Multimedia Data Hiding: Three-in-One

اخفاء بيانات متعددة الوسائط: ثلاثة–في–واحد

Dr. Hameed Abdul-Kareem Younis
Dept. of Computer Science,
College of Science,
University of Basrah
E-mail: hameedalkinani2004@yahoo.com

Dr. Issa Ahmed Abed
Dept. of Electrical Power Technologies Engineering,
Engineering Technical College Basrah
Southern Technical University
E-mail: issaahmedabd80@yahoo.com

Hussain A. Younis
Dept. of History,
College of Education for Woman, University of Basrah
E-mail: hussain394@yahoo.com

Abstract

Data hiding considers a class of process utilized to embed data into various forms of media, such as image, audio, or text. The suggested data (text, audio, binary image) in image steganography system is a new technique used to embed data into color image. Instead of using the least-significant-bit-plane (LSB 1) of the cover for embedding the data, third-least-significant-bit-plane (LSB 3) has been utilized in order to increase the robustness. First and second-least-significant-bit-plane (LSB 1 and LSB 2) can be changed according to bits of the data, to reduce the difference between the cover and the stego-cover. Then, for more protection to the data characters, a stego-key has been constructed to permute the data characters before embedding it. Experimental results of the modified technique proves that Peak-Signal-to-Noise-Ratio (PSNR) is greater than the traditional approach of LSBs substitution. Thus, the proposed system introduces good results and it is suitable for several cases of life.

Keywords: Steganography, cover-image, LSBs, PSNR, multimedia, data hiding, stego-image.

المستخلص

إخفاء البيانات يعتبر نوع من المعالجة يستفاد منه لإخفاء بيانات في صيغ متعددة من الأوساط مثل الصورة، الصوت، والنص. نظام إخفاء بيانات (نص وصوت وصوة ثنائية) صورة المقترح هو طريقة جديدة تستخدم لإخفاء بيانات داخل صورة ملونة. الطريقة المقترحة تعتمد على تخزين ثنائيات البيانات في الطبقة الثالثة LSB 3 من ثنائيات الصورة بدلا من الطبقة الأولى LSB 1، وذلك لزيادة متانة البيانات داخل الغطاء. الهدف الأساسي لهذا البحث هو تقليل الفرق بين قيم الغطاء قبل وبعد إخفاء البيانات فيه، عن طريق تغيير ثنائيات الطبقة الأولى والثانية LSB 1 و LSB 2 من الغطاء، وذلك يؤدي إلى زيادة سرية الاتصال بين المرسل والمستلم وصعوبة ملاحظة المهاجم للتشوه في بيانات الغطاء الناتجة عن عملية الإخفاء. بعد ذلك، لمزيد من الحماية للبيانات قمنا ببعثرتها باستخدام مفتاح- إخفاء قبل أن يتم إخفائها في الغطاء. النتائج أثبتت أن الطريقة المستخدمة أعطت قيمة تشابه اكبر PSNR بين الغطاء قبل وبعد الإخفاء أكثر من التشابه في الطريقة التقليدية للإخفاء. وهكذا، النظام المقترح قدم نتائج جيدة ويمكن تطبيقه في عدة حالات في الحياة.

الكلمات المفتاحية: كتابة مخفية، صورة-غطاء، البت الأقل أهمية، نسبة قمة الإشارة-إلى-الضوضاء، الوسائط المتعددة، اخفاء البيانات، صورة اخفاء.

1. Introduction

1.1 Overview

In classical cryptography, even if the information contents are protected by encryption, the existence of encrypted communications is known. In view of this, digital steganography introduces a different method in which it conceals even the evidence of encrypted messaging. In general, steganography is defined as the art and science of communicating in a covert fashion [1]. It uses the typical digital media like text, image, audio, video, and multimedia as a carrier (called *a host signal*) for hiding private information in such a way that the third parties (unauthorized person) cannot detect or even notice the presence of the communication. In that case, steganography allows for authentication, copyright protection, and embedding of messages in the image or in transmission of the image [1, 2].

A typical digital steganographic encoder is illustrated in Figure (1). The message is the data that the sender wishes to remain confidential and can be text, images, audio, video, or any other data that can be represented by a stream of bits. The cover or host is the medium in which the message is embedded and serves to hide the presence of the message. This is also referred to as the message wrapper. The message embedding method is highly dependent on the structure of the cover, and in current work, covers are restricted to being digital images. It is not required that the cover and the message have homogeneous structure. The image with the secretly embedded message produced by the encoder is the stego-image. The stego image should resemble the cover image under casual inspection and analysis. As well as, the encoder usually employs a stego-key which ensures that only recipients who know the corresponding decoding key will be able to extract the message from a stego-image.

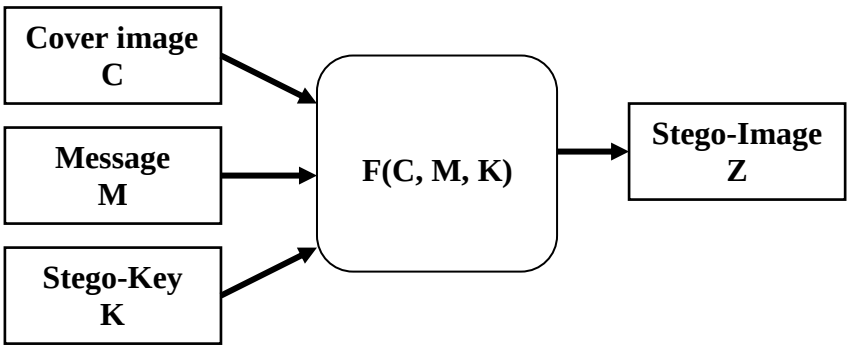


Figure (1): Steganographic Encoding.

Recovering the message from a stego-image requires the stego-image itself and a corresponding decoding key if a stego-key was used in the encoding process. The original cover image may or may not be required; in many applications it is desirable that the cover image not be needed to extract the message. It requires the cryptographic decoding key to decipher the encrypted message.

1.2 Applications

There are various applications for digital steganography of images, including copyright protection, feature tagging, and secret communications [3, 4]:

Copyright Protection: A secret copyright notice or watermark can be embedded inside an image to identify it as intellectual property. This is the watermarking scenario where the message is the watermark [5, 6]. The "watermark" can be a relatively complicated structure. In addition, when an image is sold or distributed an identification of the recipient and time stamp can be embedded to identify potential pirates. A watermark can also serve to detect whether the image has been subsequently modified [7]. Detection of an embedded watermark is performed by a statistical, correlation, or similarity test, or by measuring other quantity characteristic to the watermark in a stego-image. The insertion and analysis of watermarks to protect copyrighted material is responsible for the recent surge of interest in digital steganography and data embedding.

Feature Tagging: Captions, annotations, time stamps, and other descriptive elements can be embedded inside an image, such as the names of individuals in a photo or locations in a map. Copying the stego-image also copies all of the embedded features and only parties who possess the decoding stego-key will be capable to extract and view the features. In an image database, keywords can be embedded to facilitate search engines. If the image is a frame of a video sequence, timing markers can be embedded in the image for synchronization with audio. The number of times an image has been viewed can be embedded for "pay-perview" applications.

Secret Communications: In different cases, transmitting a cryptographic message draws unwanted attention. The use of cryptographic technology may be restricted or forbidden by law. However, the use steganography does not advertise covert communication and therefore avoids scrutiny of the sender, message, and recipient. A trade secret, blueprint, or other sensitive information can be transmitted without alerting potential attackers or eavesdroppers. It is important that the embedding occur without significant degradation or loss of perceptual quality of the cover. In a secret communications application, if an attacker notices some distortion that arouses suspicion of the presence of hidden data in a stego-image, the steganographic encoding has failed even if the attacker is unable to extract the message. Preserving perceptual transparency in an embedded watermark for copyright protection is also of paramount importance because the integrity of the original work must be maintained [6]. For applications where the perceptual transparency of embedded data is not critical, allowing more distortion in the stego-image can increase hiding capacity, robustness, or both.

Robustness: Robustness refers to the ability of embedded data to maintain intact if the stego-image undergoes transformations, for example, linear and non-linear filtering, addition of random noise, sharpening or blurring, scaling and rotations, cropping or decimation, lossy compression, and conversion from digital to analog form and then reconversion back to digital form (such as in the case when a hard copy of a stego-image is printed and then a digital image is formed by subsequently

scanning the hardcopy). Robustness is critical in copyright protection watermarks because pirates will attempt to filter and destroy any watermarks embedded in images [5, 6]. Anti-watermarking software is already available on the Internet and have been shown effective in removing some watermarks [8, 9]. These methodes can also be used to destroy the message in a stego-image.

Tamper Resistance: Beyond robustness to destruction, tamper resistance refers to the difficulty for an attacker to alter or forge a message once it has been embedded in a stego-image, such as a pirate replacing a copyright mark with one claiming legal ownership. Applications that demand high robustness usually also demand a strong degree of tamper resistance. In a copyright protection application, achieving good tamper resistance can be difficult because a copyright is effective for many years and a watermark must remain resistant to tampering even when a pirate attempts to modify it using computing technology decades in the future.

1.3 Least Significant Bit (LSB 1) Substitution

This is the simplest of the steganography methods based in the use of LSB, and therefore the most vulnerable. The embedding process consists of the sequential substitution of each Least Significant Bit (LSB 1) of the image pixel for the bit message. For its simplicity, this method can camouflage a great volume of information [10, 11, 12, 13]. This technique is quite simpleton and it presents a safety fault. It is necessary only a sequential LSB reading, starting from the first image pixel, to extract the secret message. This methods also generate a unbalanced distribution of the changed pixels, because the message is embedded at the top of the image. In the next section, an adaptive method will be proposed.

2. The Proposed Method (Adaptive LSB Substitution)

In this paper, a 256×256 color image has been proposed as a cover. Therefore, a message (data) up to 65536 bits (8192 bytes) can be hidden. The message is embedded in the LSB 3 of the cover to increase the robustness of the system and protect the message against the external influences such as noise, filter, and compression, … etc.

Let's have the message bits set $M=\{m_0, m_1, m_2, \ldots, m_{L-1}\}$, where $1 <= L < = 65536$, L is the length of the message that is embedded, and $m_i = \{0, 1\}$, for $i = 0, \ldots, L-1$. Let us have the cover image = $\{pixel_0, pixel_l, \ldots, pixel_{65535}\}$. Suppose that LSB 3 of the cover image is LSB 3 = $\{c_0, c_1, c_2, \ldots, c_{65535}\}$, where $c_j = \{0, 1\}$ for each $j = 0, \ldots, 65535$. In order to protect the message, a stego-Key is used, which is employed as a seed for pseudo-random number generator (PRNG). This creates a sequence of indexes used to permute the message bits. Figure (2) presents the block diagram of message characters permutation.

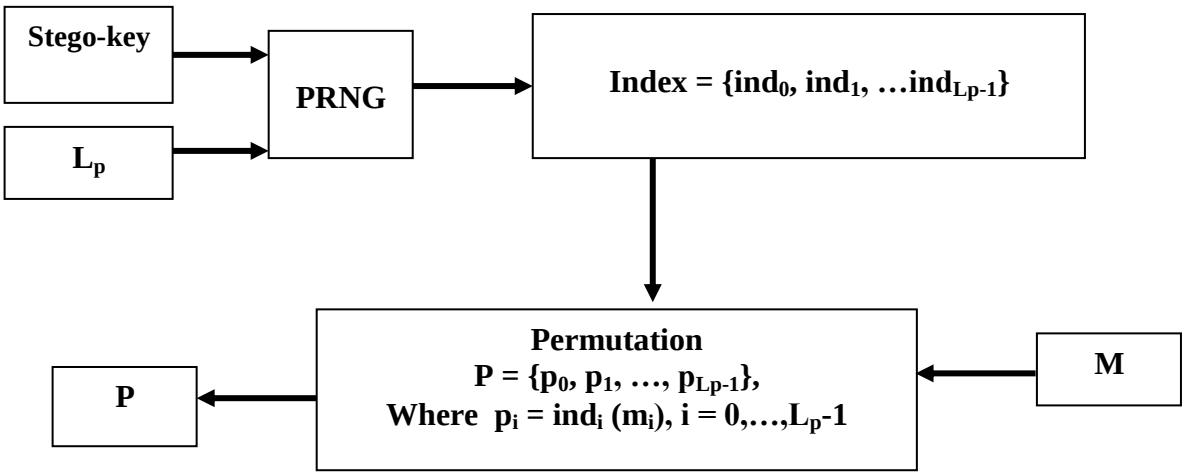


Figure (2): Message Characters Permutation.

The embedding process is very easy, which only replaces the permutated characters of the message (P) by the LSB 3 set of the cover to obtain the new stego-image Z = {$newpixel_0$, $newpixel_1$, …, $newpixel_{65535}$}.

To minimize the difference between the old value (pixel) in the cover and the new value (newpixel) in the stego-image, we suggest the following embedding algorithm:

2.1 Embedding Algorithm

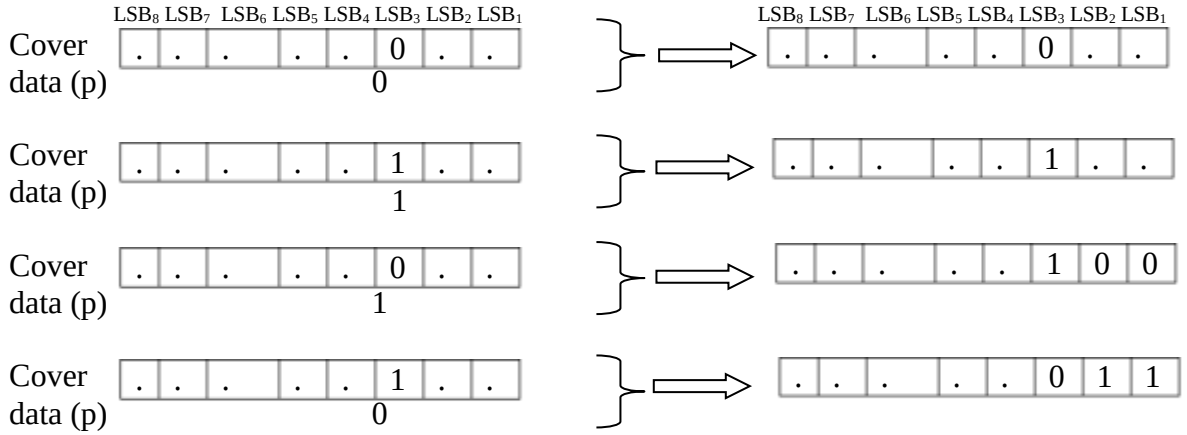
Step 1: Extract LSB 1 set of the color cover image (red color space), LSB 1 = {a_0, a_1,…,a_{65535}}. //first-bit-plane

Step 2: Extract LSB 2 set of the color cover image (red color space), LSB 2 = {b_0, b_1,…, b_{65535}}.// second-bit-plane

Step 3: For I = 1 to L do
If $p_i = c_i$ Then do nothing
Else
{
If $p_i = 1$ and $c_i = 0$ Then
{
$a_i = 0$;
$b_i = 0$;
}
Else If $p_i = 0$ and $c_i = 1$ Then
{
$a_i = 1$;
$b_i = 1$;
}
$c_i = p_i$; // embed message bit in the least-third-bit of the cover
}

In order to explain the above algorithm, let's have the following pixel in the cover image, pixel = $(3)_{10}$ = $(00000011)_2$. Suppose we need to embed p = 1 in the LSB 3, so the new pixel will be, newpixel = $(00000111)_2$ = $(7)_{10}$. Notice that the difference is 7 - 3 = 4. In the proposed algorithm, LSB 1 and LSB 2 will set to 0 when p = 1 and c = 0. So newpixel = $(00000100)_2$ = $(4)_{10}$. Then, the

deference becomes 4 – 3 = 1. On the other hand, suppose that pixel = $(4)_{10}$ = $(00000100)_2$, and p = 0. The newpixel = $(00000000)_2$ = $(0)_{10}$ .The difference is 4 – 0 = 4. In suggested algorithm, in this case, LSB 1 and LSB 2 will set to 1, therefore newpixel = $(00000011)_2$ = $(3)_{10}$. After that, the difference becomes 4 – 3 = 1. Thus, the difference in LSB 3 replacement less or equal one as in LSB 1 but in more robust as the following:



The above steps of algorithm apply three times to embed bits in the least-third-bit of the cover of the color cover image. First, to embed text in the red color space. Second, to embed audio signal in the green color space. Third, to embed binary image in the blue color space.

3. Experimental Results

In this section, a number of experiments which are used to investigate the effectiveness of proposed algorithm will be performed. The algorithm is programmed in MATLAB software 7.6 on core i5 (2.53 GHz) using three color images of size (256×256).

In the experiments, the following text as a message (M) is used. Figure (3) explains the text that need to hide it. The size of embedded text is 6800 bits.

> Robustness refers to the ability of embedded data to remain intact if the stego-image undergoes transformations, such as linear and non-linear filtering, addition of random noise, sharpening or blurring, scaling and rotations, cropping or decimation, lossy compression, and conversion from digital to analog form and then reconversion back to digital form (such as in the case when a hard copy of a stego-image is printed and then a digital image is formed by subsequently scanning the hardcopy). Robustness is critical in copyright protection watermarks because pirates will attempt to filter and destroy any watermarks embedded in images. Anti-watermarking software is already available on the Internet and have been shown effective in removing some watermarks. These techniques can also be used to destroy the message in a stego-image.

Figure (3): The text as message.

Before we hide the message we permute the text using stego-Key to compute (P) from M. Figure (4) explains the message after permutation.

oefufocs ares gs utrm eeoin acir,eg i lninttes ihr mhcaeyttsw te, hvtiuo s Ansde dliaa miwepe dn)lidth shyea sdt esrweyefsatei tet, i,o n-n rnowusub , dyrhdromloutiritdcee nndohu paklaeocadrr nglimfddgc iuottlhard rc rtehp rnfefd rasagsataafntvnaosnf .sl wevtrabyoidoo aoabesettiuetgsed ee.nan gnmmena a keaudtnscrpo hg ,eoepo(- c bpa eigy snmatuusenioessadi oi arnit geegrht mffane r nr ir rind ts sieiamestIocrrnngnsookhmealsw a gnils-f ee amRmoaearnnrstm nasaacdeko annb oqilytskcrisnqianoht ntRlatent ctirafmdae asbitdi o ootgctanshl:nsptasng vi f eoei tooa eeev e ed eg o-iny nc benam ia rrnpoioeysotbncsombsttityaogceae gsiaeaa wrltiaarpo en i cdbodrolRiirrlwsiTsi .lnnpodea ibetmmeemne h teoide ing efaatdihylrgtshe shsboomieaeals - argtcrg.roa mfsthctdns sso,i ngeesn tv on riaolnr prsmicndtbore an csitn et irsim obaeni re

Figure (4): Permutated text.

Also, the following audio signal as a message (M) is used. Figure (5) explains the audio that need to hide it. The size of embedded audio signal is 6800 bits.

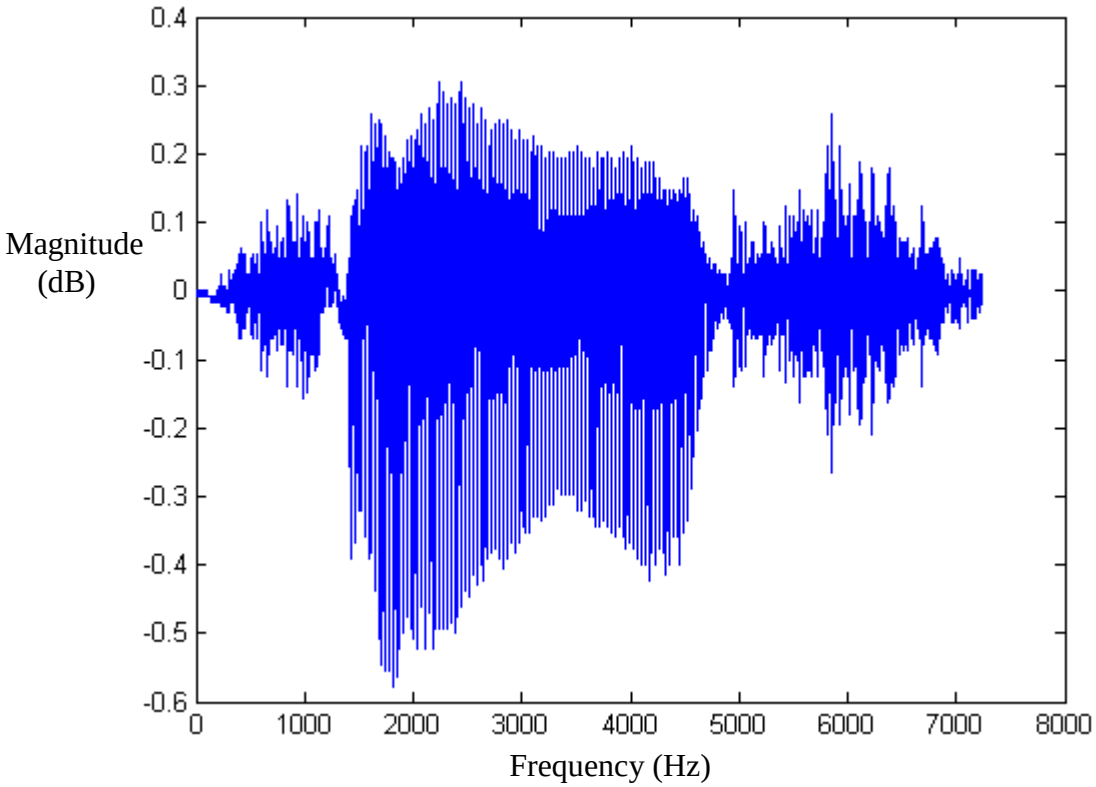


Figure (5): The audio signal as message.

Before we hide the message we permute the voice signal using stego-Key to compute (P) from M. Figure (6) explains the message after permutation.

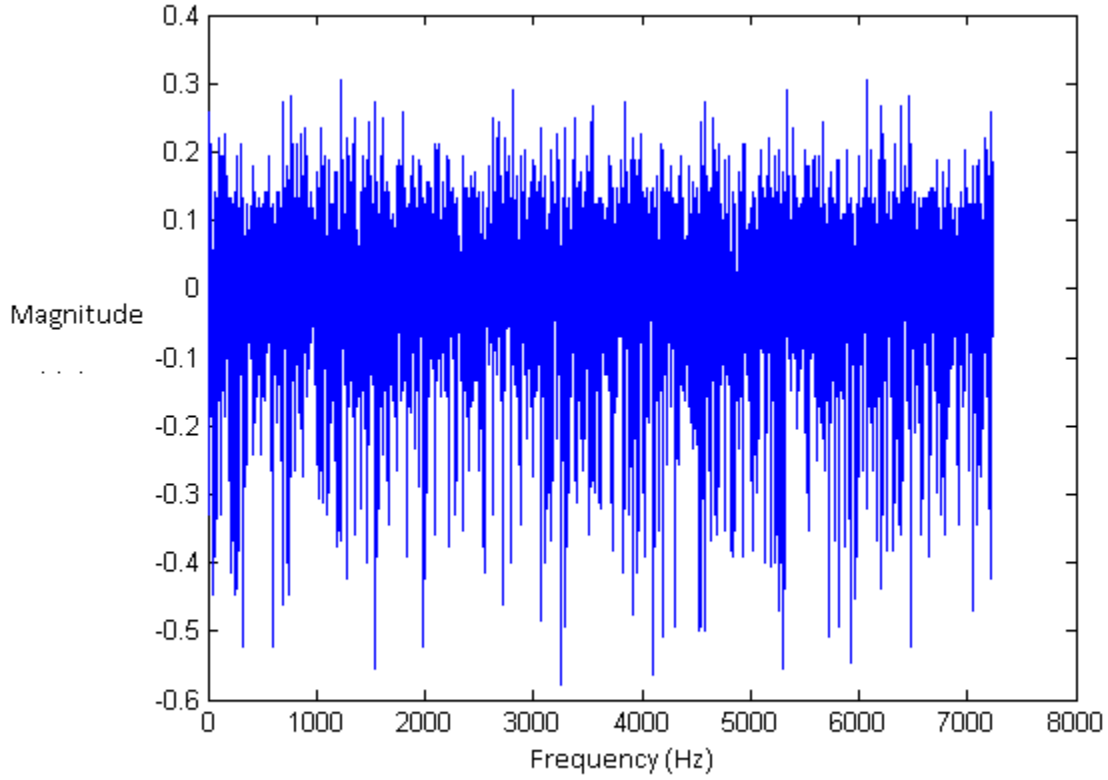


Figure (4): Permutated sound.

In our experiments, we use the Most Significant Bit (MSB) of Lena image (binary image) in the size 256*256 as a message (M). Figure (7) explains the message we need to hide it. The size of message is L=65536 bits.

Figure (7) MSB of Lena image (binary image).

Before we hide the message, we permute the message using stego-Key to compute (P) from (M). Figure (8) explains the message after permutation.

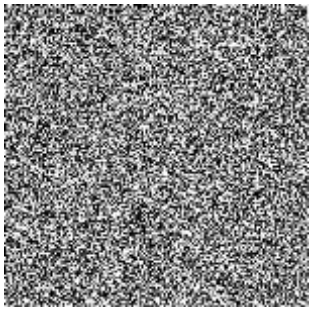


Figure (8) permutated message.

Pepper, mandrill, and boys images of size 256*256 are applied as cover images for comparison. Figure (9) shows the three cover images.

a) b) c)

Figure (9): Color cover images.
a) pepper cover image.
b) mandrill cover image.
c) boys cover image.

To measure the difference between the original cover and stego-image it can use the Peak signal-to-noise ratio (PSNR), which is expressed as the following equation [10]:

$$\text{PSNR} = 10\log_{10}\frac{255^2}{MSE} \qquad \ldots(1)$$

and Mean-Square Error (MSE) is defined as:

$$MSE = \left(\frac{1}{H \times W}\right)\sum_{i}^{H}\sum_{j}^{W}(x_{ij} - x'_{ij})^2. \qquad \ldots(2)$$

where H, W are the size of the cover image (H = 256, W = 256 in this paper), x_{ij}: is the original cover image, and x'_{ij}: is the stego-image.

For color images, the reconstruction of all three color spaces must be considered in the PSNR calculation. The *MSE* is calculated for the reconstruction of each color space. The average of these three MSEs is used to generate the *PSNR* of the reconstructed RGB image. The color PSNR equations are as follows:

$$PSNR = 10\log_{10}\frac{255^2}{MSE_{RGB}} \quad \ldots(3)$$

$$MSE_{RGB} = \frac{MSE_{red} + MSE_{green} + MSE_{blue}}{3} \quad \ldots(4)$$

where MSE_{red} (or green or blue) is similar to Equation (2) for each color space.

We use three experiments which listed here**.**

Experiment I

In this experiment, the LSB 3 method is used to embed the message (P) in the three covers separately without any modification to LSB 1 and LSB 2 of the color cover images. We obtain the following results, as shown in Table (1).

Table (1): Results of experiment I.

Cover image	PSNR (dB)
pepper	43.8626
mandrill	43.8630
boys	43.9254

Experiment II

In this experiment, LSB 3 is used to embed the message (P) in the three color covers separately, but with modifying the LSB 1 and LSB 2 of the cover image as seen in Section (2.1). We obtain the following results, as shown in Table (2). Figure (10) explains the three stego-images after embedding the message.

Table (2): Results of experiment

Cover image	PSNR (dB)
pepper	51.8999
mandrill	51.8288
boys	51.8562

Figure (10): Stego-images with LSB 3 substitution.

Experiment III

Evaluate the transparency of embedded data of the proposed method, we are increased in the amount of bits of data which are different of whose embedded, according to the quality of the cover. Figure (11) shows PSNR versus the number of embedded bits.

(a) (b) (c)

Figure (11): PSNR versus the number of the embedded bits for color pepper image.
(a) 904 bits, PSNR = 62.9370 dB.
(b) 2840 bits, PSNR = 57.8397 dB.
(c) 6800 bits, PSNR = 51.8999 dB.

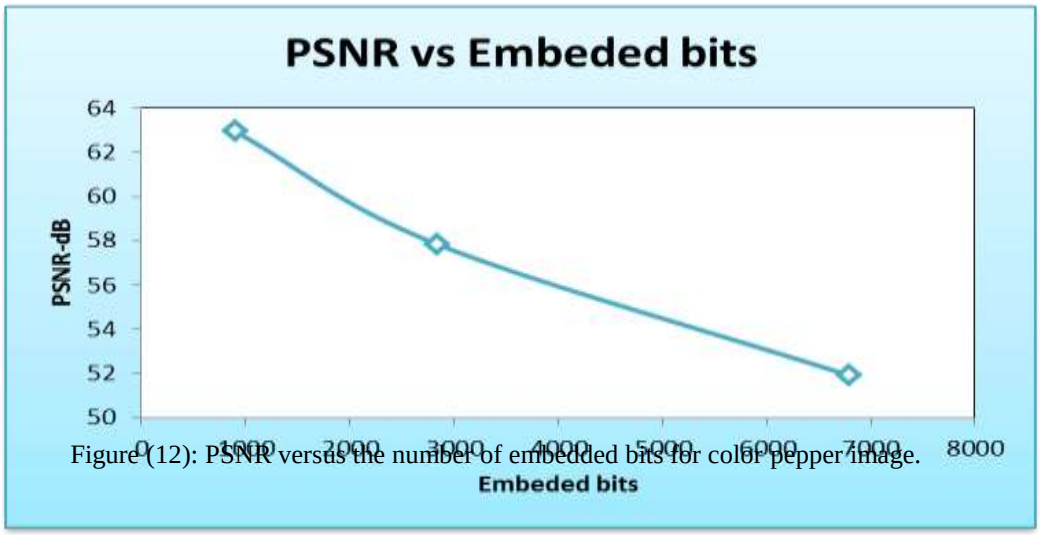


Figure (12): PSNR versus the number of embedded bits for color pepper image.

Experiment IV

In this experiment, to prove the security of proposed method by using an attack method, the stego-image has been compressed by using the hard-compression technique, which depends on the wavelet transform. The stego-image compressed in 80 threshold value. Figure (13) explain the compressed cover in 80 threshold, which explain the resistance of the proposed scheme against compression attack**.**

(a) (b)

Figure (13): (a) original image, (b) Compressed stego-imge image in 80 threshold value.

4. Conclusions

In this research, we have presented a suggested technique for data hiding. This work presents a new spatial domain data hiding method used for steganography applications. Our method of embedding message in the LSB 3 color image cover, and modifying LSB 1 and LSB 2 of the color cover, minimized the difference between the old values of the cover pixels and the stego-images. This minimization (increasing PSNR) leads to provide high secret communications, so the attacker cannot notice the difference between the stego-image and the original cover.

Based on the results obtained and detailed in the previous sections, it can include that:

- Steganography is the practice of encoding secret information in a manner such that the very existence of the information is concealed. Moreover, to detect the message's existence will be very hard for those stego-images.
- The proposed technique can be defined as a secret key steganography since it shares a secret key (stego-Key) between sender and receiver, in this technique there is no need for the knowledge of original cover in the extraction process.
- In Figure (11), both the PSNR values of the stego-image and the original cover image result in a trade-off problem. That is, we find that as the size of embedded data is increased, the PSNR is decreased and vice versa, as shown in Figure (12).
- However, the amount of information can actually be hidden in an image that depends upon the composition of the image. An image that contains high frequency areas can be manipulated (not make sense from the viewpoint of human eyes) more than an image containing primarily low frequency areas (high noticeable) of the stego-image as shows in Table (2).
- Figure (13) explain the compressed cover in 80 threshold, which explain the resistance of the proposed scheme against compression attack.
- The proposed model has proved to be easy to use and efficient in terms of security.

5. References

[1] Marvel L. M., Boncelet C. G., and Retter C. T., *"Spread Spectrum Image Steganography"*, IEEE Trans. Image Processing, pp. 1075-1083, Aug. 1999.

[2] Voyatzis G., Nikolaidis N., and Pitas I., *"Digital Watermarking: An Overview"*, EUSIPCO, Vol. 1, pp. 9-12, 1998.

[3] Johnson N., and Jajodia S., *"Exploring Steganography: Seeing the Unseen"*, IEEE Computer, pp. 26-34, February 1998.

[4] Bender W., Gruhl D., Morimoto N., and Lu A., *"Techniques for Data Hiding"*, IBM Systems Journal, Vol. 35, No. 3 and 4, pp. 313-336, 1996.

[5] Swanson M., Kobayashi M., and Tewfik A., *"Multimedia Data Embedding and Watermarking Technologies"*, In Proceedings of the IEEE, Vol. 86, No. 6, pp. 1064-1087, June 1998.

[6] Wolfgang R. B., Podilchuk, and Delp E. J., *"Perceptual Watermarks for Images and Video"*, In the Proceedings of the IEEE, May 1999. (A copy of this paper is available at http://www.ece.purdue.edu/~ace).

[7] Wolfgang R. B. and Delp E. J. , *"Fragile Watermarking Using the VW2D Watermark",* In Proceedings of the SPIE/IS&T Conference on Security and Watermarking of Multimedia Contents, Vol. 3657, San Jose, CA, January 1999.

[8] UnZign software, http://altern.org/watermark, 1997.

[9] Stirmark software, http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark, 1997.

[10] Katzenbeisser S., and Farbin A. P, *"Information Hiding Techniques for Steganography and Digital Watermarking"*, Artech House, Boston-London, February 2000.

[11] Bender W., Butera W., Gruhl D., Hwang R., Paiz F. J., and Pogreb S., *"Applications for Data Hiding"*, IBM Systems Journal, Vol. 39, No. 3 and 4, 2000.

[12] Wayner P., *Disappearing Cryptography 3rd Edition: Information Hiding: Steganography & Watermarking.* Amsterdam: MK/Morgan Kaufmann Publishers. ISBN 978-0123744791, 2009.

[13] Johnson N. F., *"Digital Watermarking and Steganography: Fundamentals and Techniques"* , The Computer Journal, 2009.