International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 6, No.5 (Sep-2017)

http://dx.doi.org/10.12785/ijcds/060505

Steganography Algorithm within 2-LSBs with Indicators-Based Randomness

Tawfiq Barhoom[1] and Wesam Saqer[2]

[1] *Faculty of Information Technology of the Islamic University of Gaza, Gaza, Palestine*
[2] *Faculty of Information Technology of the Islamic University of Gaza, Gaza, Palestine*

Received 20 Mar. 2017, Revised 22 Apr. 2017, Accepted 3 Jul. 2017, Published 1 Sep. 2017

Abstract: Steganography is the field of science which is concerned with covert communications by hiding secret information within an innocent-looking medium, and sending the medium as a carrier over the communication channel. One of the most used techniques is Least Significant Bit (LSB) Substitution, which hides secret data by substituting the LSB of each binary sequence with the bits of the secret data. In our work we introduced a new algorithm of random information hiding based on indicators. Images were used through our research as the cover mediums for experiments and image quality metrics were used for evaluation.

Keywords: Steganography, Information Hiding, Covert Communication

1. Introduction

Due to the need for transmitting secret data, steganography plays an important role in secure communications. Steganography accomplishes data security by hiding the communication taking place between meant parties. Covert communication is done by hiding the secret data within an innocent-looking medium and then sending the medium over the communication channel, such that it doesn't arouse any suspension. The medium into which data is embedded is called the cover medium, and the resulting object after embedding is called the stego medium. Cover mediums could be of several types, as images, audio, videos, etc. The advantage of steganography is that there is no knowledge of the message existence in the first place [1], which in turn makes the information avoid being attacked. Data embedding could be accomplished using a secret key for more security, so it is hard to extract the embedded data in absence of the secret key. Figure 1 shows steganographic systems architecture.

Data embedding operation is done by altering cover mediums values to make them contain the secret data. However, altering values of some parts of a cover file may destroy it or result in noticeable distortion. Steganographic systems could be evaluated depending on three criteria.

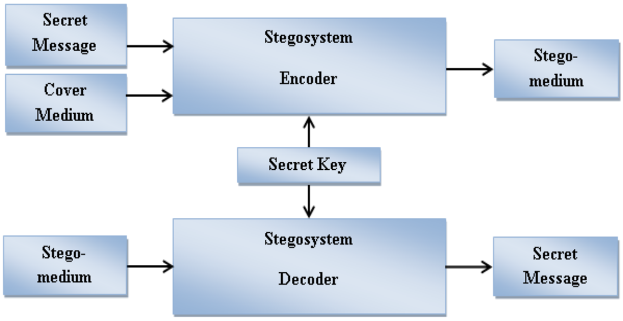


Figure 1. Steganography System Architecture

First, imperceptibility, which is how much the stego mediums have no perceptually noticeable distortion. So, a stego medium should be as identical to the cover medium as possible. Second, robustness, which is the degree of how much the steganographic system can withstand against attacks. Hidden data security is enhanced by enhancing imperceptibility and robustness. Also, we can increase the capacity, which is the maximum amount of secret data that a cover medium can undetectably contain.

2. Least Significant Bit Substitution

One of the earliest and most popular techniques is Least Significant Bit Substitution technique (LSB). LSB term refers to the smallest (right-most) bit of a binary sequence[2]. LSB Substitution technique is defined as

E-mail: tbarhoom@iugaza.edu.ps, wsqr@students.iugaza.edu.ps

the operation of replacing the LSB of a cover binary sequence with a secret bit in order to make the cover binary sequence contain the secret bit. The simplest data hiding algorithm is Hide & Seek algorithm, since it uses LSB substitution to embed secret data into cover sequences sequentially. The process of hiding data into cover sequences using LSB substitution could be accomplished either sequentially or randomly. Many algorithms were set for hiding data in random fashion. Each algorithm has its own way of randomization.

3. Related Work

Many researchers have presented algorithms for hiding data into cover mediums. Next a brief overview is given on some of the most related algorithms to ours.

In research [3] it has been proposed using secret key and the red channel of the cover image for deciding position of embedding inside cover images. Simply, the value of red channel is XORed with the secret key and when the result is 0, the secret bit is embedded into blue channel, and otherwise into green channel.

Researchers [4] used RC4algorithm (Rivest Cipher 4)for embedding randomization. Also they introduced a technique, called bit-inversion, for improving the image quality.

In research of [5] an algorithm has been presented called filtering-based which uses LSB in different manner. Where the algorithm doesn't embed within the LSBs the secret data bits, rather it embeds indications identifying whether a pixel contains hidden data or not.

Reference [6] has proposed performing XOR operation between least two significant bits of each cover byte and the secret data bits, and substituting the result for the least two significant bits. They also used Genetic algorithm (GA) to optimize stego image quality.

In the research of [7], hiding into image process is done first inside odd pixels then even pixels.

In research [8] have introduced two new techniques for hiding data within audio mediums. First, Sample Selection which is used to determine the cover samples inside which secret data is embedded. Second, Bit Selection which is used to select the cover bits within a sample.

Researchers in [9] have introduced a new method that uses certain functions to assign frame of the cover video to be the index frame. The index frame is used for locating the frames within which secret data is embedded.

As shown, many algorithms were set for information hiding, each of which has its points of strength and weakness. In our work a new algorithm is proposed, called Indicators-based LSB, for hiding secret data with high randomness and extra capacity, depending on indicators.

4. Proposed Algorithm

Indicators-based LSB is a steganographic algorithm which uses LSB substitution for data embedding, depending on indicators for randomization. The algorithm illustrated in figure 2 for data hiding process and figure 3 for data retrieving process.

Indicators are two certain bits inside the cover bytes according to which it is decided where and how many secret bits to embed at a time. Indicator bits were predefined to be the fourth and the third bit from the right (bits of indices 3 and 2) of each cover bit. Bit of index 3 is used for identifying the cover byte into which secret bit(s) is embedded. It is called Location Indicator. Bit of index 2 is used for identifying how many secret bits to embed within the cover byte. It is called Amount Indicator.

If the value of the Location Indicator is zero, the embedding is done into some previous cover byte before the current cover byte. If the value is one, secret bit(s) is embedded into exactly the next cover byte, and so forth. Also, through each iteration of the embedding process, the Amount Indicator is checked. If the value of Amount Indicator is zero, then one bit is embedded. If it is one, then two bits would be embedded at once.

Consequently, to find how much data could be embedded into a cover image, it is required to check all of the color channels of every single pixel to identify how many bits they can contain. However, we can calculate the minimum amount of data in bit by supposing that all color channels (Red, Green and Blue) would contain only one bit.

$$DataSize_{min} = Pixels \times 3 \qquad (3)$$

For clarifying how the algorithm works, in Table 1, an example of embedding some data is explained as next:

Secret data bits: 1110110001100110
Secret key bits: 01001101

Table 1: Cover Bytes Brfore and After Embedding

Cover bytes Index	Cover bytes order to contain data	The series of cover bytes before the embedding process	The resulting bytes after the embedding process
0	1	10111000	101110[10)
1	4	10000001	100000[11)
2	7	10100001	101000[11)
3	2	01111000	011110(01]
4	3	01110010	0111001(0)
5	8	10011100	1001110(0)
6	5	11010001	1101000[0]
7	6	01010111	010101[00]

8	9	01010101	0101010(1)
9		10110101	10110101
10		11010011	11010011
11	10	11110000	111100[11]

(b) The bit new value is identical to its original
[b] The bit new value is different from its original
(bb] Only the right bit new value is different from the original
[bb) Only the left bit new value is different from the original
(bb) Both new values are identical to the original
[bb] Both new values are different from the original

As shown in Table 1, the order of the cover bytes that were embedded into is 0, 3, 4, 1, 6, 7, 2, 5, 8, 11 and the amount of embedded bits into each of which respectively is 2, 2, 1, 2, 1, 2, 2, 1, 1, 2. So, can realize that unlike sequential LSB approach, the hiding process is not sequential and on the other hand, some bytes contain only one secret bit and others contain two bits, which in turn increase hidden data security.

Next, Figure 2 shows the flow chart of the embedding process.

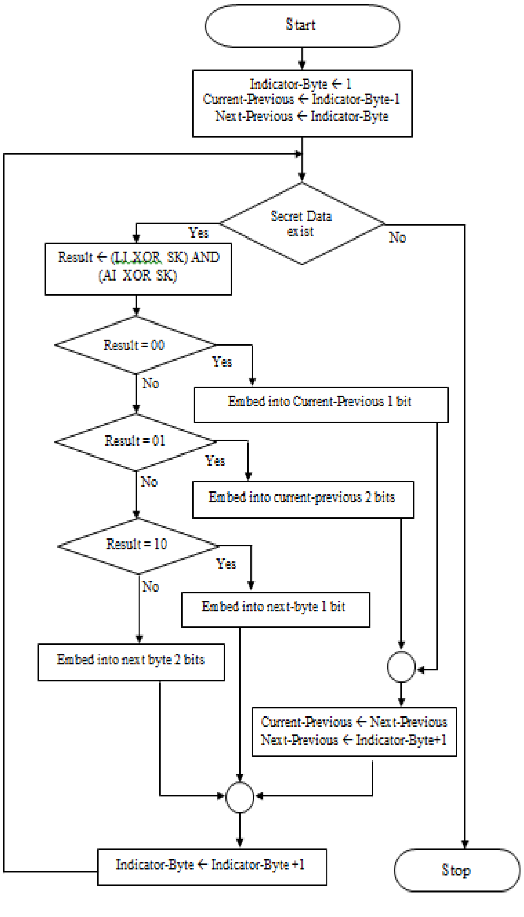

Start
Indicator-Byte ← 1
Current-Previous ← Indicator-Byte-1
Next-Previous ← Indicator-Byte
Secret Data exist
Yes
No
Result ← (LI XOR SK) AND (AI XOR SK)
Result = 00
Yes
No
Embed into Current-Previous 1 bit
Result = 01
Yes
No
Embed into current-previous 2 bits
Result = 10
Yes
No
Embed into next-byte 1 bit
Embed into next byte 2 bits
Current-Previous ← Next-Previous
Next-Previous ← Indicator-Byte+1
Indicator-Byte ← Indicator-Byte +1
Stop

Figure 2. The Embedding Flow Chart, LI= Location Indicator, AI= Amount Indicator, SK= Secret Key

The retrieving process is simply the inverse of the embedding process. Next is Figure 3 showing the flow chart of the retrieving process.

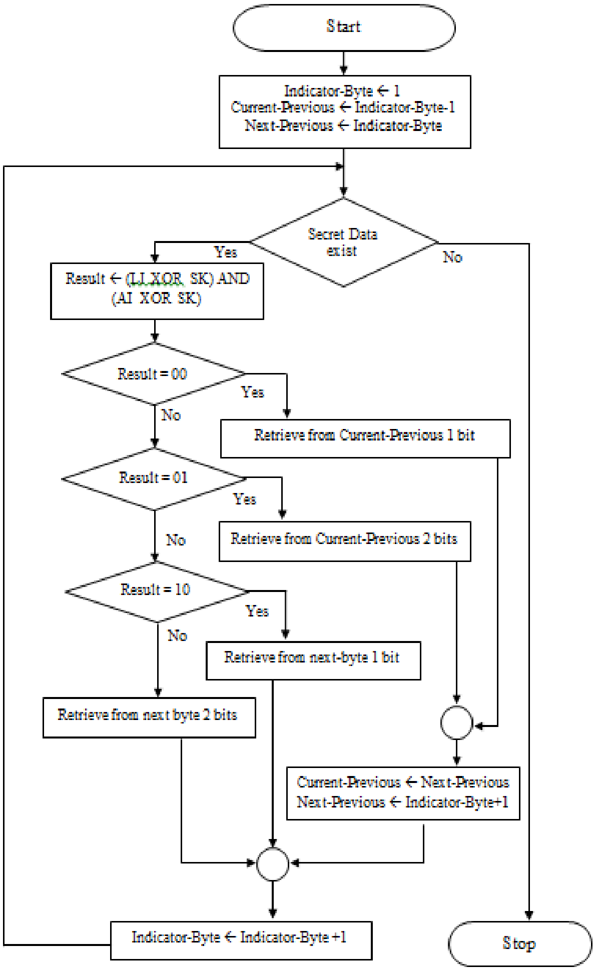

Start
Indicator-Byte ← 1
Current-Previous ← Indicator-Byte-1
Next-Previous ← Indicator-Byte
Secret Data exist
Yes
No
Result ← (LI XOR SK) AND (AI XOR SK)
Result = 00
Yes
No
Retrieve from Current-Previous 1 bit
Result = 01
Yes
No
Retrieve from Current-Previous 2 bits
Result = 10
Yes
No
Retrieve from next-byte 1 bit
Retrieve from next byte 2 bits
Current-Previous ← Next-Previous
Next-Previous ← Indicator-Byte+1
Indicator-Byte ← Indicator-Byte +1
Stop

Figure 3. The Retrieving Flow Chart, LI= Location Indicator, AI= Amount Indicator, SK= Secret Key

5. IMAGE QUALITY METRICS PSNR AND MSE

PSNR and MSE are the most common and widely-used metrics for image quality evaluation [10].

Through experiments these metrics are used for evaluation stego images quality. PSNR measures the similarity between two images, while MSE measures the difference between two images. For MSE, it is better if its value is too small or close to zero. For PSNR, the higher the value, the better the image quality. PSNR values between 20 and 40 can be considered as typical values [11]. PSNR and MSE are defined as follows [10]:

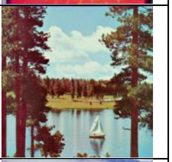
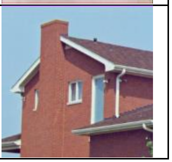
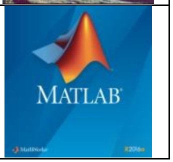
$$\mathrm{MSE} = \left(\frac{1}{\mathrm{MN}}\right) \sum_{i=1}^{M} \sum_{j=1}^{N} \left(X_{ij} - \overline{X}_{ij}\right)^2 \qquad (1)$$

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

6. EXPERIMENTS AND RESULTS

The efficiency of Indicators-based LSB algorithm is shown as embedding into LSB and 2nd LSB. To measure the efficiency of our algorithm, the algorithm has been tested over a data set consisting of some cover images. The cover images were gathered from USC-SIPI-ID which contains the famous images globally used for steganographic algorithms evaluation such as Pepper, and randomly from Internet. All of the images are of PNG type, since spatial domain is used for embedding. Cover images details are shown below in Table 2.

TABLE 2: COVER IMAGES FOR EXPERIMENTS

No	Name	Dimensions (pixel)	Number of Bytes	Image
1	Splash	512 × 512	786,432	
2	Sailboat on lake	512 × 512	786,432	
3	Airplane F-16	512 × 512	786,432	
4	Nature	512 × 512	786,432	
5	Parking	512 × 512	786,432	
6	Peppers	512 × 512	786,432	
7	House 2	512 × 512	786,432	
8	House 1	256 × 256	196,608	
9	Tree	256 × 256	196,608	
10	MATLAB Logo	1030 × 1060	3,275,400	

Data that would be hidden inside cover images is not fixed; rather it is relative to the number of the image pixels. The data is generated to fill certain percentage of the image pixels. Experiments start by filling 10% of image pixels and increase the percentage by 5 each time until 45%. For image 10, called MATLAB Logo, the percentage was increased by 5 each time until 55%. For each resulting stego image, MSE and PSNR were found, and the image itself is subjected to a steganalysis tool, called StegExpose, to see if it is detected or not. Table 3 shows the average of MSE values, the average of PSNR values and the average of the Undetectability of all stego images for each cover image.

TABLE 3: RESULTS OF STEGO IMAGES OBTAINED BY EMBEDDING EXPERIMENTS

NO	Name	MSE Average	PSNR Average	Undetectability Average
1	Splash	0.37	52.86	75.00%
2	Sailboat on lake	0.37	52.88	62.50%
3	Airplane F-16	0.37	52.92	62.50%
4	Nature	0.37	52.94	62.50%
5	Parking	0.39	52.69	62.50%
6	Peppers	037	52.89	50.00%
7	House 2	0.36	53.00	37.50%
8	House 1	0.38	52.79	25.00%
9	Tree	0.38	52.83	12.50%
10	MATLAB Logo	0.48	51.88	90.00%

For MSE, which is the statistical difference between cover and stego images, as shown in Table 2 and Figure 4, the MSE average values range starts from 0.37 to 0.48 among all the images. Therefore, the difference between cover and stego images is too small.

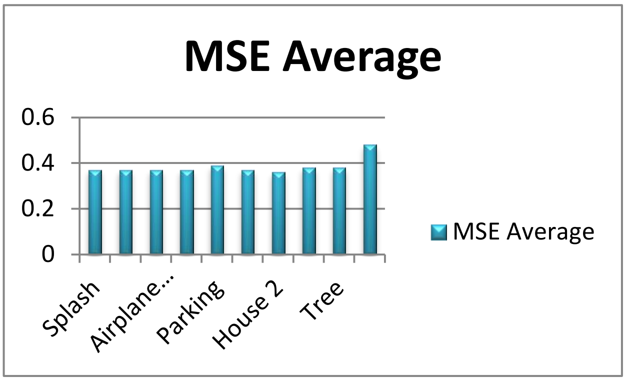


Figure 4. MSE Average for Cover and Stego Images

Additionally, for PSNR, which is the similarity between cover and stego images, the PSNR average values range starts from 52.86 to 51.88 among all the images as shown in Table 3 and Figure 5. Since PSNR values exceed 40, then the algorithm is considered very imperceptible.

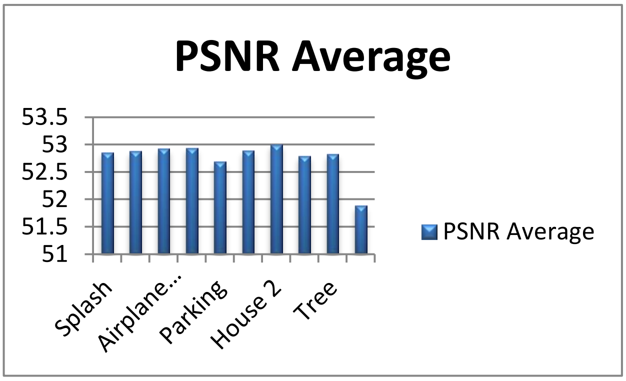


Figure 5. PSNR average for cover and stego images

Subsequently, both MSE and PSNR metrics indicates that stego images are not perceptually detectable, which means the algorithm works with high imperceptibility.

Furthermore, all the stego images were subjected to a steganalysis tool. For images of dimensions of 512×512, most of the images were detected when the embedded data has filled more than 30% of the cover bytes.

However, it cannot be considered as a rule that filling less than 30% of an image makes it undetectable as shown in Table 3 and Figure 6. The detectability depends on the structure of the LSBs values of the cover image itself. Since images have general pattern for their statistical characteristics, as PoVs of LSBs and histograms, then, most of statistical attacks algorithms depend on these characteristics patterns to decide whether an image is suspicious or not. Where steganalysis algorithms check the characteristics of the image, and if a characteristic is out of its general pattern, the image is considered suspicious.

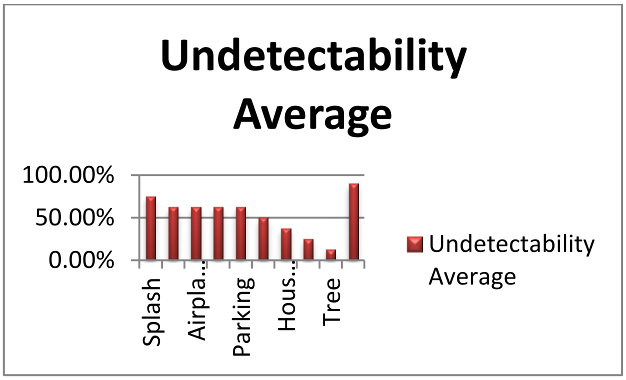


Figure 6. undetectable average for cover and stego images

7. Conclusion

Steganography algorithms are developed to enhance information hiding security and capacity. There are some aspects to consider when an algorithm is developed. These aspects are imperceptibility, capacity and robustness. In our work a new algorithm was presented based on LSB substitution, called Indicators-based LSB. The algorithm aim is to hide secret information with high imperceptibility, robustness and moderate capacity.

The algorithm works with high imperceptibility, since it uses LSB substitution. Robustness is very high because of the high randomness of the embedding process. Capacity is increased by embedding sometimes secret data into 2nd LSB beside the LSB.

8. References

[1] R. Krenn, "Steganography and steganalysis," Retrieved September, vol. 8, p. 2007, 2004.

[2] P. Bateman and H. G. Schaathun, "Image steganography and steganalysis," Department Of Computing, Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, Surrey, United Kingdom, 4th August, 2008.

[3] S. M. Karim, M. S. Rahman, and M. I. Hossain, "A new approach for LSB based image steganography using secret key," in Computer and Information Technology (ICCIT), 2011 14th International Conference on, 2011, pp. 286-291.

[4] N. Akhtar, P. Johri, and S. Khan, "Enhancing the security and quality of LSB based image steganography," in Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference on, 2013, pp. 385-390.

[5] M. R. Islam, A. Siddiqa, M. P. Uddin, A. K. Mandal, and M. D. Hossain, "An efficient filtering based approach improving LSB image steganography using status bit along with AES cryptography," in Informatics, Electronics & Vision (ICIEV), 2014 International Conference on, 2014, pp. 1-6.

[6] S. Laha and R. Roy, "An improved image steganography scheme with high visual image quality," in Computing, Communication and Security (ICCCS), 2015 International Conference on, 2015, pp. 1-6.

[7] S. Singh and J. Kaur, "Odd-Even Message Bit Sequence Based Image Steganography," International Journal of Computer Science and Information Technologies, vol. 6, 2015.

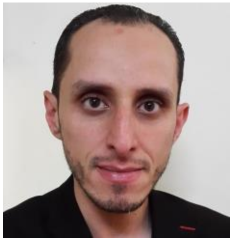
[8] M. Asad, J. Gilani, and A. Khalid, "An enhanced least significant bit modification technique for audio steganography," in Computer Networks and Information Technology (ICCNIT), 2011 International Conference on, 2011, pp. 143-147.

[9] R. Balaji and G. Naveen, "Secure data transmission using video Steganography," in Electro/Information Technology (EIT), 2011 IEEE International Conference on, 2011, pp. 1-5.

[10] A. Al-Mohammad, "Steganography-based secret and reliable communications: Improving steganographic capacity and imperceptibility," Brunel University, School of Information Systems, Computing and Mathematics Theses, 2010.

[11] C. Eric, "Hiding in plain sight, Stegnography and the art of Covert Communication," Wiley, Indianapolis, Indiana, ISBN, vol. 10, p. 0471444499, 2003.

Tawfiq S. Barhoom Head of Computer Science and Software Development Departments Faculty of IT, Islamic University-Gaza, he got B.Sc. Computer Science from Omdurman Ahlia University-Sudan,(1991-1995) and Master degree, and he has – M.Sc. Computer science, Department of computer science and engineering from Shang hai Jiao Tong University (SJTU)– ShangHai – China, (1996- 1999) and his has – PhD in Applied computer Technologies, Department of computer science and engineering from ShangHai Jiao Tong University (SJTU) – Shanghai – China, (2001- 2004).



Wesam M. Saqer received the B.A. degree in computer science from Al-Aqsa University, Gaza, Palestine in 2011. He obtained Master Degree in Information Technology from Islamic University, Gaza, Palestine in 2017. His main research interests are Steganography and Steganalysis.