Hiding Data Using LSB-3

Aiad Ibraheem Abdul-Sada
Dept. of Computer Science, College of Education, University of Basrah, Basrah, Iraq.

ISSN –1817 –2695
(Received 5/3/2007 , Accepted 19/9/2007)

ABSTRACT

In this paper, a new steganography method based on the spatial domain is proposed. Instead of using the LSB-1 of the cover for embedding the message, LSB-3 has been used to increase the robustness. LSB-1,2 may be modified according to the bit of the message, to minimize the difference between the cover and the stego-cover. For more protection to the message bits a stego-Key has been used to permute the message bits before embedding it. Experimental results of the modified method shows that PSNR is grater than the conventional method of LSBs replacement.

Keywords: Image, Steganography, LSBs, PSNR, PRNG.

1. INTRODUCION

1.1 Motivation

In conventional cryptography, even if the information contents are protected by encryption, the existence of encrypted communications is known. In view of this, digital steganography provides an alternative approach in which it conceals even the evidence of encrypted messaging. Generally, steganography is defined as the art and science of communicating in a covert fashion [1]. It utilizes the typical digital media such as 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 this way, 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 shown 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 technique is strongly dependent on the structure of the cover, and in this paper 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. In addition, 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.

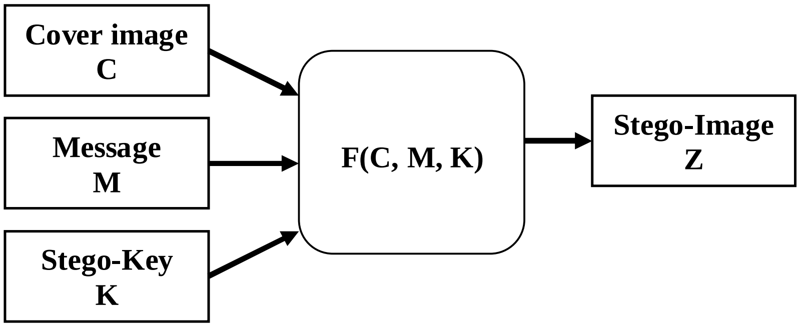


Figure (1): Steganographic Encoding

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

1.2 Applications

There are many 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 able 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 many situations, 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. distortion of the cover image. 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 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 [5, 6]. Anti-watermarking software is already available on the Internet and have been shown effective in removing some watermarks [8, 9]. These techniques 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) Replacement

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]. 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 method also generate a unbalanced distribution of the changed pixels, because the message is embedded at the top of the image. In the next section, a modified method will be proposed.

2. THE PROPOSED METHOD (LSB-3)

In this paper, a 256*256 gray level image has been used as a cover. So, we can hide a message up to 65536 bits (8192 bytes). The message in the LSB-3 of the cover is embedded to increase the robustness of the system and protect the message against the external influences such as noise, filter, compression,…etc.

Lets have the message bits set M={m_0, m_1, m_2, …,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,..,L-1. Lets have the cover image={$pixel_0$, $pixe_l$,…, $pixel_{65535}$}. Suppose that LSB-3 of the cover image is LSB3={c_0, c_1, c_2, …,c_{65535}}, where c_j={0,1} for each j=0,..,65535. 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) shows the block diagram of message bits permutation.

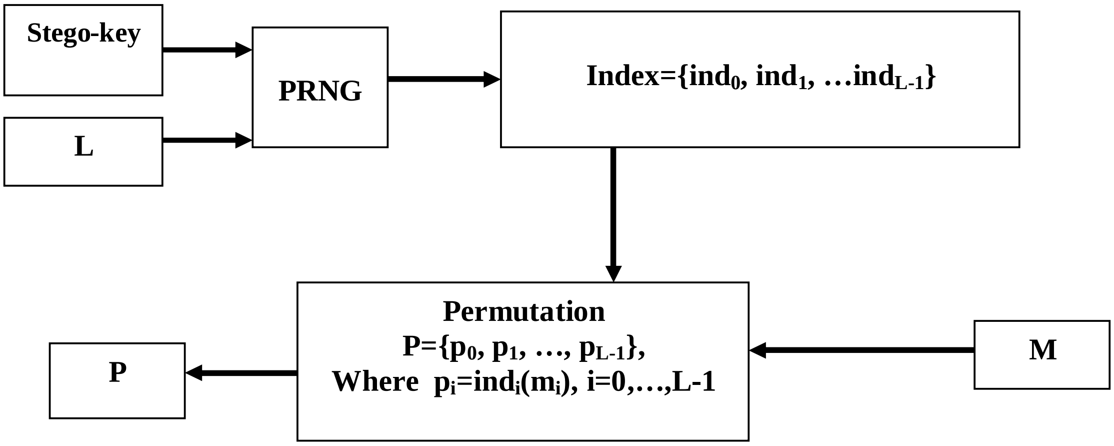

Stego-key
L
PRNG
Index={ind$_0$, ind$_1$, …ind$_{L-1}$}
Permutation
P={p$_0$, p$_1$, …, p$_{L-1}$},
Where p$_i$=ind$_i$(m$_i$), i=0,…,L-1
M
P

Figure (2):Message Bits Permutation.

The embedding process is very easy, which is only replace the permutated bits 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:-

Embedding Algorithm

```
Step 1: Extract LSB-1 set of the cover image, LSB1={$a_0$, $a_1$,…,$a_{65535}$}. //first plane
Step 2: Extract LSB-2 set of the cover image, LSB2={$b_0$, $b_1$,…, $b_{65535}$}.// second 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 third bit of the cover
              }
```

To explain the above algorithm, lets 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 our algorithm, we will set LSB-1,2 to 0 when p=1 and c=0. So newpixel=$(00000100)_2$=$(4)_{10}$. As you see 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 our algorithm, in this case, we will set LSB-1,2 to 1, so newpixel=$(00000011)_2$=$(3)_{10}$. As you see 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.

3. Experimental Results

In our experiments, we use the Most Significant Bit (MSB) of Lena image in the size 256*256 as a message (M). Figure (3) explains the hidden message . The size of message is L=65536 bits.

Figure (3) MSB of Lena image.

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

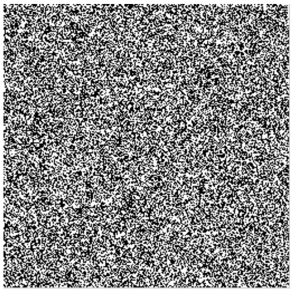
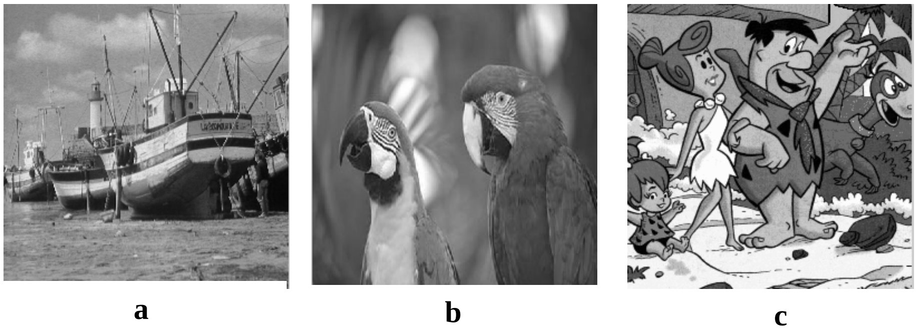


Figure (4) permutated message.

Boat, birds, and flinstone images of size 256*256 are applied as cover images for comparison. Figure (5) shows the three cover images.



a **b** **c**

Figure(5): cover images.
a-boat cover image
b-bird cover image
c-flinstone cover image

To measure the difference between the original cover and stego-image we use the Peak Signal to Noise Ration (PSNR), which expressed as the following equation [10]:-

$$\text{PSNR} = 10\log_{10}\frac{255^2}{MSE} \qquad \textbf{...1}$$

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

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

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. We use two experiments which listed here:-

Experiment 1

In this experiment, the LSB-1 has been used to embed the message (P) in the three covers separately [10]. Table(1) shows the result of this experiment. LSB-1 has high PSNR but low robustness.

Table(1): Results of experiment 1

Cover image	PSNR
Boat	51.0859 dB
Bird	51.1334 dB
Flinstone	51.3396 dB

Experiment 2

In this experiment, the LSB-3 method is used to embed the message (P) in the three covers separately without any modification to LSB-1,2 of the cover images. Table (2) shows the results for different covers.

Table (2): Results of experiment 2

Cover image	PSNR
Boat	39.1132 dB
Bird	39.0955 dB
Flinstone	39.1188 dB

Experiment 3

In this experiment, LSB-3 is used to embed the message (P) in the three covers separately, but with modifying the LSB-1,2 of the cover image according to our proposed algorithm. We obtain the following results, as shown in Table(3). Figure (6) explains the three stego-images after embedding the message.

Table (3): Results of experiment 3

Cover image	PSNR
Boat	42.4163 dB
Bird	42.4062 dB
Flinstone	42.2932 dB

Figure (6): stego-images with LSB-3

4. Conclusions

Our method of embedding message in the LSB-3 image cover, and modifying LSB-1,2 of the 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 notices the difference between the stego-image and the original cover.

References

[1] L. M. Marvel et al., “Spread spectrum image steganography”, IEEE Trans. Image Processing, pp. 1075-1083, Aug. 1999.

[2] G. Voyatzis et al., “Digital watermarking: an overview”, EUSIPCO, Vol. 1, pp. 9-12, 1998.

[3] N. Johnson and S. Jajodia, “Exploring steganography: seeing the unseen”, IEEE Computer, pp. 26-34, February 1998.

[4] W Bender, D. Gruhl, N. Morimoto, and A. Lu, “Techniques for data hiding”, IBM Systems Journal, Vol. 35, No. 3 and 4, pp. 313-336, 1996.

[5] M. Swanson, M. Kobayashi, and A. Tewfik, “Multimedia data embedding and watermarking technologies”, Proceedings of the IEEE, Vol. 86, No. 6, pp. 1064-1087, June 1998.

[6] R. B. Wolfgang, C. Podilchuk and E. J. Delp, “Perceptual watermarks for images and video”, to appear in the Proceedings of the IEEE, May 1999. (A copy of this paper is available at: http://www.ece.purdue.edu/~ace).

[7] R. B. Wolfgang and E. J. Delp, “Fragile watermarking using the VW2D watermark”, 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] S. Katzenbeisser, Farbin, A. P,"Information Hiding Techniques for Steganography and Digital Watermarking", Artech House, Boston - London, February 2000.

إخفاء البيانات باستخدام طبقة LSB-3

أياد إبراهيم عبد السادة

قسم علوم الحاسبات, كلية التربية, جامعة البصرة, البصرة, العراق.

المستخلص

تم في هذا البحث, تصميم طريقة جديدة لإخفاء رسالة (سلسلة من الثنائيات) في المجال المكاني لغطاء الصورة. الطريقة المقترحة تعتمد على تخزين ثنائيات الرسالة في الطبقة الثالثة LSB-3 من ثنائيات الصورة بدلا من الطبقة الأولى LSB-1, وذلك لزيادة متانة بيانات الرسالة داخل الغطاء. الهدف الأساسي لهذا البحث هو تقليل الفرق بين قيم الغطاء قبل وبعد إخفاء بيانات الرسالة فيه, عن طريق تعديل ثنائيات الطبقة الأولى والثانية LSB1,2 من الغطاء, وذلك يؤدي إلى زيادة سرية الاتصال بين المرسل والمستلم وصعوبة ملاحظة المهاجم للتشوه في بيانات الغطاء الناتجة عن عملية الإخفاء. ولمزيد من الحماية لبيانات الرسالة قمنا ببعثرتها باستخدام مفتاح− إخفاء قبل أن يتم إخفائها في الغطاء. النتائج أظهرت أن الطريقة المستخدمة أعطت قيمة تشابه اكبر PSNR بين الغطاء قبل وبعد الإخفاء أكثر من التشابه في الطريقة التقليدية للإخفاء.

الكلمات المفتاحية: صورة, كتابة مخفية, PRNG, PSNR, LSBs .