Text in Image Hiding using Developed LSB and Random Method

Elaf Ali Abbood, Rusul Mohammed Neamah, Shaymaa Abdulkadhm
Computer Department, Science College for Women, University of Babylon, Babylon, Iraq

Article Info

Article history:

Received Jan 2, 2018
Revised Mar 10, 2018
Accepted Mar 16, 2018

Keyword:

LSB method
PSNR and SSIM metrics
Random number generation
Security
Steganography

ABSTRACT

Information Hiding is a task that face difficult challenges in current time. The reason for these challenges is the rapid development of methods of detection of hidden information. So, researchers have been interested in developing methods of concealment, making it difficult for attackers to access hidden information using new methods of concealment. Such as the introducing a complex algorithms, use a random methods and invent more complicated and difficult steps. This paper presents a new method of hiding information within the image. This method creates a new sequence of mysterious and difficult steps by dividing the secret text on all image and random distributing of bits to each row. Then using a special reverse method to hide the bits in that row. The LSB method has also been developed to make it more difficult to hide the pixel. The results presented illustrate the strength and security of the method and provide greater protection for hidden information. Also, the result illustrate the quality of the stego image compared with the original image using PSNR and SSIM quality measures.

Copyright © 2018 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Elaf Ali Abbood,
Computer Department,
Science College for Women,
University of Babylon,
Babylon, Iraq.
Email: wsci.elaf.ali@uobabylon.edu.iq

1. INTRODUCTION

The transfer of important and confidential data over the Internet has become one of main challenges with all this development in information technology and communication. Today, confidential data can be ensured by various methodologies of concealing data. Cryptography, steganography, and watermarking are three general techniques to conceal information. One of them, hideing the existence of a message, and the other means hiding information as a media format such as image, audio, video, and even a text so that other people do not notice the existence of information in an abovementioned format. And finally, watermarking means to protect copyright. In recent years, approachs to conceal information have paid great attention to steganography and watermarking techniques [8].

One of the best techniques for secure communication is steganography a covert writing. Steganography is the science of invisible communication which hides any private data within an innocent-looking cover object. The aim is to design a steganography algorithm which not only hide the message behind the image, but also provide more security than others[1, [2]. Steganography techniques, on the other hand, hide the existence of the secret message itself, which makes it cumbersome for a third person to discover the message [1]. In image steganography, the information is hidden exclusively in images. Today, steganography is mostly used on computers with digital data being the bearers and networks being the rapid conveyance channels [3]. Figure 1 illustrates the common behavior and the main concepts for the steganography technique.

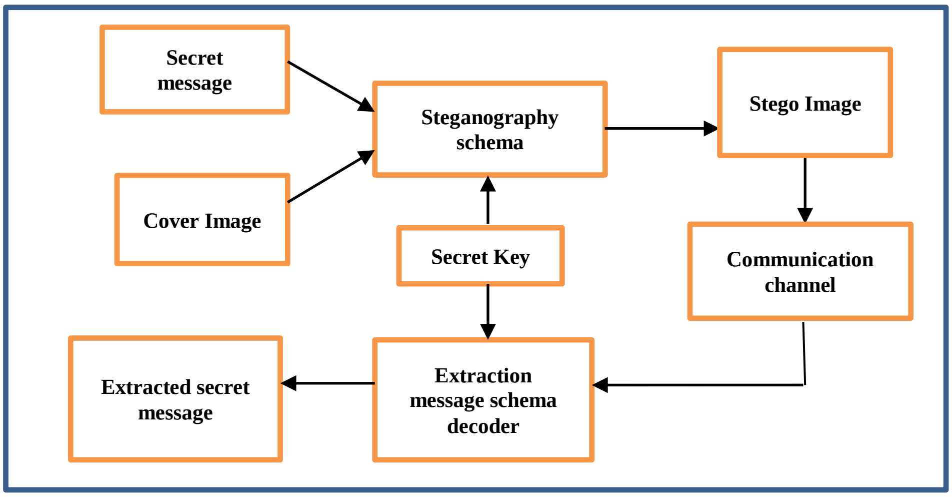


Figure 1. Basic concepts and behavior of steganography

The steganography includes to main steps there are hiding and the extracting information. In the hidding step, the secret text will be embedded in a selected location i nthe cover image based on a suitable steganography method. Then, the resulting stego-image is sent to the receiver. In the extracting step, the receiver applying the extraction function to recovery the secret text [12].

Steganography differs from cryptography in the sense that where cryptography concentrates on keeping the substance of a message mystery, steganography concentrates on keeping the presence of a message mystery [3], [4]. There are many techniques related to steganography is fingerprinting. In fingerprinting unique marks are embedded in distinct copies of the carrier object that are supplied to different customers [6].

The output image is called stego-image that is similar to the cover media. This stego-image is then sent to the receiver where the receiver retrieves the hidden message by implementing a de-steganography process. A stego-key is used for an embedding or encoding process to limit unraveling or extraction of the embedded data in cover media [5].

The stego-key depending on random generation uses a specific kernel in order to send it to the recipient to generate the same key in order to retrieve the hidden data and data is embedded in a cover image using LSB method. Least-Significant-Bit (LSB) is one of the popular and frequently used steganography techniques to conceal a mystery message in a digital medium [7].

The method recommended in this paper is used a random key generator method for improvement in robust and security of steganography. The random number generator locates the hiding positions in the cover image in each row separately and each row will hide the same number of bits of secret text. That’s by dividing the secret text on all the rows of the cover image. specific method for hiding bits in each pixel will be used.

2. RELATED WORKS

In a steganography field, there are many researchers using a random number generation method combined with another technique for hiding information in image. Obaida M. and Awad A. introduced that the secret text bits are inserted into the cover image pixels randomly. Whereas, inserting of secret text bits into the cover image will be in any bit of the pixel randomly by comparing the message bit with the pixel bit that chosen randomly from the second to the last bit. The least significant bit of the pixel will be 1 if the comparison result is matched. And the least significant bit of the pixel will be 0 if the comparison result is not [3].

Babita el al used a random key generating method to encrypt the secret message using XOR method then embedding the cipher message into an image of RGB format in a special arrangement. The resulted RGB image converted into the bitmap file format [9]. Balvinder el al. encrypted the secret text using XOR encryption method using 8-bit random key. Then, applied XOR operation between one bit of 8-bit random key and 2nd LSB of cover image pixel. If the XOR result of above operation is 1 then hide one bit of the secret key into LSB of same pixel of cover image. Otherwise, there is no hiding in any bit in that pixel. The substitution process will be continued depending on the length of encrypted message. The random key hidden in the same first byte of cover image [10].

Noor K. hidden information by generate the secret key using LFSR method then encrypted the secret message using AES method. The encrypted message distributed over the cover image pixels using permutation technique. Then hide the bits of the encrypted message in cover image in the randomly selected pixels [11]. Ashwini and komal provided sequential encoding and random encoding ways for embedding the secret data inside cover image. Sequential encoding used one key entered by user and sequentially selected the pixel for embedding cipher text obtained by XORing the secret message with key provided by user. In random encoding two keys are provided by the user ,one is to lock the function on receiver side and other for to carry out encryption. In this case pixels are selected randomly for embedding secret data i.e either text or image using random number generator which automatically implements link list concept [1].

The technique that used by Rupali B. and Vaishali S. included three main steps. That's by found complementing of the secret text. Then, hiding the complemented text in cover image pixels using pseudo random number generator, finally, hide the bits of complemented text in each pixel using the inverted bit LSB method [5]. Meenakshi S. and *et al* introduced an RGB image embedding method based on sixteen-pixel differencing with n-bit Least Significant Bit (LSB) substitution. In their method, the image is divided into 4×4 non intersecting blocks then and calculate the average difference value in each block. Based on the resulted value the block is classified to fall into one of four levels such as, lower, lower-middle, higher-middle and higher. If block belongs to lower level then 2-bit LSB substitution is used in it. Similarly, for lower-middle, higher-middle and higher level blocks 3, 4, and 5 bit LSB substitution is used [13]. In this paper, a new method is proposed for hiding secret text in image depending on the generation of random number and developed LSB method using a simple hash function.

3. THE PROPOSED METHOD

This paper introduces a new proposed method to hide a secret text in gray image using random technique with a secret key and simple hash function. The allocation of the secret text into the image depends on the length of secret text. Where, the secret text will be divided on all rows of image in equal manner excepted the last row, that need to hide additional information. In each row, there is a fixed number of columns that needs to hide a secret text bits, that’s chosen randomly. The hiding of bit in column depends on a special schema. Flow chart in Figure 2 shows the sequence of the method.

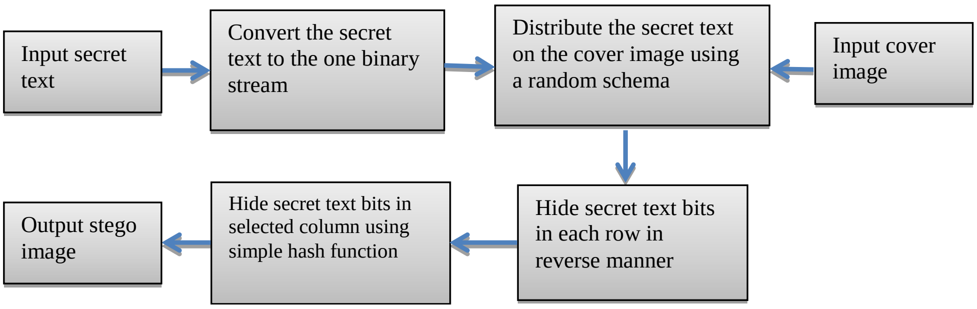

Input secret text
Convert the secret text to the one binary stream
Distribute the secret text on the cover image using a random schema
Input cover image
Hide secret text bits in each row in reverse manner
Hide secret text bits in selected column using simple hash function
Output stego image

Figure 2. Proposed method processed stages

The proposed method steps depicted in the following:

Step 1 (Convert the secret text to the one binary stream): The secret text is a string consisting of letters, spaces and special characters. Each character in secret string converts to seven bits. When merge all the characters, we will have one stream of binary bits. For example: If the secret text is "To", when converting it to binary and combining the two letters, the resulting string will be "10101001101111".

Step 2 (Distribute the secret text on the cover image): The secret text is distributed on cover image evenly across all rows except the last one. This is done by dividing the number of secret text bits on the number of image rows. This is computed as in Equation (1).

$$no\_col = no\_textbits / (no\_row-1) \tag{1}$$

Where no_row represents the number of image's rows, no_col represents the number of columns that needed to hide the secret stream in each row. And no_textbits denotes to the length of binary string of secret text.

Step 3 (Determine the random columns locations in each row): In each row, we used a pseudo random number generator to determine the location of columns that need to hide the secret text in each row. This is done by the following:

Step 3-1 Create a seed to generate a random numbers. And to more security, this seed is created using shuffle function. This seed will represent part of the secret key that will hide in last row of the image and sent to the receiver. Step 3-2 For each row, generate a no_col of columns randomly within a certain range and without duplicating in that row:

```
Flag=false;
while~Flage
R=ceil(n*rand(no_col,1))
Flage=numel(unique(R))= =no_col
End
```

Where, R represents the resulted array that's contains the columns' indexes. ceil is a function rounds the elements of A to the nearest integers greater than or equal to A, and n refer to the total number of columns in cover image. rand refers to random generation function. numel represents a function returns the array elements number. unique is a function that’s delete the duplicating in array.

Step 4 (Hide one bit in each pixels have location x,y) Where x represents the current row, and y represents the column that is selected randomly. This hiding method will be in reverse manner for the current row. For example, if the number of columns allocated for each row is 10 columns, then there are 10 bits of secret text that would be hidden in each row. Where, the tenth bit will hide in the first column and the ninth bit will hide in the second column and so on.

Step 5 Each pixel have 8 bits. To hide one bit in one pixel, we will use the updated LSB method that’s including a simple hash function. This method includes hiding a bit in one of the less three significance bits in the current pixel. The chosen among the three bits done using a simple hash function that used mod operation for the index of the current random selected column. As depicted in Equation (2).

$$x = \text{len\_pxl} - (\text{col\_indx} \bmod 3) \qquad (2)$$

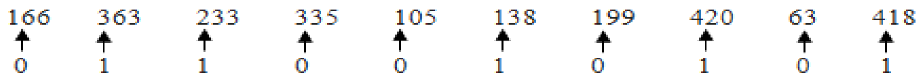
Where, x represent the location of hiding the bit in the pixel, len_pxl depicted to the length of pixel that will be 8 bits, col_indx denoted to the index of the current random selected column. If the resulted x is equal to 0, 1 or 2, then the bit will hide in the eighth, seventh or sixth bit of the pixel, respectively. The following example illustrates the above steps of our method in details.

Let's the secret text have 700 characters. When converting it to binary, each character will convert to 7 bits. That means, the length of secret text in binary will be 7* 700=4900 bits. For the cover image with size 512*512 pixel, each row will contain number of bits of secret text computed as following: no_col=floor (4900/(512-1))+1=10 bits of the secret text hidden in each row of the cover image. Where floor(A) rounds the elements of A to the nearest integers less than or equal to A.

Now, we use the random function to generate 10 columns' indexes in range (1-512). First, generate the start random seed using shuffle function for example: seed=1897039246. This seed will be considered as secret key that's sent to receiver by hiding it in the last row of cover image using LSB method. The receiver will use this seed to generate the same sequences of random numbers to get the secret text. Then, generate 10 column indexes in range (1-512) for each row in cover image except the last one using step (3-2). When n=512, no_col=10, the first 10 columns indexes generated in our example will be:

165 363 233 335 105 138 199 420 63 418

If the first 10 bits in secret text are: 1010100110, then the hiding of these bits in the first row and random selected columns will be in reverse manner as:



That means the 10th secret bit will hide in the 166th column and the 9th bit will hide in 363th column, and so on. Now, we determined the pixel(row, column) from cover image for each bit in secret text. To hide the 10th bit (0) in the pixel(1,166), we use mod operation as a simple hash function as in step (3-4) to find the index of hiding the secret bit in the pixel bits. If that index (x) is computed as:

$$x = \text{len\_pxl}-(\text{col\_indx} \bmod 3) \longrightarrow x=8-(166 \bmod 3)=7$$

Then, the 10th secret bit (0) will hide in the 7th bit of the pixel (1,166). If the gray color of the pixel (1,166)= 0 1 0 1 0 0 1 1, then the 7th bit of the pixel (1) will be replaced by the 10th secret bit (0) of the secret text. The result of the pixel (1,166)=0 1 0 1 0 0 0 1.

When the stego image arrives to the receiver, the receiver will be extracted the secret text by applying the same hiding steps. In the following, the steps of extraction the secret text:

Step 1: Extraction of the seed random generation and length of the secret text. First the receiver must extract the seed of generate the random sequence and also extract the length of secret text that hide in the last row of the stego image using LSB traditional method.

Step 2: Find the number of columns that used to hide the secret text bits using Equation (1).

Step 3: Generate a random locations for the number of columns in each row. That's done using a pseudo random generation method started using the seed that extracted in step 1.

Step 4: Extraction of secret text bits. To extract the secret text bits, we use the same method to hide these bits in image. Where, the first bit will be extracted from the last random column and the second bit will be extracted from the second last random column and so on. To extract the bit from the pixel, we use the mod operation using Equation (2).

Step 5: Convert the secret text bits to characters. The extracted secret text is one sequence of binary bits. To covert the secret text from binary to characters, each seven bits will convert to one character.

4. RESULTS

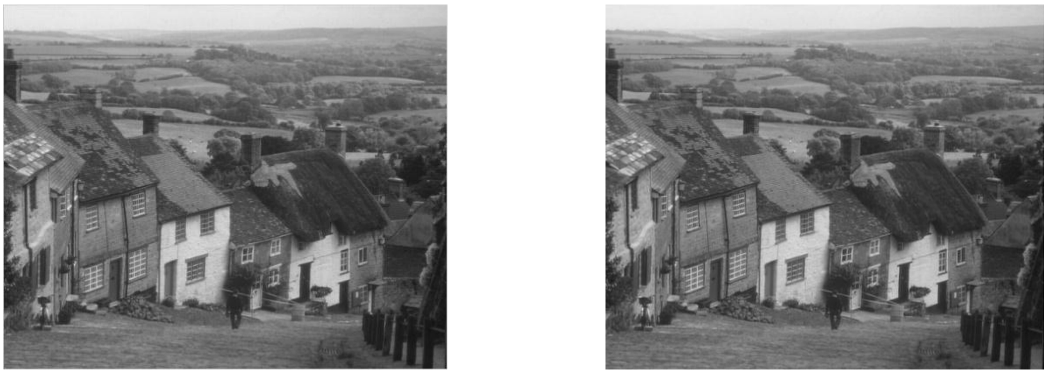
The proposed method used an efficient and secure schema to hide secret text in image. This section illustrates the results of applying the proposed method to hide different secret text size on different images and measure the accuracy of the resulted images using PSNR and SSIM measurements that's describes in Equation (3) and Equation (4), respectively.

$$\mathrm{PSNR} = 10\log_{10}\left[\frac{255^2}{\frac{1}{MN}\sum_i\sum_j(y_{ij}-x_{ij})^2}\right] \quad (3)$$

Where y_{ij} indicated to the strego image pixels and x_{ij} indicated to the cover image values. M and N is the image size.

$$\mathrm{SSIM}(m,n) = \left(\frac{2\mu_m\mu_n+d_1}{\mu_m^2+\mu_n^2+d_1}\right)\times\left(\frac{2\sigma_{mn}+d_2}{\sigma_m^2+\sigma_n^2+d_2}\right) \quad (4)$$

Where µ denoted to the mean of the (m, n) window. σ are indicated to the standard deviation of (m, n) window. Figure 3, Figure 4 and Figure 5 illustrated the effect of the proposed method on 512x512 Goldhill, Boat and Lena images when hiding a secret text with size 4900, 14700 and 24500 bits, respectively.



(a) The original goldhill image

(b) The stego image

Figure 3. Hiding a secret text with size 4900 bits in 512x512 Goldhill image

(a) The original boat image (b) The stego image

Figure 4. Hiding a secret text with size 14700 bits in 512x512 Boat image

(a) The original lina image (b) The stego image

Figure 5. Hiding a secret text with size 24500 bits in 512x512 Lina image

Table 1 shows the illustrated the results of PSNR and SSIM measurements when applying the proposed method on different images and different size of secret text.

Table 1. Illustrated the results of PSNR and SSIM measurements when applying the proposed method on different images and different size of secret text

Cover image	Secret text 4900 bits		Secret text 14700 bits		Secret text 24500 bits	
512x512	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Goldhill	60.0489	0.99966	55.2512	0.99896	52.9806	0.998234
Boat	60.0056	0.999513	55.2738	0.998554	52.9859	0.997577
Lena	60.0299	0.999444	55.197	0.998328	52.9817	0.997242

5. COCLUSIONS

The main objective of developing methods of information concealment is to increase the security of these roads and to protect information from detection by attackers. The more complex the method of concealment and the more steps followed, the more cynical the method and harder for the attackers to break it and access to hidden information. This search uses more than one technique to increase security of concealment and to strengthen the protection of hidden information. Where the method of random generation in the selection of pixels in each row of the image in which we hide the secret text. Reverse inversion of bits of secret text was used in each row. It also used an improved LSB method which hid the bit of the secret text in three less important bits. And selects the bit in which we will hide the random numbers generated earlier.

These steps increase the security of the method and give good results in terms of measuring the impact of the image after the concealment compared to the original image.

REFERENCES

[1] Ashwini B. and Komal B., "Hybrid Approach for Embedding Text or Image in Cover Images", *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 5, no. 5, 2016.

[2] Yang Ren-er and *et al*, "Image Steganography Combined with DES Encryption Pre-processing", *Sixth International Conference on Measuring Technology and Mechatronics Automation*, pp. 323-326, 2014.

[3] Obaida Mohammad Awad Al-Hazaimeh, "Hiding Data in Images Using New Random Technique", *International Journal of Computer Science Issues*, vol. 9, 2012.

[4] Artz, D., "Digital Steganography: Hiding Data within Data", *IEEE Internet Computing Journal*, 2001.

[5] Rupali Bhardwaj and Vaishali Sharma," Image Steganography Based on Complemented Message and Inverted bit LSB Substitution", *6th International Conference on Advances In Computing & Communications*, 2016.

[6] M.A.Wakure and S.A.Wakure, "A Digital Image Steganography", *International Journal of Computer Science Trends and Technology (IJCST)*, vol. 5, no. 2, 2017.

[7] Ebrahim Alrashed and Suood Suood Alroomi, "Hungarian-Puzzled Text with Dynamic Quadratic Embedding Steganography", vol. 7, no. 2, 2017.

[8] Reihane Saniei and Karim Faez, "The Security of Arithmetic Compression Based Text Steganography Method", *International Journal of Electrical and Computer Engineering*, vol. 3, no. 6, pp. 797-804, 2013.

[9] Babita1 and *el al*, "An Approach to Improve Image Steganography using Random Key Generation Method", *International Journal of Information and Computation Technology*, vol. 3, no. 4 , pp. 235-240, 2013.

[10] Balvinder Singh and *el al*, "A Steganography Algorithm for Hiding Secret Message inside Image using Random Key", *International Journal of Engineering Research & Technology*, vol. 3, no. 12, 2014.

[11] Noor Kareem Jumaa, "Hiding of Random Permutated Encrypted Text using LSB Steganography with Random Pixels Generator", *International Journal of Computer Applications*, vol. 113, no. 13, 2015.

[12] Mojtaba B. and Karim F., "An Adaptive Steganography Scheme Based on Visual Quality and Embedding Capacity Improvement", *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 4, no. 4, pp. 573-584, August 2014.

[13] Meenakshi S Arya, Meenu Rani, Charndeep Singh Bedi, "Improved Capacity Image Steganography Algorithm using 16-Pixel Differencing with n-bit LSB Substitution for RGB Images", *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 6, pp. 2735-2741, December 2016.