International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 4, August 2024, pp. 4055~4067
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i4.pp4055-4067

Message steganography using separate locations and blocks

Rashad J. Rasras[1], Mutaz Rasmi Abu Sara[2], Ziad Alqadi[1]
[1]Department of Electrical Engineering, Al-Balqa' Applied University, Amman, Jordan
[2]IT Department, Faculty of Engineering and Information Technology, Palestine Ahliya University, Bethlehem, Palestine

Article Info

Article history:

Received Nov 19, 2023
Revised Mar 26, 2024
Accepted Apr 16, 2024

Keywords:

Chaotic key
Covering block
Image block
Message binary matrix
Private key
Steganography
Stego block

ABSTRACT

A novel method of message steganography is introduced to solve the disadvantages of traditional least significant bit (LSB) based methods by dividing the covering-stego image into a secret number of blocks. A chaotic logistic map model was performed using the chaotic parameters and the number of image blocks for generating a chaotic key. This key was then sorted, and the locations of blocks 1 to 8 were used to select the required blocks to be used as covering-stego blocks. The introduced method simplifies the process of message bits hiding and extracting by adopting a batch method of bits hiding and extracting. A comparative analysis was conducted between the outcomes of proposed method and those of prevalent approaches to outline the enhancements in both speed and quality of message steganography.

Corresponding Author:

Rashad J. Rasras
Department of Electrical Engineering, Al-Balqa' Applied University
Amman 11134, P.O. Box 15008, Jordan
Email: rashad.rasras@bau.edu.jo

1. INTRODUCTION

The art of hiding a text message in a colored digital image ensuring that the image remains visually unchanged, and the hidden data is imperceptible to the naked eye, is known as message steganography, as shown in Figure 1 [1]–[5]. The process involves both hiding and extracting functions. The hiding function involves manipulating the covering image, the secret message, and optionally the private key (PK), to generate a stego image, which conceals the secret message, as illustrated in Figure 2. The extracting function manipulates the stego image and the PK to extract the secret message, as depicted in Figure 3 [6]–[12]. In [13], using Arabic language as the cover text, the author presented a new steganography algorithm. Following the pre-processing of the cover text, the algorithm adds the necessary diacritical marks (like Hamzah Al-Wasl) to the recovered words according to their third letter type (lunar or solar), so concealing parts of hidden messages within the Arabic letters. The secret message's length is calculated in the suggested algorithm to guarantee that the intended receiver can accurately extract it. In [14], a support vector machine (SVM) classifier-based agent system was suggested for concealing a secret message under a particular cover image. To ensure accurate outcomes, the common dataset for steganography typically includes an 80% training and 20% testing split. In research [15], images are used to store patients' private information and messages created by the physician. Sequential exchange of message bits and image pixels. In [16], a secret text is encrypted by using the blowfish algorithm, and then hidden within an image using the least significant bit (LSB) technique. The LSB method was developed to conceal crucial information by modifying the least significant bit, often the first or the first two bits, depending on the cover data. A novel approach that combines encryption and information concealment was introduced [17] to enhance the security of data transmission over networks. This method aims to achieve robust encryption, followed by concealing the secret message within the encrypted image data. The concealment process utilizes a secret key, represented by a cosine curve. The resulting stego-encryption image is then prepared for storage in an internet of things (IoT) database, facilitating seamless data flow within IoT networks. In [18], a video steganography scheme based on object motion and dual-clutch transmission (DCT) psychovisual techniques was proposed for concealing messages. The technique involves embedding a secret message along the object motion within the video frames. Motion analysis is employed to identify suitable embedding regions within the frames.

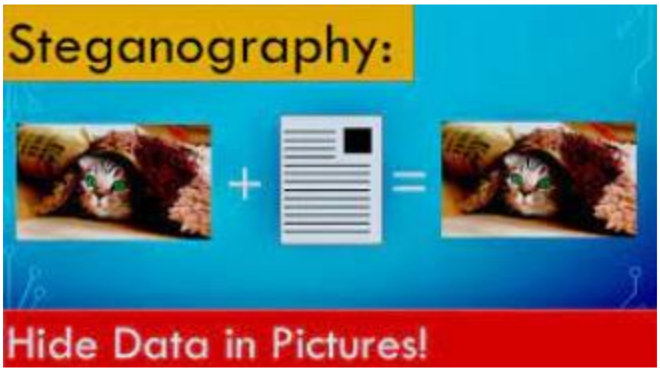

Figure 1. Process of message steganography

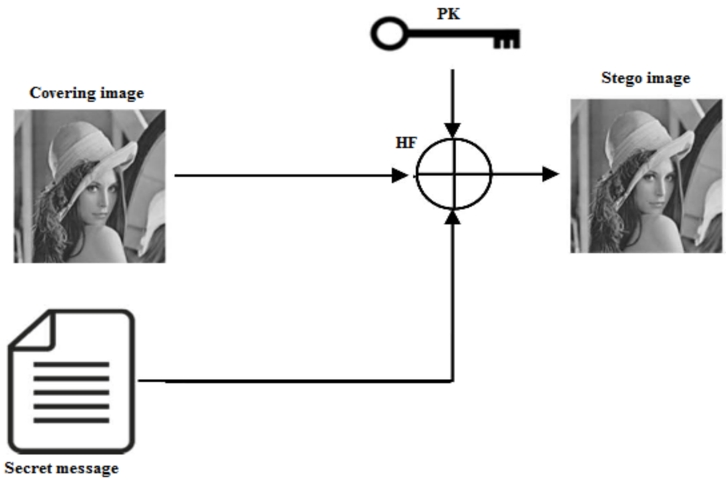

Figure 2. Hiding function

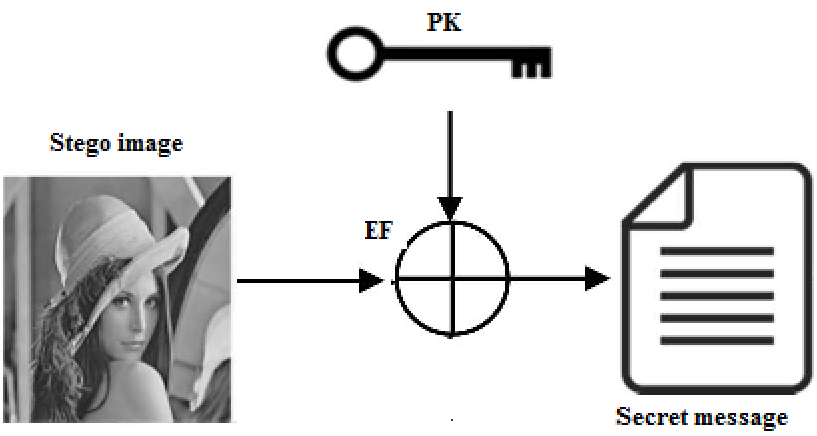

Figure 3. Extracting function

Digital color images are the most convenient medium used to conceal secret messages for the following reasons:

- Digital color image (DCI) has a huge size, providing ample capacity for hiding information. Thus DCIs can be used to conceal both short and long messages [19]–[21].
- Processing DCI is straightforward. Each DCI is represented by a 3D matrix, with one 2D matrix for each color component (red, green, and blue), as illustrated in Figure 4. Consequently, digital image processing essentially involves simplified processing of a digital matrix [22]–[26].
- It is easy to utilize any part or block of the DCI for a desired application.
- Reshaping the image matrix from a 3D matrix to a single row matrix, and vice versa, is straightforward.
- The pixel values correspond to the ASCII character values of the message, and they share the same range of decimal integers (from 0 to 255), as illustrated in Figure 5.
- DCIs are readily available and easily accessible, with numerous resources and equipment readily available.

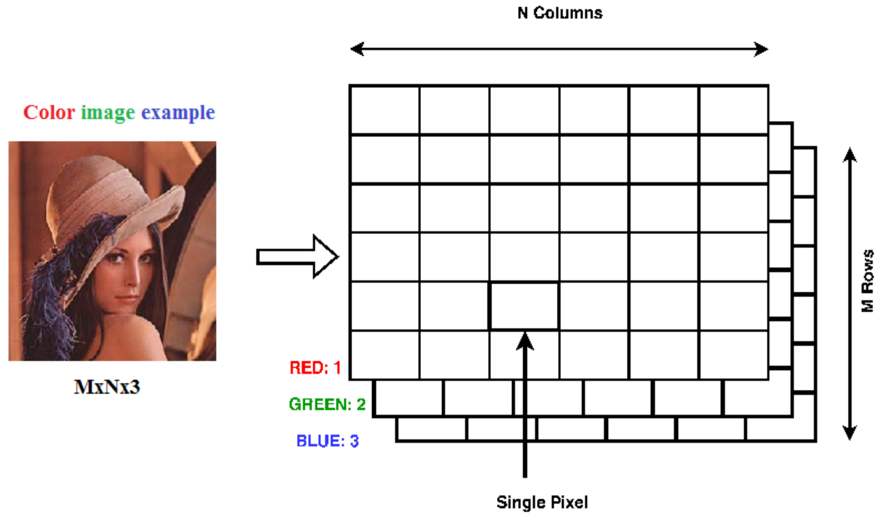

Figure 4. DCI 3D matrix

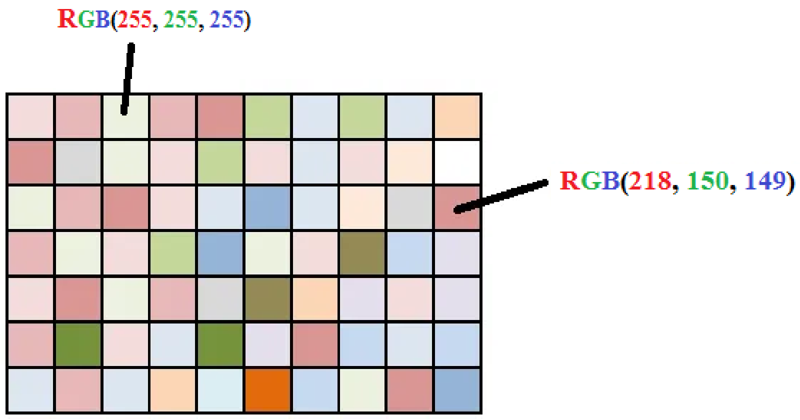

Figure 5. RGB DCI pixels

The message intended to be concealed within a covering image is first converted from decimal ASCII to binary format to create the message binary matrix (MBM) as illustrated in Figure 6. Subsequently, this matrix of binary digits is inserted into the covering DCI. The bytes of the covering-stego image are also converted to binary format, and the least significant bits of these covering bytes are both employed for both hiding and extracting message bits Figure 7.

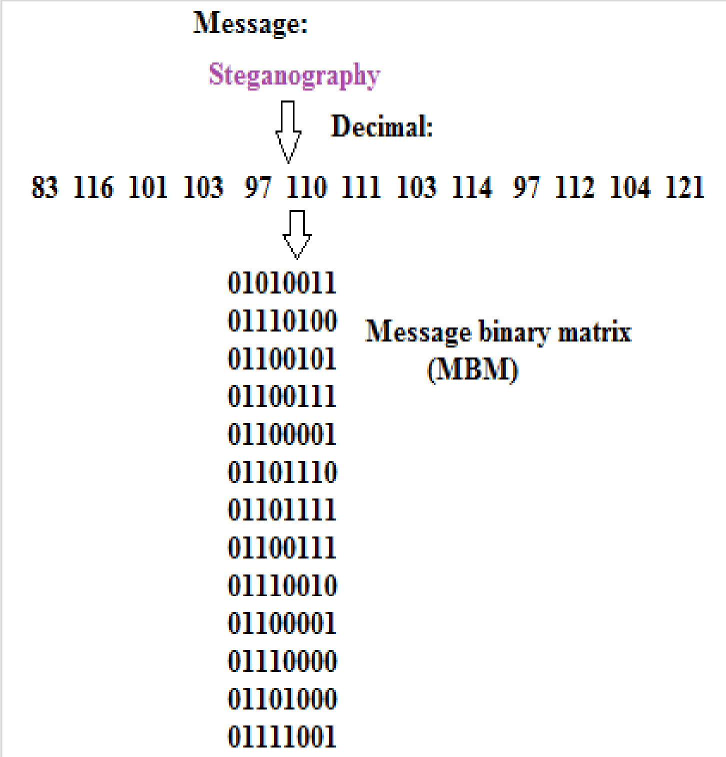

Figure 6. Creating MBM

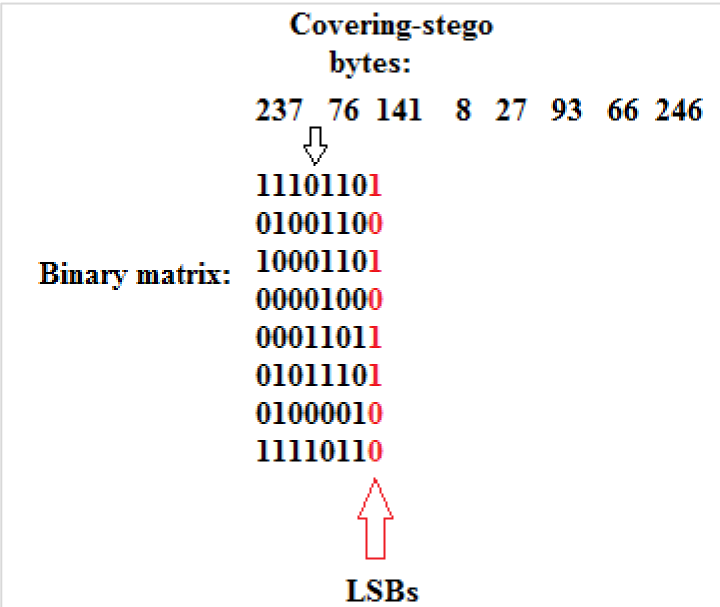

Figure 7. Used LSBs of the covering-stego bytes

The classical least significant bit (CLSB) method [1]-[3] of data steganography is one of the most popular techniques employed to conceal secret data within a covering medium. Many methods are derived from this approach. The CLSB method utilizes the LSBs of the covering-stego bytes to hide and extract the binary message bits as shown in Figure 8. Each character requires 8 covering-stego bytes, with the characters being hidden and extracted byte by byte in successive locations within the covering image. Figure 9 illustrates an example of hiding a single message character.

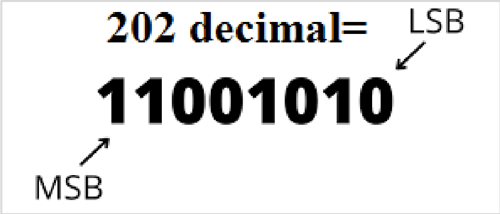

Figure 8. Used LSB in the CLSB method

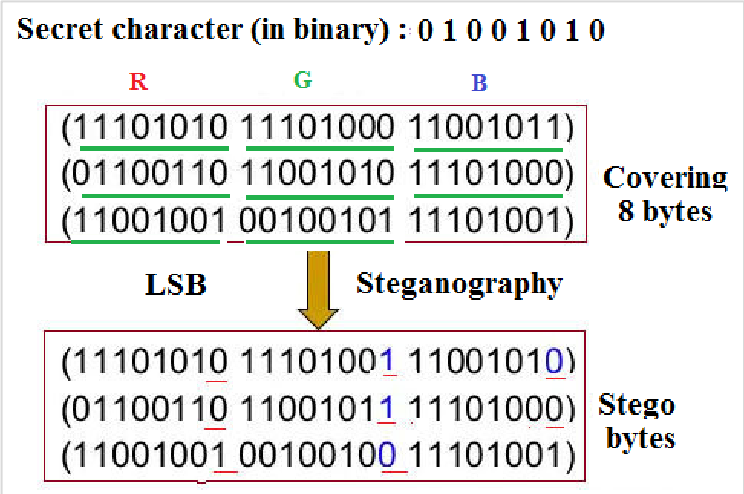

Figure 9. Process of hiding one message character

CLSB-based methods possess several characteristics, some of which are regarded as disadvantages, necessitating resolution by the proposed method:

- Lack of security: The CLSB method does not employ a private key (PK), leaving the hidden message unprotected. Consequently, any individual with programming skills can potentially hack into the message.
- The hiding and extraction processes commence from the initial position within the covering-stego image.
- Message characters are to be hidden and extracted character by character, necessitating the implementation of a complex of logical operation, which makes the hiding and extracting functions complicated.
- Message characters are hidden in successive locations of the covering image.
- The aims of the research are to introduce a new method of data steganography that will address the shortcomings of the CLSB method by providing the following features:
- Securing and protecting the hidden message by employing a sophisticated private key (PK),
- Concealing and extracting the message within secret blocks, selected by using the values of the PK. This key will provide a good key space capable of resisting hacking attacks.
- Using a batch method for hiding and extracting message bits, which simplifies the hiding and extracting functions and enhances their speed?
- Speed up the process of message steganography by increasing the throughput of data steganography.
- Maintaining high-quality stego images, ensuring that the method meets the quality standards outlined in Table 1 for effective steganography.

Table 1. Stego system quality requirements [1]-[5]

Quality parameter	Measured between source CDI and stego CDI	Measured between source message and the extracted one
Mean square error (MSE)	Low	Zero
Peak signal to noise ratio (PSNR)	High	Infinity
Correlation coefficient (CC)	High	One
Number of samples change rate (NSCR)	Low	Zero
Remarks	High quality	Excellent quality

2. METHOD

The suggested technique divides the covering-stego image into equal blocks using a PK whose structure is shown in Table 2 and includes the values for three double data type parameters. Eight blocks from the obtained set of image blocks were selected to hide the MBM as shown in Figure 10. Each column of the MBM was hidden in the LSBs of the selected block.

Covering-stego blocks were selected based on values obtained from the secret key, chaotic parameters R1 and X1, and the number of image blocks. A chaotic logistic map model (KLMM) is used to produce a 1D chaotic key using these parameters. This key is then sorted, and the indexes of the blocks from 1 to 8 are determined. These indexes formed the contents of the secret key. The secret key used in this method allows to separate the message into different and noncontiguous image block, fortifying the concealed message against potential hacking attempts.

Table 2. PK structure

Fractional division parameter (P)	
CLMM parameters	
R1	X1
Example	
P=0.018	
R1=3.77	X1=0.14

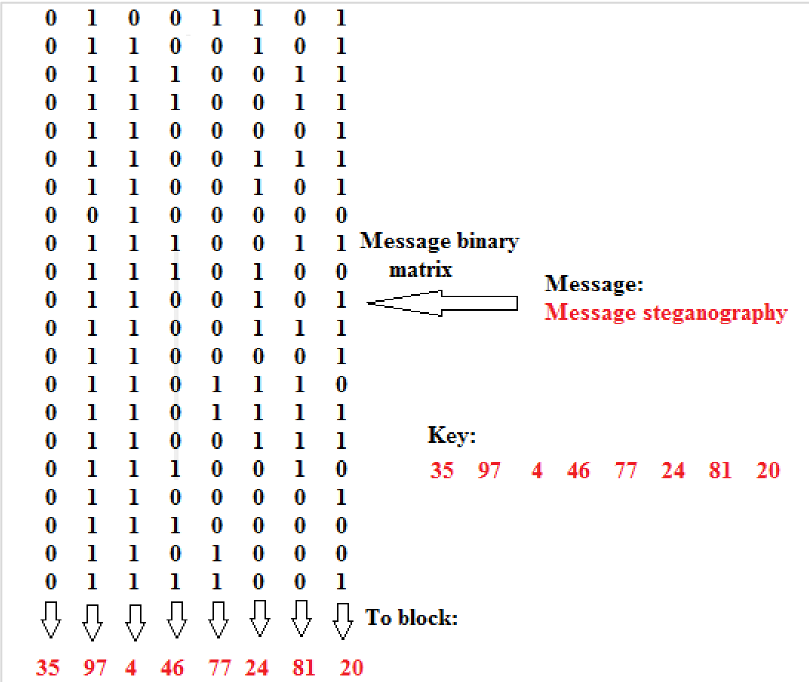

Figure 10. Using secret key to separate the hidden message

The suggested method's hiding function was implemented by performing the following steps:

Step 1:

Inputs preparation step consists of the following operations sequence:

- Read the covering image,
- Calculate the image size,
- Reshape the image into one row matrix,
- Get the secret message,
- Convert the message to decimal,
- Get the message length,
- Select the PK (get the values of P, R1 and X1),
- Calculate the image block size and find the number of image blocks.

The operations in this step were performed by executing the following sequence of MATLAB codes:

```
%image preparation
 a1=imread('C:\Users\win 7\Desktop\Lena.jpg^');
 [n1 n2 n3]=size(a1);L1=n1*n2*n3
 a=reshape(a1,1,L1);
%Message preparation
 m1='Message steganography^';
 m2=unit8(m1);
 L=length(m2);
%get the PK and do calculation:
 P=0.01;
 r1=3.77;x1=0.15;
 BS=L1*P;
 NB=fix(L1/BS)
```

Step 2:
Secret key generation step consists of operations sequence:
- Run the CLMM to get the chaotic key,
- Sort the chaotic key to get an indices key,
- Find the indexes of the 8 blocks to be used for data hiding; these indexes will form the secret key.

This step can be implemented by executing the following sequence of MATLAB operations:

```
%Chaotic key generation
 for i=1:NB
        x1=r1*x1*(1-x1);
        k11(i)=x1;
 end
 [aa key]=sort(k11);
%creating the secret key
 for i=1:8
        d(i)=find(key==i);
 end
```

- Step 3: Message hiding includes the following operations:
- Get MBM by converting decimal message to binary,
- Extract the required covering blocks,
- For each column in MBM, convert the block to binary. Let the LSBs of the block equal the associated column from MBM. Convert the stego block to decimal. Return the stego block to the stego image,
- Reshape the image to a 3D matrix to get the stego image.

This step can be implemented by executing the following sequence of MATLAB operations:

```
%Converting the message to binary
%to get MBM
 m3=dec2bin(m2,8);
 s1=a;
%Extract the required covering blocks
 for i=1:8
        block(i,:)=s1(1,d(i)-1)*L+1:d(i)*L);
 end
%Apply MSM columns hiding
 for i=1:8
        c1=block(i,:);
        c2=dec2bin(c1,8);
        c2(1:L,8)=m3(:,i);
        c3=bin2dec(c2);
        s1(1,(d(i)-1)*L+1:d(i)*L)=unit8(c3);
 end
%Reshape the image
        s=reshape(s1,n1,n2,n3);
```

The extracting function was implemented using the following operations sequence:

Step 1:
Inputs preparation:
- Read the image,
- Gegt the image size,
- Reshape the image to one row matrix,
- Get the PK; calculate the block size and the number of blocks.

This step was implemented by executing the following sequence of MATLAB operations:

```
%Stego image preparation
 a1=imread('C:\Users\win 7\Desktop\SLena.jpg^' );
 [n1 n2 n3]=size(a1);L1=n1*n2*n3
 s1=reshape(a1,1,L1);
%Get th PK
%and do calculation
 P1=0.01;
 r11=3.77;x11=0.15;
 BS1=L1*P;
 NB1=fix(l1/BS1);
```

Step 2:
Secret key generation:
- Run the CLMM to get the chaotic key,
- Sort the chaotic key to get an indices key,
- Find the indexes of the 8 blocks to be used for data extracting; these indexes will form the secret key.

This step was implemented by executing the following sequence of MATLAB operations:

```
%Generate chaotic key
 for i=1:NB1
        x11=r11*x11*(1-x11);
        k111(i)=x11;
 end
 [aa key]=sort(k111);
%Get the secret key
 for i=1:8
        d1(i)=find(key1==i);
 end
```

Step 3:
Message extracting:
- Extract the stego blocks,
- For each block convert the block to binary,
- Get the LSBs from the binary block,
- Let the associated column in the MBM equal the LSBs,
- Convert MBM to decimal,
- Convert the decimals to characters to get the secret message.

This step was implemented by executing the following sequence of MATLAB operations:

```
%Extract the stego blocks
 for i=1:8
      block(i,:)=s1(1,d1(i)-1)*L+1:d1(i)*L);
 end
%message extracting
 for i=1:8
        c11=block1(i,1:L);
        c21=dec2bin(c11,8);
        m51(1:L,i)=c21(:,8);
 end
 m61=bin2dec(m51)^';
 char(m61)
```

3. RESULTS AND DISCUSSION

The proposed method was implemented using various lengths of messages. The quality was tested visually as shown in Figure 11. The obtained stego image after hiding a message with 100,000 characters was very close to the covering image. This proves that the proposed method satisfied the quality requirements of the stego image.

The following short messages were processed using the proposed method, and the obtained extracted messages were always identical to the source messages. This confirms the quality of the extracted messages. Quality parameters between the covering image and the stego image were calculated for each used message. Table 3 shows the obtained quality parameters values.

- 'Secure LSB method',

– 'Protecting hidden message',
– 'Multiple blocks data hiding',
– 'Secret key to select covering blocks',
– 'Efficient method of data steganography'.

From Table 3 we can observe that the proposed method produced a stego image that is closed to covering image. The values of the obtained quality parameters satisfied the stego image quality requirements. The experiment was repeated using long messages, and the quality results shown in Table 4 confirm that the proposed method satisfied the quality requirements.

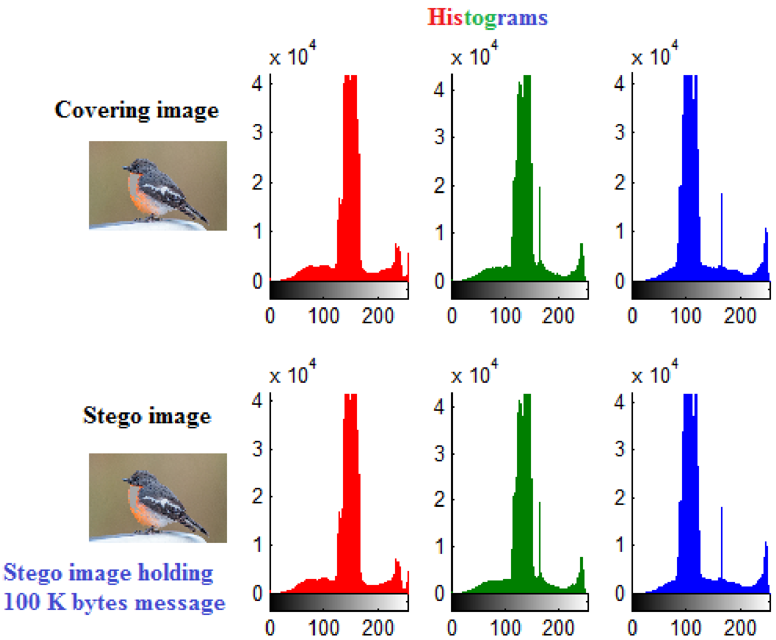

Figure 11. Covering and stego images (example)

Table 3. Quality results using short messages

Message number	Message length(character)	MSE	PSNR	CC	NSCR
1	17	0.000010132	225.8234	1.0000	0.0010
2	25	0.000016342	221.0431	1.0000	0.0016
3	27	0.000017486	220.3665	1.0000	0.0017
4	36	0.000024349	217.0553	1.0000	0.0024
5	38	0.000027454	215.8551	1.0000	0.0027
Remarks		Low	High	One	Low
		Satisfies quality requirements			

Table 4. Quality results using long messages

Message length (K bytes)	MSE	PSNR	CC	NSCR
1	0.00067263	183.8684	1.0000	0.0673
5	0.0033	167.8913	1.0000	0.3324
10	0.0067	160.8922	1.0000	0.6693
15	0.0101	156.7981	1.0000	1.0079
25	0.0167	151.7607	1.0000	1.6680
50	0.0335	144.7988	1.0000	3.3462
75	0.0502	140.7432	1.0000	5.0197
100	0.0668	137.8802	1.0000	6.6838
Remarks	Low	High	One	Low
	Satisfies quality requirements			

The MSE and PSNR values depend on the message length. Increasing the message length will increase the MSE value and decrease the PSNR value, as illustrated in Figure 12. However, the quality of the stego image remains good even for long messages.

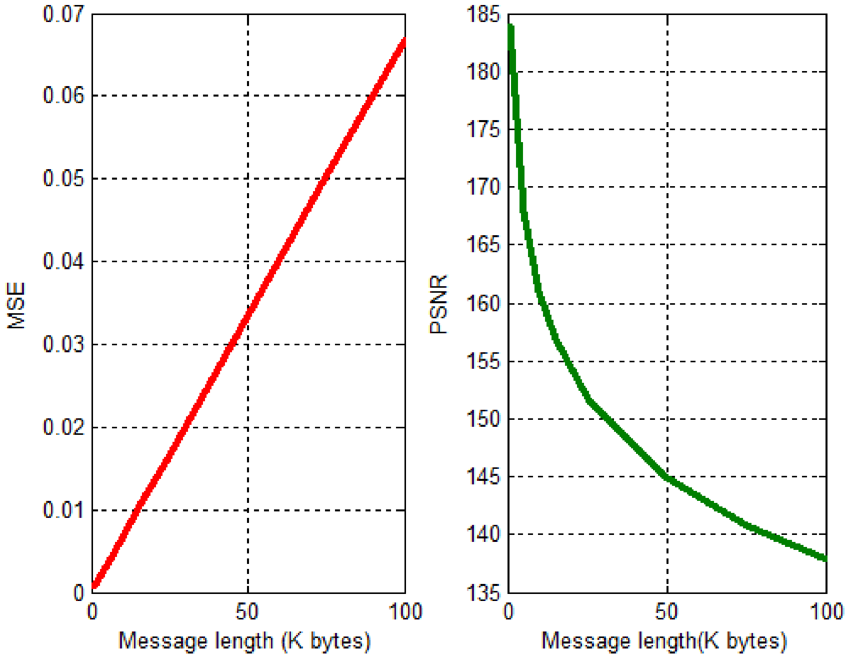

Figure 12. MSE and PSNR vs message length

The speed of the proposed method was tested by processing the previous messages. The hiding time (HT), extracting time (ET), total processing time short messages ($PT = HT + ET$), and throughput were calculated. Tables 5 and 6 show the obtained speed results when using short messages and long messages respectively.

Table 5. Speed results using short messages.

Message number	Message length (character)	HT (second)	ET (second)	PT (second)	TP (K bytes per second)
1	17	0.0200	0.0030	0.0230	0.7218
2	25	0.0230	0.0030	0.0260	0.9390
3	27	0.0230	0.0020	0.0250	1.0547
4	36	0.0230	0.0030	0.0260	1.3522
5	38	0.0250	0.0030	0.0280	1.3253

Table 6. Speed results using long messages

Message length (K bytes)	HT (second)	ET (second)	PT (second)	TP (K bytes per second)
1	0.0960	0.0120	0.1080	9.2593
5	0.2110	0.0460	0.2570	19.4553
10	0.4090	0.0920	0.5010	19.9601
15	0.6110	0.1410	0.7520	19.9468
25	1	0.2370	1.2370	20.2102
50	1.9560	0.4660	2.4220	20.6441
75	3.1210	0.7180	3.8390	19.5363
100	3.9210	0.9500	4.8710	20.5297
Average	1.4156	0.3327	1.7484	18.6927

The proposed method provided a good speed, with the required times exhibiting a linear relationship with the message length. Additionally, the throughput remains stable for long messages, as shown in Figure 13. The obtained results showed that the proposed method speeds up the process of message steganography significantly compared to other methods. Tables 7 and 8 show the results of comparisons based on using the covering images shown in Figure 14.

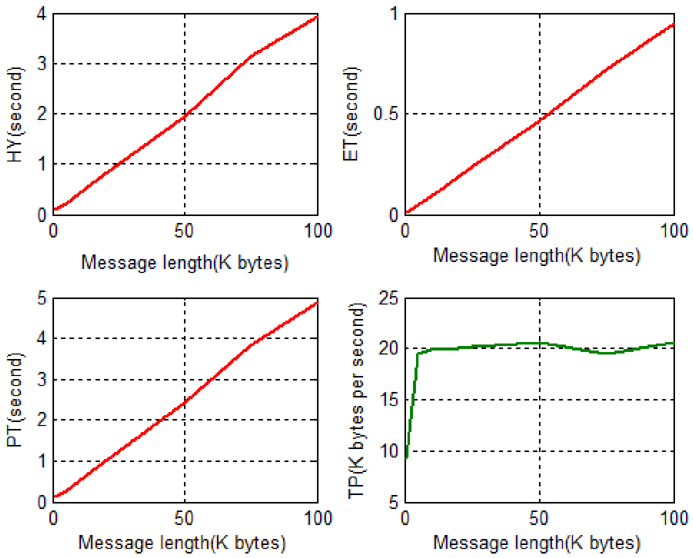

Figure 13. Speed parameters versus message length

Table 7. Speed comparisons

Proposed method			
SI	Fish	View	Boat
L(byte)	750	1000	1500
HT(second)	0.0630	0.0850	0.0820
ET(second)	0.0100	0.0130	0.0170
TT(second)	0.0730	0.0980	0.0990
CLSB			
L(byte)	750	1000	1500
HT(second)	0.062	0.078	0.093
ET(second)	0.078	0.062	0.109
TT(second)	0.1400	0.1400	0.2020
SLSB			
L(byte)	750	1000	1500
HT(second)	2.044	3.338	9.376
ET(second)	0.073	0.063	0.109
TT(second)	2.1170	3.4010	9.4850
DSLSB			
L(byte)	750	1000	1500
HT(second)	0.343	0.592	1.029
ET(second)	0.078	0.062	0.109
TT(second)	0.4210	0.6540	1.1380

Table 8. Speed up of the proposed method

Method	Average total time	Speedup of the proposed method
Proposed Method	0.0900	1.0000
CLSB	0.1607	1.7856
SLSB	5.0010	55.5667
DSLSB	0.7377	8.1967

Figure 14. Used covering images

4. CONCLUSION

A simple, secure, and efficient method of message steganography was proposed. This method is easy to use for both short and long messages steganography. It utilizes a simplified hiding and extracting function, where the hiding and extracting of the bits are performed by a simple batching method. The proposed method hided the message binary bits in different blocks and in separate locations of the covering image. Additionally, it utilizes a private key to protect the hidden message, which had a length of 192 bits, making it resilient against hacking attacks. The image was divided into blocks via the private key; which was also used to run a chaotic logistic map model that created a chaotic key with a number of elements equal to the number of image blocks. The created chaotic key was sorted to get an indices key, and the positions of blocks 1 to 8 were utilized for hiding and extracting. The proposed method underwent testing and implementation using various short and long messages. It was demonstrated that the method satisfied the quality requirements for both the extracted messages and the stego images. Additionally, the speed of the proposed method was tested, revealing a significant speedup compared to other existing LSB-based methods.

REFERENCES

[1] S. Kaur and S. Jindal, "Image steganography using hybrid edge detection and first component alteration technique," *International Journal of Hybrid Information Technology*, vol. 6, no. 5, pp. 59–66, Sep. 2013, doi: 10.14257/ijhit.2013.6.5.06.

[2] A. Martín, G. Sapiro, and G. Seroussi, "Is image stenography natural?," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2040–2050, Dec. 2005, doi: 10.1109/TIP.2005.859370.

[3] D. Bhattacharyya, A. Roy, P. Roy, and T. H. Kim, "Receiver compatible data hiding in color image," *International Journal of Advanced Science and Technology*, vol. 24, pp. 15–24, 2018.

[4] K. Chang, C. Jung, S. Lee, and W. Yang, "High quality perceptual steganographic techniques," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2939, Springer Berlin Heidelberg, 2004, pp. 518–531. doi: 10.1007/978-3-540-24624-4_42.

[5] G. C. Kessler, "Steganography: hiding data within data," *Windows & .NET Magazine*, 2001.

[6] G. Swain and S. K. Lenka, "Steganography using the twelve square substitution cipher and an index variable," in *2011 3rd International Conference on Electronics Computer Technology*, Apr. 2011, vol. 3, pp. 84–88, doi: 10.1109/ICECTECH.2011.5941806.

[7] H. Noda, M. Niimi, and E. Kawaguchi, "High-performance JPEG steganography using quantization index modulation in DCT domain," *Pattern Recognition Letters*, vol. 27, no. 5, pp. 455–461, Apr. 2006, doi: 10.1016/j.patrec.2005.09.008.

[8] K. Hempstalk, "A Java steganography tool," 2005. Accessed: Nov 19, 2023. [Online]. Available: http://diit.sourceforge.net/files/Proposal.pdf

[9] H. Motameni, M. Norouzi, M. Jahandar, and A. Hatami, "Labeling method in steganography," *World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering*, vol. 1, no. 6, pp. 1600–1605, 2007.

[10] D. Mohammed, "Message segmentation to enhance the security of LSB image steganography," *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 3, 2012, doi: 10.14569/ijacsa.2012.030310.

[11] R. J. Rasras, B. Zahran, M. R. A. Sara, and Z. AlQadi, "Developing digital signal clustering method using local binary pattern histogram," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 1, pp. 872–878, Feb. 2021, doi: 10.11591/ijece.v11i1.pp872-878.

[12] R. J. Rasras, M. R. A. Sara, Z. A. Al Qadi, and R. A. Zneit, "Comparative analysis of LSB, LSB2, PVD methods of data steganography," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 3, pp. 748–754, Jun. 2019, doi: 10.30534/ijatcse/2019/64832019.

[13] R. H. Ali, B. N. Dhannoon, and M. I. Hamel, "Arabic text steganography using lunar and solar diacritics," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 31, no. 3, pp. 1559–1567, Sep. 2023, doi: 10.11591/ijeecs.v31.i3.pp1559-1567.

[14] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, no. 9–10, pp. 1613–1626, Jun. 2003, doi: 10.1016/S0167-8655(02)00402-6.

[15] E. H. J. Halboos and A. M. Albakry, "Improve steganography system using agents software based on statistical and classification technique," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 12, no. 3, pp. 1595–1606, Jun. 2023, doi: 10.11591/eei.v12i3.4540.

[16] R. Das and I. Das, "Secure data transfer in IoT environment: Adopting both cryptography and steganography techniques," in *2016 Second International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, Sep. 2016, pp. 296–301, doi: 10.1109/ICRCICN.2016.7813674.

[17] S. A. Shawkat, N. Tagougui, and M. Kherallah, "Evolutionary programming approach for securing medical images using genetic algorithm and standard deviation," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 21, no. 6, pp. 1286–1297, Dec. 2023, doi: 10.12928/TELKOMNIKA.v21i6.25231.

[18] M. A. A. K. Al-Dabbas, A. Alabaichi, and A. S. Abbas, "Dual method cryptography image by two force secure and steganography secret message in IoT," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 6, pp. 2928–2938, Dec. 2020, doi: 10.12928/TELKOMNIKA.v18i6.15847.

[19] M. Fuad and F. Ernawan, "Video steganography based on DCT psychovisual and object motion," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 9, no. 3, pp. 1015–1023, Jun. 2020, doi: 10.11591/eei.v9i3.1859.

[20] R. J. Rasras, "Developing new Multilevel security algorithm for data encryption-decryption (MLS_ED)," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 6, pp. 3228–3235, Dec. 2019, doi: 10.30534/ijatcse/2019/90862019.

[21] S. K. Salim, M. M. Msallam, and H. I. Olewi, "Hide text in an image using Blowfish algorithm and development of least significant bit technique," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 29, no. 1, pp. 339–347, Jan. 2023, doi: 10.11591/ijeecs.v29.i1.pp339-347.

[22] R. J. Rasras, M. R. A. Sara, and Z. Alqadi, "Efficient method to message-image cryptography using reordered image-key,"

Traitement du Signal, vol. 40, no. 1, pp. 235–240, Feb. 2023, doi: 10.18280/ts.400122.
[23] X. Zhou, W. Gong, W. Fu, and L. Jin, "An improved method for LSB based color image steganography combined with cryptography," in *2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS)*, Jun. 2016, pp. 1–4, doi: 10.1109/ICIS.2016.7550955.
[24] M. Abu-Faraj, A. Al-Hyari, and Z. Alqadi, "A complex matrix private key to enhance the security level of image cryptography," *Symmetry*, vol. 14, no. 4, Mar. 2022, doi: 10.3390/sym14040664.
[25] M. M. Emam, A. A. Aly, and F. A. Omara, "An improved image steganography method based on LSB technique with random pixel selection," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 3, 2016, doi: 10.14569/IJACSA.2016.070350.
[26] T. Bhuiyan, A. H. Sarower, R. Karim, and M. Hassan, "An image steganography algorithm using LSB replacement through XOR substitution," in *2019 International Conference on Information and Communications Technology (ICOIACT)*, Jul. 2019, pp. 44–49, doi: 10.1109/ICOIACT46704.2019.8938486.

BIOGRAPHIES OF AUTHORS

Rashad J. Rasras received the Ph.D. degree from National Technical University (Kharkov Polytechnic Institute) 2001, with research in automated intelligent control systems. Currently. He is an associate professor at Department of Electrical Engineering, Al-Balqa' Applied University. His research interests include image processing, machine learning, signal processing and advanced computer architecture. He can be contacted at email: rashad.rasras@bau.edu.jo.

Mutaz Rasmi Abu Sara received the master's degree in computer science (database systems) in 2006 and Ph.D. from Saint Petersburg Electro technical University in 2010 with research and development of integrated database circuit components for CAD schematic, His research interest includes database systems, algorithms and machine learning. 2011-2020 he worked as assistant professor at Taibah University in K.S.A, currently he works as assistant professor at Palestine Ahliya University. He can be contacted at email: mutaz_abusara@yahoo.com.

Ziad AlQadi received the Ph.D. degree from National Technical University (Kiev Polytechnic Institute) 1986 with research in parallel computer architecture. Currently, he is a professor at the Department of Electrical Engineering, Al-Balqa' Applied University. His research interests include signal processing, parallel processing, and image processing. He can be contacted at email: natia_maw@yahoo.com.