A Novel Technique for Data Steganography

Amjad Y. Hindi
Department of Communication Technology Engineering, Faculty of Engineering Technology, Al-Balqa Applied University, Amman-Jordan
amjadhindi@bau.edu.jo

Majed O. Dwairi
Department of Communication Technology Engineering, Faculty of Engineering Technology, Al-Balqa Applied University, Amman-Jordan
majeddw@bau.edu.jo

Ziad A. AlQadi
Department of Computer &Network Engineering, Faculty of Engineering Technology, Al-Balqa Applied University, Amman, Jordan
natalia_maw@yahoo.com

***Abstract*—In this paper, a novel stego-method will be introduced, which can be used to hide any secret message in any holding color image. The proposed method will be implemented and tested and the calculated parameters will be compared with the LSB method parameters. It will be shown that the proposed method provides a high-security level by using two keys to extract the secret message from the holding image, making it very difficult to hack.**

Keywords-steganography; hiding time; extracting time; MSE; PSNR

I. INTRODUCTION

Steganography is a process of hiding data in covering data, such as a hidden text message in a color image [1, 2]. Steganography is an important process and many applications utilize it. Hiding a text message in color image can be performed as shown in Figure 1 by selecting a stego-system encoder to hide the message and a stego-system decoder to extract the message from the holding color image [3].

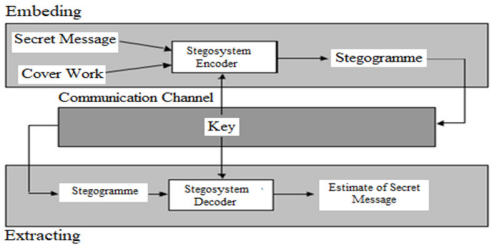

Fig. 1. Steganography process

One of the most popular methods of secret message hiding is the least significant bit (LSB), and many methods are based on it [4-6]. LSB reserves 8 bytes from the covering image to hide a character from the secret message, the first bit of the binary version of the character is stored in the least significant bit of the first selected byte of the image, the second bit in the least significant bit of the second byte of the image and so on [4]. Table I shows how to hide the letter "A" (ASCII A is equal to 65 in decimal and 01000001 in binary) in a sequence of 8 bytes in a color image. LSB is easy to implement, the changes to the image are not essential and cannot be observed with the naked eye, and it is easy to discover the text message and therefore it is not considered safe.

TABLE I. HIDING LETTER A

Byte sequence	Color value before hiding	Binary value before hiding	Binary value after hiding	Color value after hiding	Remarks
S1	240	11110000	11110000	241	Changed
S2	177	10110001	10110000	176	Changed
S3	180	10110100	10110100	180	No change
S4	150	10010110	10010110	150	No change
S5	190	10111110	10111110	190	No change
S6	200	11001000	11001000	200	No change
S7	110	01101110	01101111	111	Changed
S8	135	10000111	10000110	134	Changed

So, in order to take the advantages and to discard the disadvantages of LSB data hiding method we modified it to improve the security level of data hiding [7]. LSB method provides a small mean square error (MSE), and a high peak signal to noise ratio (PSNR) between the original and the holding images, these parameters are very important for analysis purposes [4] and they are calculated by:

$$MSE = \frac{1}{mn}\sum_{0}^{m-1}\sum_{0}^{n-1} \| f(i,j) - g(i,j) \|^2 \quad (1)$$

$$PSNR = 20\log(\frac{MAX_f}{\sqrt{MSE}}) \quad (2)$$

where f represents the matrix data of the original image, g represents the matrix data of the holding image in question, m represents the number of pixel rows of the images and i represents the index of the current row, n represents the number of pixel columns of the image, j represents the index of the current column, and MAX_f is the maximum signal value that exists in our original "known to be good" image.

II. RELATED WORK

LSB steganography is widely used to hide secret messages into color images due to its simplicity [15]. In [4], a method based on LSB was proposed, which creates a key of random positions to hide the message. This method is very secure but the MSE increases when the message length increases. In [10], a technique was introduced on hiding a secret image into a cover image, where both images should have the same size. The technique first compresses the secret message using the Set Partitioning in Hierarchical Trees (SPIHT) algorithm, then the output of this compression was hided into the covering image

Corresponding author: Majed O. Dwairi

using the default LSB technique. The compression is made by wavelet transform and then by SPIHT coding. Image quality is retained with high PSNR values. In [11], the authors hided a secret message file into a covering image, the image should be colored and transformed into 3 matrices (R, G, and B). The message converts to binary, depending on the secret message bit using OR operation or AND operation, sequentially (RGB, BGR, RGB, BGR...). The results showed better performance in terms of quality of the obtained stego-image. Authors in [9] used LSB and DCT to perform steganography. The comparison gave a good result according to the PSNR values when compared with previous works and the security was increased by using DCT. In [8], the message was embedded by hiding each byte of the message in three pixels based on randomization in the cover image using Pseudo Random Number Generator (PRNG) of each pixel's value. This method achieved a very high maximum hiding capacity and higher visual quality as indicated by PSNR. In [16], the authors tried to overcome the disadvantage of the LSB method by appending the encrypted data in an image in place of plain textual data. To encrypt the data RSA and Diffie Hellman algorithms were used. To check the efficacy of their proposed method, they calculated the number of instructions executed at sender and receiver site since the number of instructions executed is a measure of the time complexity of the process. The result showed that the use of encryption in stego-analysis does not affect the time complexity if Diffie Hellman algorithm is used instead of the RSA algorithm. In [15], a method that hided a secret text message was proposed based on searching for identical bits between the secret message and image pixels values. The proposed method was compared with the LSB benchmarking method for hiding a secret message which hides the secret message directly in the least two significant bits of the image pixels. The proposed method was more efficient, simple, appropriate and accurate than the LSB method, the change in the image resolution was quite low and it made the secret message more secure. In [17], the authors used the Pixel-Value Differencing (PVD) method as an image steganography mechanism. They eliminated the overflow problem of pixel values in the stego image exceeding the range 0 ... 255. Moreover, for providing more security, they used different number of bits in different pixel components. It was very difficult to trace how many bits are embedded in a pixel of the stego image. The obtained results provided better visual quality of the stego-image compared to the PVD method.

III. THE PROPOSED METHOD

The proposed method for data hiding can be implemented by applying the following steps:

- Select the original color image, and find the image size (*n1*:number of rows, *n2*: number of columns, and *n3*: number of colors).
- Select the message to be hidden in the image and find the message length (*n4*).
- Define an 8 digit number to be used as private key (*key1*).
- Divide *key1* into 2 equal parts (*part1* and *part2*, each of them is a 4 digit number.
- Reshape the original image from 2D matrix to 2D matrix with size *n1*n3*, *n2*.
- Calculate the row and column indexes (where to start the hiding message) using a defined hash function, in our case we used the following functions:

$$\text{Row index} = \text{floor}(\text{rand}(1)*(n1*n3-n4));$$

$$\text{Column index} = \text{floor}(\text{rand}(1)*(n2-n4));$$

- Apply LSB method to hide a message using the indexes.
- Reshape the holding image back to a 3D matrix.
- Generate the second key (*key2*) by using another hash function. We used the XOR function with the indexes and the two parts.
- Save *n4*, *key1* and *key2* to be used to extract the message from the image.

To extract the data, the proposed method requires the following steps to be implemented:

- Select the holding image.
- Reshape the image matrix from 3D to 2D matrix.
- Get *n4*, *key1* and *key2*.
- XOR the first part of *key1* with the first part of *key2* to get the row index.
- XOR the second part of *key1* with the second part of *key2* to get the column index.
- Use the indexes to retrieve *n4* characters from the image.

Figure 2 shows the block diagram of the proposed stego-system, while Figure 3 shows a simple example of how to perform some calculations using the proposed method.

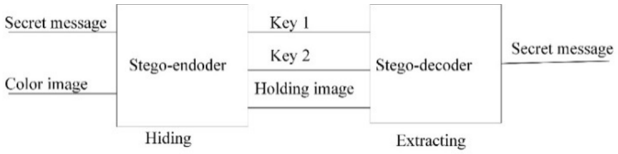

Fig. 2. The proposed stego-system

Key1	5	3	2	7	3	7	1	2
Index x	999							
Index y	95							
Key 2								
	5	9	2	8	3	8	0	7
	First half of Key1 xor index x				**Second half of Key1 XOR index y**			
Key1 XOR Key2	999				95			

Fig. 3. A calculation example of the proposed method (image size=384×512×3), *n4*=100

Using this method will increase security level, because we have to know *key1* and *key2* and the way of their calculations.

This method also decrease MSE, which will be very small whatever the message length was. Figures 4 and 5 show the original image and the holding image after hiding 100 characters, and here we can notice that there are no visible differences between the two images.

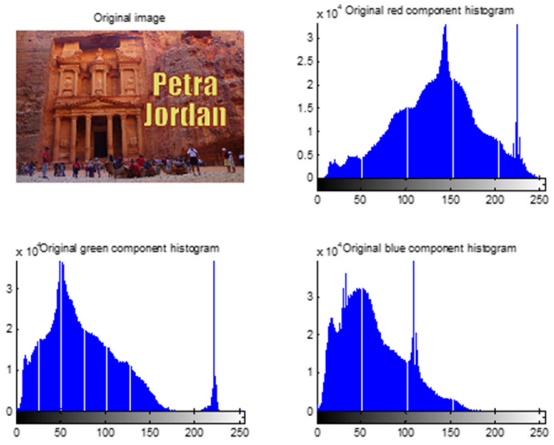

Fig. 4. Original image

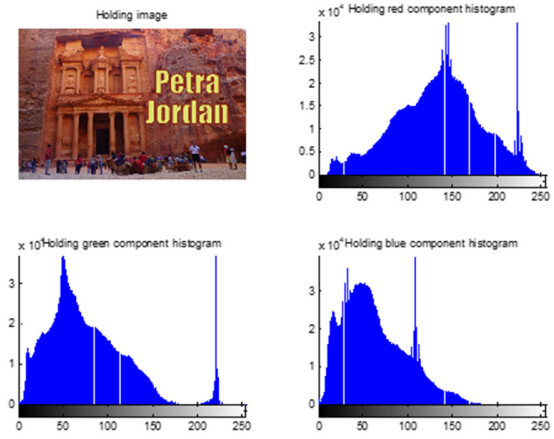

Fig. 5. Holding image

IV. IMPLEMENTATION, RESULTS, AND DISCUSSION

The proposed method was implemented several times using various images and different size messages. The experimental results are exhibited and analyzed below.

A. Experiment 1: Hiding a Fixed Length Message

The following message with length of 100 characters was hidden in different types and size images. Steganography is the process of hiding of a secret message within an ordinary message and extracting at its destination [20]. From the results shown in Table II we can conclude the following which can be considered as the advantages of the proposed stego-method:

- Hiding and extracting times are significantly small.
- Hiding time slowly increases due to the increase in image size. In the experiment it fell in the range 0.093000-2.506000.
- Extracting time slowly increases due to the increase in image size and in the experiment it fell in the range 0.442000-0.469000.
- The proposed method provides a significantly small MSE.
- The proposed method provides a significantly high PSNR.

TABLE II. EXPERIMENT 1 RESULTS

Image #	Type	Size (each color)	Hiding time (s)	Extracting time (s)	MSE	PSNR
1	bmp	268x337	0.140000	0.459000	0.008400	158.6776
2	bmp	225x225	0.113000	0.462000	0.016300	152.0137
3	bmp	128x128	0.093000	0.442000	0.049000	140.8332
4	bmp	469x800	0.204000	0.412000	0.002100	172.2939
5	bmp	300x600	0.169000	0.418000	0.004400	165.0364
6	tif	941x1203	0.515000	0.420000	0.000718	183.2172
7	tif	768x1024	0.845000	0.403000	0.001000	179.7110
8	tif	516x600	0.242000	0.420000	0.002600	170.1870
9	tif	320x450	0.155000	0.401000	0.005600	162.6265
10	tif	201x251	0.118000	0.436000	0.015600	152.4010
11	png	750x975	0.964000	0.467000	0.001100	178.9544
12	jpg	184x274	0.124000	0.487000	0.015700	152.3940
13	jpg	1655x2498	2.506000	0.469000	0.000196	196.1701
14	jpg	320x450	0.165000	0.449000	0.005500	162.7715
15	jpg	201x251	0.118000	0.440000	0.015400	152.5456
16	jpg	214x236	0.113000	0.511000	0.013400	153.9270
17	jpg	180x279	0.114000	0.520000	0.016400	151.9212
18	jpg	217x232	0.110000	0.481000	0.015900	152.2288
19	jpg	180x280	0.110000	0.508000	0.015500	152.4927
20	jpg	300x300	0.131000	0.476000	0.002000	172.9714

B. Experiment 2: Hiding Various Messages in a Colored Image

A tiff color image with size 516×600×3 was taken, and several messages with different lengths were hidden in the image and extracted from the holding image using the proposed method. The results of this experiment are shown in Table III.

TABLE III. EXPERIMENT 2 RESULTS

Message length (chars)	Hiding time (s)	Extracting time (s)	MSE	PSNR
10	0.210000	0.379000	0.00028316	192.5202
20	0.226000	0.397000	0.00055771	185.7420
30	0.226300	0.397100	0.00095823	180.3295
40	0.228000	0.402000	0.0011	178.6954
50	0.229000	0.403000	0.0014	176.5868
200	0.694000	0.463000	0.0060	162.0298

C. Experiment 3: LSB Implementation

The LSB method was implemented to compare the calculated LSB parameters with the proposed method's parameters. Tables IV and V show the results. We can see that the calculated values of the proposed stego-method parameters are very close to the LSB parameters values.

V. CONCLUSION

The proposed stego-method was presented, implemented and tested. The obtained results were acceptable when compared with the results of the LSB method. LSB method suffers from low security level. The proposed stego-method

provides a high security level, because it needs two keys to extract the secret message from the holding image, each of these keys consisting of eight decimal digits making the process of penetration very difficult.

TABLE IV. TIME COMPARISON

Image #	Proposed		LSB	
	Hiding time (s)	Extracting time (s)	Hiding time (s)	Extracting time (s)
1	0.140000	0.459000	0.129000	0.021000
2	0.113000	0.462000	0.109000	0.433000
3	0.093000	0.442000	0.090000	0.452000
4	0.204000	0.412000	0.229000	0.415000
5	0.169000	0.418000	0.155000	0.398000
6	0.515000	0.420000	0.573000	0.488000
7	0.845000	0.403000	0.865000	0.409000
8	0.242000	0.420000	0.212000	0.403000
9	0.155000	0.401000	0.149000	0.424000
10	0.118000	0.436000	0.112000	0.428000

TABLE V. ERROR COMPARISON

Image #	Proposed		LSB	
	MSE	PSNR	MSE	PSNR
1	0.008400	158.6776	0.007600	159.6321
2	0.016300	152.0137	0.014800	152.9688
3	0.049000	140.8332	0.049300	140.7710
4	0.002100	172.2939	0.001500	176.0419
5	0.004400	165.0364	0.004100	165.8746
6	0.000718	183.2172	0.000726	183.1030
7	0.001000	179.7110	0.000996	179.9380
8	0.002600	170.1870	0.002600	170.3018
9	0.005600	162.6265	0.005500	162.8848
10	0.015600	152.4010	0.016500	151.8625

REFERENCES

[1] E. Zukerman, "Review: OpenPuff steganography tool hides confidential data in plain sigh", PC World, available at: www.pcworld.com/article/2026357/review-openpuff-steganography-tool-hides-confidential-data-in-plain-sight.html

[2] Encryption and steganography, available at: http://www.cis.upenn.edu/~cis110/13fa/hw/hw04/steganography.html#steganography

[3] S. Channalli, A. Jadhav, "Steganography an art of hiding data'', International Journal on Computer Science and Engineering, Vol. 1, No. 3, pp. 137-141, 2009

[4] A. A. Z. Alqadi, M. K. A. Zalata, G. M. Qaryouti, "Comparative analysis of color image steganography", International Journal of Computer Science and Mobile Computing, Vol. 5, No. 11, pp. 37-43, 2016

[5] R. Z. Wang, C. F. Lin, J. C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm", Pattern Recognition, Vol. 34, No. 3, pp. 671-683, 2000

[6] C. K. Chan, L. M. Chen, "Hiding data in images by simple LSB substitution", Pattern Recognition, Vol. 37, No. 3, pp. 469-474, 2004

[7] C. H. Yang, "Inverted pattern approach to improve image quality of information hiding by LSB substitution", Pattern Recognition, Vol. 41, No. 8, pp. 2674-2683, 2008

[8] M. M. Emam, A. A. Aly, F. A. Omara, "An improved image steganography method based on LSB technique with random pixel selection", International Journal of Advanced Computer Science & Applications, Vol. 7, No. 3, pp. 361-366, 2016

[9] G. Kaur, A. Kochhar, "A steganography implementation based on LSB & DCT", International Journal for Science and Emerging Technologies with Latest Trends, Vol. 4, No. 1, pp. 35-41, 2012

[10] M. J. Thenmozhi, T. Menakadevi, "A new secure image steganography using Lsb and Spiht based compression method", International Journal of Engineering Research & Science, Vol. 2, No. 3, pp. 80-85, 2016

[11] S. Shabnam, K. Hemachandran, "LSB based steganography using bit masking method on RGB planes", International Journal of Computer Science and Information Technologies, Vol. 7, No. 3, pp. 1169-1173, 2016

[12] B. Datta, U. Mukherjee, S. K. Bandyopadhyay, "LSB layer independent robust steganography using binary addition", Procedia Computer Science, Vol. 85, pp. 425-432, 2016

[13] A. S. Pandit, S. R. Khope, "Review on image steganography", International Journal of Engineering Science, Vol. 6, No. 5, pp. 6115-6117, 2016

[14] D. Artz, "Digital steganography: Hiding data within data", IEEE Internet Computing, Vol. 5, No. 3, pp. 75-80, 2001

[15] A. M. A. Shatnawi, "A new method in image steganography with improved image quality", Applied Mathematical Sciences, Vol. 6, No. 79, pp. 3907-3915, 2012

[16] S. Gupta, A. Goyal, B. Bhushan, "Information hiding using least significant bit steganography and cryptography", International Journal of Modern Education and Computer Science, Vol. 6, pp. 27-34, 2012

[17] J. K. Mandal, D. Das, "Color image steganography based on pixel value differencing in spatial domain", International Journal of Information Sciences and Techniques, Vol. 2, No. 4, pp. 83-93, 2012

[18] M. O. Al-Dwairi, A. Hendi, Z. AlQadi, "An efficient and highly secure technique to encrypt-decrypt color images", Engineering, Technology & Applied Science Research, Vol. 9, No. 3, pp. 4165-4168, 2019

[19] A. Y. Hendi, M. O. Dwairi, Z. A. Al-Qadi, M. S. Soliman, "A novel simple and highly secure method for data encryption-decryption", International Journal of Communication Networks and Information Security, Vol. 11, No. 1, pp. 232-238, 2012

[20] G. Sonal, H. Mer, "A survey: Image Steganography using different method", International Journal of Novel Research and Development, Vol. 2, No. 4, pp. 48-51, 2017