | 0.0382 | 62.2997 |
| 8b | Pepper image | Cameraman image | 2D LSBG (1 bpp) (1LSB to 2G) | 0.0385 | 62.2742 |
| 9a | White image | Watch image | LSB (3 bpp) | 1.8798 | 45.3894 |
| 9b | White image | Watch image | LSBG (3 bpp) (1LSB to 3G) | 1.7155 | 45.7867 |
| 10a | White image | Watch image | LSBG (3 bpp) (1LSB to 7G) | 1.6939 | 45.8418 |
| 10b | White image | Watch image | 2D LSBG (3 bpp) (1LSB to 2G) | 1.7011 | 45.8232 |
| 11a | Pepper image | Cameraman image | LSB (3 bpp) | 0.2424 | 54.2850 |
| 11b | Pepper image | Cameraman image | 2D LSBG (3 bpp) (1LSB to 3G) | 0.2668 | 53.8677 |
| 12a | White image | Baboon image | LSB (2 bpp) | 0.1319 | 56.9265 |
| 12b | White image | Baboon image | LSBG (2 bpp) (1LSB to 15G) | 0.1333 | 56.8822 |
| 13a | White image | Watch image | LSBG (2 bpp) (1LSB to 2G) | 0.5203 | 50.9681 |
| 13b | White image | Watch image | Shifted LSBG (2 bpp) (1LSB to 2G) | 0.4687 | 51.4209 |

# 5 CONCLUSION AND FUTURE WORK

## 5.1 Discussion and Conclusion

From generally analyzing the experiments results with the different cases applied, we find out that when simple LSB method is applied, once the histogram is analyzed we can define that the deviation is in a specific range of the histogram and the deviation is high and can be simply noticed.

While in applying LSBG with different rates, when we analyze the histogram of the stego image we find out that the deviation is applied on a higher range and the range cannot be accurately defined. The other interesting improvement that the deviation effect is getting reduced singe the deviation itself has been stretched and spread in the histogram range. The more we increase the LSBG rate, the more the deviation in the histogram of the stego image is reduced and distributed in a higher range of the histogram.

Three different cover images have been selected for the experiments. Initially with the watch image we applied embedding for the secret messages, LSB and LSBG methods with three embedding rates of 1 bpp, 2bpp, 3bppand different LSBG rates for the gapping. It is clear that from analyzing the histograms of the different cases especially after applying LSBG method, it can be noticed the improvement of reducing the deviation value. The second cover image is the cameraman image and it was selected with a higher image size to give the capability of applying a higher LSBG rates.

When we analyze a specific case of the baboon image as cover image, if we study the original histogram of the cover image we can see that the image has a wide dynamic range. Once the properties of the image are analyzed, we can notice that the majority of the distributed pixels in the cover image have the same intensity values. That is why when we applied LSB and LSBG, we can see that there is no great difference in the deviation of the stego histograms of both. But quite the opposite and that is when we applied the basic LSB it results the embedding in the specific upper part of the cover image. In addition, in LSBG it did distribute the embedding in a greater part of the image but still the majority of the selected pixels intensities of LSBG are equal to selected pixels intensities the LSB. Moreover, by that it is logical to have the same changes and to have approximately the same stego histograms reactions. This case shows the great importance of selecting a proper cover image for LSBG method and in general for all other methods.

From analyzing the MSE and PSNR parameters for the watch image we can conclude that for both cases of the secret messages and with the different embedding rates that the LSBG method improved the process by reducing the MSE value and logically increasing the PSNR value for the different cases. For the Cameraman image cases we can notice the same improvement in the MSE and PSNR values with the secret white secret image cases. However, there was no improvement in applying the pepper secret image for the Cameraman image but even the MSE was increased and PSNR was reduced when LSBG method. Although it can be still considered acceptable since the value of diffidence can be considered as small and the MSE, PSNR of both LSB, LSBG methods can be approximately considered the same. For the embedding rates stating from 1bpp it has the best MSE values as the smallest and the best PSNR values as the greatest although the amount of pixels to be edited is the greatest since only 1 bit can be used in each pixel. In 2bpp embedding rate we are having a higher MSE values and lower PSNR values but still it can be considered as reasonable results. The advantage of having 2bpp over 1bpp is the capability of applying a higher LSBG rates. For 3 bpp embedding rate it has the least amount of pixels to be edited but the amount of possible change in each pixel is great and that can be noticed clearly as we have the highest values of MSEs and the lowest values of PSNRs in the applied 3bpp embedding rate cases.

## 5.2 Recommendations

From our study, we recommend to select a cover image with a narrow dynamic range histogram to ensure the improvement of the performance in applying LSBG method. We suggest using multiple options for cover images and applying multiple LSBG rates to them and from analyzing the results, the case with best result will be chosen. It is recommended to apply the highest LSBG rate possible to ensure a greater distribution of LSB embedded pixels in the selected cover image. Another recommendation is to apply the shifting property of LSBG to enhance the key complexity of the steganography method used. It is also recommended to apply encryption as pre-stage for LSBG to add another level of security to the system. We suggest applying a lossless compression as pre-stage to LSBG steganography which can solve the secret capacity problem by reducing it and give the capability to use a higher LSBG rates for embedding in the cover medium.

## 5.3 Future Works

The proposed possible future work is listed as follows:

- One of the possible fields that can be studied is to apply the LSBG method in other transform domains instead of the direct embedding the spatial domain. For example to study the performance of LSBG in the frequency domain after applying Fourier transform.
- Another possible field to be studied is to apply the LSBG in the multiband colored images. Applying LSBG can be done on specific selected band. And there is also the capability to apply LSBG in multiple bands of the total available bands of the cover image.
- In our work, image steganography was selected for the study. There is the possibility to apply the LSBG method in audio steganography and other possible steganography domains.
- It is possible to study the process of secret messages multiplexing while using the LSBG method.
- Another important field is to produce an adaptive LSBG method that can study its inputs and automatically chose the best LSBG rate to be used for that specific case.
- The advantage of the proposed method that it does not change the embedding technique but instead it changes the selection of the pixels to be used of embedding. For that, the gapping method can be even applied to other steganography methods that use different embedding techniques than the basic LSB method.

# 6 REFERENCES

[1] R. J. Anderson and F. A. P. Petitcolas, "On the Limits of Steganography," *IEEE Journal on Selected Areas in Communications,* vol. 16, no. 4, pp. 474 - 481, 1998.

[2] Petitcolas, F. A. P. R. J. Anderson and M. G. Kuhn, "Information Hiding - A Survey," *Proceedings of the IEEE,* vol. 87, no. 7, pp. 1062 - 1078, 1999.

[3] L. M. Marvel, C. G. Boncelet and C. T. Retter, "Spread Spectrum Image Steganography," *IEEE Transactions on Image Processing,* vol. 8, no. 8, pp. 1075 - 1083, 1999.

[4] N. Nikolaidis and I. Pitas, "Digital Image Watermarking: an Overview," in *Proceedings IEEE International Conference on Multimedia Computing and Systems*, 1999.

[5] N. F. Johnson and S. Jajodia, "Exploring Steganography: Seeing the Unseen," *IEEE, Computer ,* vol. 31, no. 2, pp. 313 - 336, 1998.

[6] W. Bender, D. Gruhl, N. Morimoto and A. Lu, "Techniques for data hiding," *IBM Systems Journal,* vol. 35, no. 3.4, pp. 313 - 336, 1996.

[7] G. L. Smitha and E. Baburaj, "A Survey on Image Steganography Based on Least Significant Bit Matched Revisited (LSBMR) Algorithm," in *International Conference on Emerging Technological Trends (ICETT) ,* 2016.

[8] T. Shelare and V. Powar, "A Secure Data Transmission Approach Using B+trees In Steganography," in *International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT),* 2016.

[9] J. Kumar, "A Novel Approach to Image Steganography using Quadtree Partition," in *2nd International Conference on Next Generation Computing Technologies (NGCT),* 2016.

[10] A. Abuadbba and I. Khalil, "Walsh-Hadamard Based 3D Steganography for Protecting Sensitive Information in Point-of-Care," *IEEE Transactions on Biomedical Engineering,* vol. 64, no. 9, pp. 2186 - 2195, 2017.

[11] V. Sharon, B. Karthikeyan, S. Chakravarthy and V. Vaithiyanathan, "Stego Pi : An Automated Security Module for Text and Image Steganography using

Raspberry Pi," in *International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 2016.

[12] A. A. J. Altaay, S. b. Sahib and M. Zamani, "An Introduction to Image Steganography Techniques," in *International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, 2012.

[13] M. Kude and M. Borse, "Skintone Detection Based Steganography Using Wavelet Transform," in *International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, 2016.

[14] J. Fridrich, M. Goljan and R. Du, "Detecting LSB Steganography in Color and Gray-Scale Images," *IEEE MultiMedia*, vol. 8, no. 4, pp. 22 - 28, 2001.

## APPENDICES

**APPENDIX A:** MATLAB code for LSBG Encoding

**APPENDIX B:** MATLAB code for LSBG Decoding

## APPENDIX A

MATLAB code for LSBG Encoding

```
clear;
clc;

% LSB_2bit encode
% the code is edited in 26-12-2017
% The secret message is (50x50) matrix  with fixed pixel value of
255
% grayscale image with size(256x256) of  is chosen as a cover image
% 10000 pixels are needed for (2bit)LSB in the cover image since the
secret
% message capacity is (50x50x8bit)=20000 and 2bit is inserted in
each pixel
% so 10000 are needed
% last pixel is in row 40 column 16

secret= imread('spepper.bmp');
% produce (50x50)matrix (image)

secret1= dec2bin(secret);
% initially the secret message is a(50x50)matrix with pixel value of
255 converted to binary

secret2=transpose(secret1);
secretbin= secret2(:);
% the binary matrix is reformed to a vector

N= size(secretbin);
% find the total number of binary bits to be inserted

n=N(1);

cover= imread('watch.tiff');
% read the carrier (cover) image

[R,C]=size(cover);
% find number of rows and columns of the cover image

stego = zeros(R,C);
% initially the stego image is equal to cover image
%stego = cover ;

w=1;
% used for counting for secret bits

for x=2:2:C                                      % for columns
    for y=2:2:R                                           % for rows
```

```
            a= cover(x,y) ;
% selected pixel decimal value

% b= bin2dec(secretbin(w)) ; % for 1bit LSB insertion

b=bitshift(bin2dec(secretbin(w)),1)+bin2dec(secretbin(w+1)) ;
% secret value to be inserted (2 bits)(1st bit is shifted and second
is not)
% b=
bitshift(bin2dec(secretbin(w)),2)+bitshift(bin2dec(secretbin(w+1)),1
)+bin2dec(secretbin(w+2)); % for 3bit LSB insertion

            w=w+2;
            a1=bitshift(a,-2);
% delete 2 LSB of the cover pixel value
            a2=bitshift(a1,2);
% have the 2 LSB values as (00)
            stego(x,y)=a2+b;
% insert secret 2bits in the selected and edited cover pixel by
simply adding the decimal value

          if w == n+1
% if the counter reached to the last bit in the secret stream (last
w= n-1 and w+1= n w+2= n+1)
             break
% end the y for loop
          end

    end
    if w == n+1
% if the counter reached to the last bit in the secret stream (last
w= n-1 and w+1= n w+2= n+1)
       break
% end the x for loop
    end
end


figure
imshow(secret);

figure
imshow(cover);

figure
imshow(stego);
```

## APPENDIX B

MATLAB code for LSBG Decoding

```
clear;
clc;

% the code is edited in 4-12-2017
% The secret message is (2x2) matrix  [255 255 ; 255 255]
% grayscale image with size(256x256) of  is chosen as a coverimage


stego= imread('stego-4-12-2017.tiff');
% read the stego image
[R,C]=size(stego);
% find number of rows and columns



empty= zeros(8,4);                         % prepare a empty matrix
empty1=empty(:);                           % convert to vector
extractedbin= char(empty1);
% create an empty bit stream to be filled after decoding

n=32  ;
% total number of secret message binary bits
w=1  ;
% counter for 2LSB to be extracted


for x=1:R                                         % for columns
    for y=1:C                                     % for rows

            a=dec2bin(stego(x,y));
% a is binary value of selected stego pixel
            S=size(a);
% count how many bits in converted binary value
            s=S(2);
% the second number is the number of bits
            b=a(s-1:s);
% b is the 2LSB binary value (last 2 bits)
           extractedbin(w:w+1)= b ;
% 2lsb bits are inserted in sequence in extractedbin vector
           w=w+2;
% LSB bit counter is incremented by 2
            if w == n+1
% if the counter reached to the total number of secret binary bits
                break
% end the loops as all data stream has been extracted
            end
    end
    if w == n+1
% if the counter reached to the total number of secret binary bits
```

```
         break
% end the loops as all data stream has been extracted
    end
end


extracted= bin2dec(extractedbin);
% convert the extracted binary bitstream into pixel matrix
```

## RESUME

**Name Surname:** Waleed Tuza

**Date of birth:** 9-2-1992

**Education:**

- **Bachelor:** 2013, APPLIED SCIENCE PRIVATE UNIVERSITY in Amman – Jordan. Communication and Electronics Engineering
- **Masters:** ISTANBUL AYDIN UNIVERSITY in Istanbul – Turkey. Electronics Engineering (2nd year).

**Certificates:**

- SIAE Microelettronica microwave systems training certificate.
- TOEFL IBT English language certificate.
- TÖMER (B1) Turkish language certificate.

**Contact:**

waleed.tuza@gmail.com