Enhanced Algorithm for Steganography Based on Least Significant Bit and Secret Image Compression

1st Mayar khaled
Mathematics department
Faculty of science, Benha University
Egypt
mayar.khaled337@gmail.com

2nd Ahmed H. Abu El-Atta
Computer Science department
Faculty of Computers and artificial intelligence,
Benha University
Benha, Egypt
ahmed.aboalatah@fci.bu.edu.eg

***Abstract*— the data can be secured using various methods, such as cryptography, watermarking, and steganography. For secure communication, an image is utilized as the cover in image steganography. Hiding the secret image requires a large number of bits, thus you need a larger number of pixels in the cover, and this reduces imperceptibility. In this paper, large capacity is hidden in less space inside the cover. The secret image is compressed by only one of the two compression methods, the first is applied Huffman encoding and in the second method, the secret image is compressed to approximately half its size. The second method results in a higher Peak Signal-to-Noise Ratio (PSNR) rating and greater robustness. The tent chaotic map is used to obtain random locations. The secret image is hidden using the Least Significant Bit technique (LSB) in the cover pixels corresponding to resulting locations of the chaotic map in the last bit, two, or three randomly, which provides a high level of security. The complete method is carried out in MATLAB, and the results are analyzed, which shows greater protection for hidden information. According to the quantitative results, our proposed technique gets the highest PSNR between 44.00 and 47.986.**

Keywords— Chaotic map, steganography, Least Significant Bit technique, Huffman encoding, secret image compression.

I. INTRODUCTION

The widespread use of the internet these days has made information transmission speedy and easy. Since the information is communicated across a public network like the internet, data security remains a significant challenge. Information concealment is a useful approach for improving data transmission security. To generate stego data, this methodology hides all secret information in the cover file [1–3].

The two main forms of safety systems are encryption and information concealing [4]. Both types are responsible for information security, but their techniques are different. Cryptography is a method of encrypting secret information so that the secret data can only be decoded correctly by the intended receiver who has the secret key. Steganography and watermarking are two of the most used methods for information hiding. By embedding the data within the cover, steganography aims to hide any connection between the receiver and the sender, which keeps the secret information only visible to the receiver. Watermarking is a technique for protecting digital information owners' copyright [5].

Steganography is the method of concealing information in which secret information (text, image, audio, or video) is hidden within the cover file (text, image, audio, or video) in the form that the Human Visual System (HVS) cannot detect. As a response, people are unaware that there is a concealed message except for of the sender and receiver. Steganography is a Greek term that means "covered writing"; "stegano" means "cover" or "concealed" and "graphy" means "writing" [6].

Steganography has three characteristics [7, 8]: imperceptibility, capability (payload embedding), and robustness are essential in the face of common attacks. The number of hidden bits inserted for each cover pixel determines the capacity. More classified data could be inserted in a cover image if the capability was higher. PSNR is usually used to determine imperceptibility. The better the quality of the stego image, the higher the PSNR value. Robustness helps to secure personal data from being hacked or stolen.

Because of the redundancy in its representation, the image is a good medium for hiding data [9]. There are two domains to consider when employing an image as a cover file for hiding hidden data: spatial domain and frequency domain. The intensity values of the cover image are employed to conceal hidden data in the spatial domain. In the frequency domain, an image is translated to frequency coefficients, and the hidden information is concealed inside these coefficients [10].

The LSB technique is a simple and fast spatial domain hiding technique [11]. It substitutes the cover image's least significant bits with bits from the hidden data, resulting in the stego image that resembles a cover image.

Hiding the secret image requires a large number of bits, thus we need a larger number of pixels in the cover, and this reduces imperceptibility. In this paper, we aim to provide a solution to hide the same image sizes used in other research, but in fewer pixels to increase the imperceptibility of the stego image. The secret image is compressed by only one of the two compression methods, first by Huffman encoding and the second method is performed (secret image compression) to get better results. Applying the tent chaotic map to obtain random values that correspond to the pixels of a cover. To increase the security of hidden data, we create an equation to get random bits inside each pixel. Using the LSB technique to hide secret image was compressed in the last bit, two, or three.

The remainder of this work is laid out as follows: the related work is presented in section II, in section III the research method is given, in section IV the brief of results and discussions are given, and the conclusion is presented in section V.

II. RELATED WORK

The LSB approach and chaotic maps are used by many steganographers to hide information in images. In [12] secret image is hidden in pixels of the cover image's LSB, using a chaotic sequence created by the 1D logistic map. The binary

values of the secret image are separated into four independent two-bit values, which are stored in a separate two-dimensional array of the same size as the cover file. As a result, the secret image is hidden in the cover, and get the stego image.

Two techniques for image steganography in the spatial domain are discussed in this paper [13]. In steganography, these algorithms use chaos theory to detect locations of shuffled bits. The first technique uses the well-known LSB methodology, while the second technique uses a new method that looks for the same bits in secret information and a cover file. To extract the shuffled location bits, a modified logistic chaotic map is used in the chaotic map to produce integer chaotic values.

The method was used in [14] based on reversing LSBs and some mathematical calculations. The mathematical operations are finding the minimum and maximum values. The image file is broken into two equal-sized sections. Insert quotients of the value of maximum in five LSBs of the first pixel in the first part of the cover image by reversing values of these five LSBs, and in the second part of the cover, insert the remainder of the maximum value in 3 LSBs of the first pixel. The stego image is created when all secret data have been embedded in the cover image.

The method was based on 2^k correction and the Canny edge detector. Coherent bit length and Huffman encoding are also used in the new method. The Huffman table is then created. Before the secret data are inserted, Huffman encoding is used to code it according to the Huffman table. Replace L bits of payload message by L bits of coherent bit length L calculated based on relevant edge pixels. Finally, the 2^k correction approach is used to improve the imperceptibility of the stego image [15].

Dash et al. [16] suggested an improved chaotic encryption-based image hiding system. To secure secret data, the chaotic neural network system (CNN) was used to implement the cryptography process. After the encryption method, the encoded data were scrambled and inserted into the edge region of the cover image with 3-bit LSB is used.

N. Kar et al. [17] used a DNA-based approach for sending data hidden within a video file. The video file must be transformed into image frames. Burger chaotic map is used. The Least Significant Bit substitution approach is used to choose random frames and hide data in them at random locations.

Using a genetic algorithm (GA), a high-capacity image steganography method based on hidden information updating has been developed. New methodology in [18] the hidden information was inserted using LSB substitution steganography. Before secret information is embedded in the LSBs of the cover image, it is arranged and modified. GA is in control of the parameters that are utilized to arrange and update the hidden information. Various sets of images were used for the 1 bpp, 2 bpp, and 3 bpp payload capacity of this paper.

E. Abbood et al [4] provided a new proposed approach for hiding a secret message in a grayscale image utilizing a random methodology and a simple hash function. The confidential text is distributed throughout all rows except for the last one on the cover image. The amount of secret message bits is divided by the total number of image rows to achieve this. A pseudo-random number generator is used in each row to identify the position of the columns that are needed to conceal secret message by using the LSB approach.

- Advantage of previous work

In [12] the use of the chaotic map increased security.

In [13] a modified logistic map was used and looking for similar bits in the secret information and a cover file.

In [14] high robustness.

In [15] high robustness and good imperceptibility.

In [16] cryptography process to secure secret data.

In [17] high robustness against attacks.

In [18] higher visual quality.

In [4] higher security.

- Disadvantages of previous work

In [12] hiding the data in the last 2 bits only and not compressing it led to an increase the number of cover pixels that were used to hide the secret information.

In [13] lower payload capacity.

In [14] Increase the number of cover pixels that were used to hide the secret information.

In [15] lower payload capacity.

In [16] lower payload capacity.

In [17] lower imperceptibility and payload capacity.

In [18] poor robustness.

In [4] lower imperceptibility.

III. RESEARCH METHOD

In this proposed method, the secret image is compressed by one of the following two methods: Huffman encoding (HE) or secret image compression (SIC). Applying the tent chaotic map equation to obtain random values are between 0 and 1 that correspond to the pixels of a cover image. We create the equation (3) mentioned in section D, to get the random bits that are hidden in them (the last bit, two, or three) by using the LSB technique, as shown in fig.1

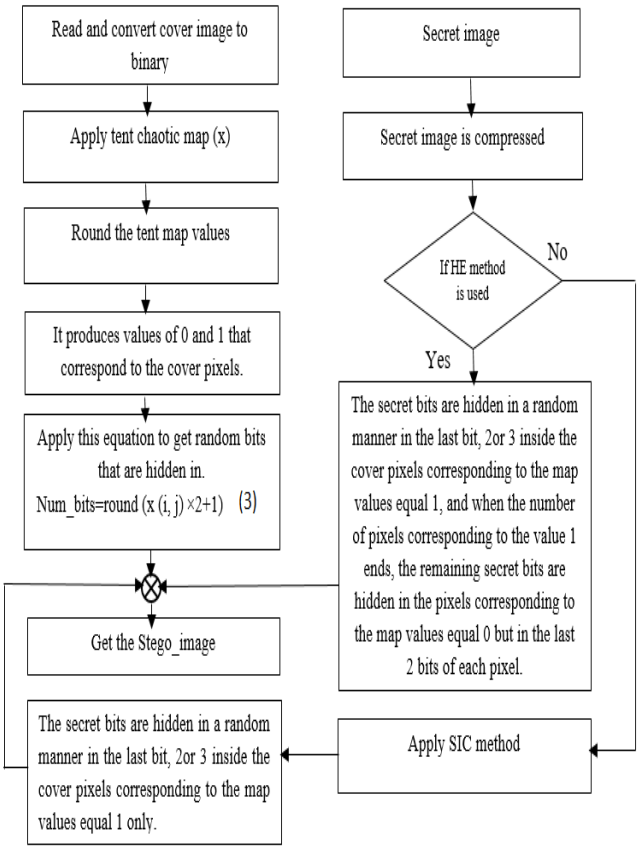


Fig. 1 Work chart for proposed algorithm processed stages.

A. *Huffman encoding (HE)*

The secret image is first encoded using the Huffman Table before being embedded into the cover image. David A. Huffman introduced Huffman encoding in 1952 [19]. It's used to compress data without losing quality. It constructs a binary tree by revealing the data in small bits. Firstly it arranges the signals and their frequencies in a sorted array. In the beginning, each symbol with its frequency represents a leaf node. Then two symbols with the lowest frequencies are combined their frequencies together. The parent node receives the sum. Repeat the process until there is only one node left, which is known as the root node. Allocate 0's and 1's to each node, then read from the root node to the leaf to convert the codes. A Huffman Table is created in this way.

B. *Secret image compression (SIC)*

The secret image is compressed to approximately half its size by dividing the image into blocks, calculating the average for each block, and then round the average values, the value of each pixel and the average values are transformed into a binary, 4 bits of Most Significant Bit (MSB) for each pixel and each average is hidden inside the cover to get a stego image. This leads to an increase in the value of PSNR and higher security. To get the secret image from stego image, 4 bits of MSB for an average of each block are combined with 4 bits of MSB for each pixel in this block. In this method, we get an image close to the secret image as shown in Fig 4.

In the case of the secret image size equal to 128×128, divide the matrix into 16 × 16 blocks, and each block contains 8 × 8 pixels, calculate the average for each block, and round the average values, obtain a matrix (16 × 16) for the average values, and merge first 4-bits for each pixel inside the block with first 4-bits of the average corresponding to this block to get a secret image.

In the case of the secret image size equal to 256×256, divide the matrix into 16 × 16 blocks, and each block contains 16 × 16 pixels, with the application of the same previous method.

	1	2	3	4	5	6	7	8
1	0111	0111	1000	1000	0111	1000	1000	1000
2	0111	0111	1000	1000	1000	1000	1000	1000
3	0111	0111	1000	1000	1000	1000	1000	1000
4	0111	0111	1000	1000	1000	1000	1000	1000
5	0111	0111	1000	1000	1000	1000	1000	1000
6	0111	0111	1000	1000	1000	1000	1000	1000
7	0111	0111	1000	1000	1000	1000	1000	1000
8	0111	0111	1000	1000	1000	1000	1000	1000

Fig. 2 Matrix of the first 4-bits for each pixel inside the first block in the secret image.

Fig. 2 shows the first block of the secret image blocks and the value of each pixel in this block after taking only the first 4 bits that are hidden in the cover image.

For example:

To get a secret image, after extracting the secret and average bits and converting them to blocks, we take the first block as an example

(1, 1) = 0111, (1, 3) =1000, (3, 3) =1000, ……, (8, 8)
And average corresponding to this block (1, 1) =130= 10000010, as shown in Fig 4, merge first 4-bits for each pixel inside the first block with first 4-bits of the average (1, 1), as shown in Fig. 3 to get the first block for a secret image.
(1, 1) = 0111, 0111⊕1000 = 01111000
(1, 3) = 1000, 1000⊕1000 = 10001000
(3, 3) = 1000, 1000⊕1000 = 10001000

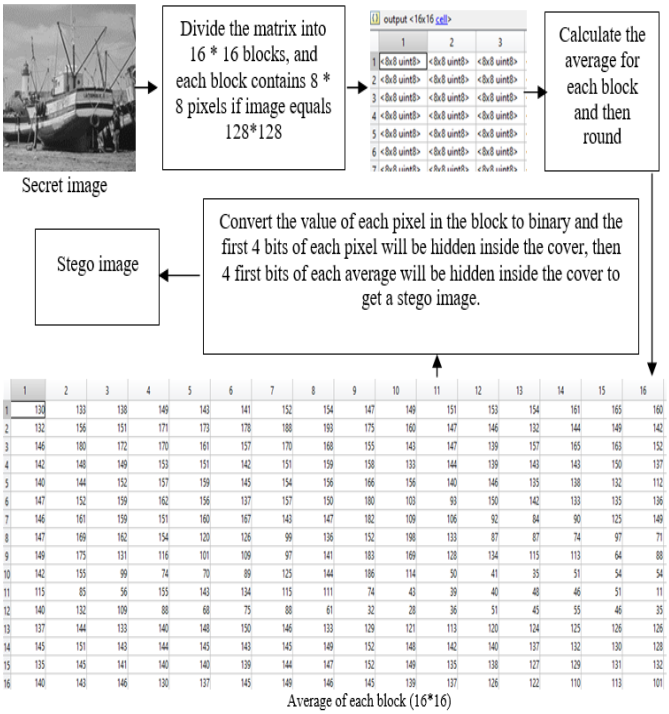
	1	2	3	4	5	6	7	8
1	01111000	01111000	10001000	10001000	01111000	10001000	10001000	10001000
2	01111000	01111000	10001000	10001000	10001000	10001000	10001000	10001000
3	01111000	01111000	10001000	10001000	10001000	10001000	10001000	10001000
4	01111000	01111000	10001000	10001000	10001000	10001000	10001000	10001000
5	01111000	01111000	10001000	10001000	10001000	10001000	10001000	10001000
6	01111000	01111000	10001000	10001000	10001000	10001000	10001000	10001000
7	01111000	01111000	10001000	10001000	10001000	10001000	10001000	10001000
8	01111000	01111000	10001000	10001000	10001000	10001000	10001000	10001000

Fig. 3 First block in secret image after merging 4-bits in each pixel in the first block with 4-bits of the first pixel in an average matrix.

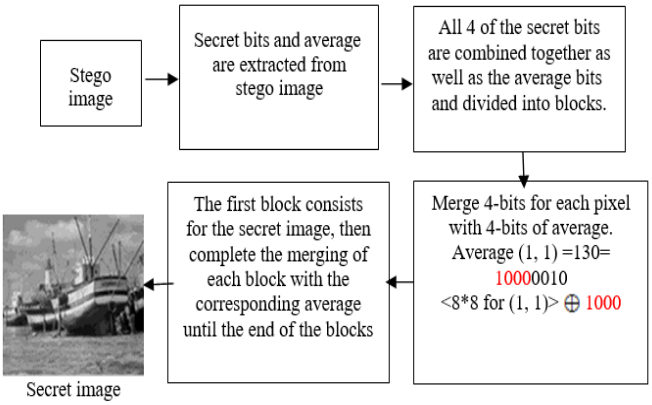
Applying the same previous method with all blocks for example,
Average (1, 2) =133 = 10000101, merge 4MSB for each pixel inside the block (1, 2) ⊕1000.
Average (1, 4) =149 = 10010101, merge 4MSB for each pixel inside the block (1, 4) ⊕ 1001.

.
.

Average (16, 16) =101 = 01100101, merge 4MSB for each pixel inside the block (16, 16) ⊕ 0110.



(a) Secret image compression



(b) Secret image decompression.

Fig. 4 The proposed algorithm of secret image compression and decompression.

Huffman encoding and secret image compression methods are used in this paper to compress the secret image to hide the bits of these images in a fewer number of cover pixels and increase the PSNR value.

When using Huffman encoding, the number of bits has been reduced by a small percentage, so the secret image compression method is applied to reduce the secret bits by about half, hide fewer bits inside the cover pixels and increase imperceptibility. HE has one advantage over SIC, it is a lossless type, while SIC is a loss type, it gives an approximation of the original image.

C. *Chaotic map*

Due to its usefulness in high safety maintenance for digital steganography, the use of chaotic maps has increased in recent years. Chaotic maps were described in steganography and encryption [20, 21] as a good choice for properties such as ergodicity, random behavior, and parameters of control.

The output of the tent map (1) is one-dimensional nonlinear systems [22], therefore, we encountered a problem in using the tent map in this way, and for this reason, we converted the tent map (1) into a two-dimensional array to produce Eq. (2), so that we can use it in this proposed algorithm.

$$T(x) = \begin{cases} ux, & x<0.5 \\ u(1-x), & x\geq 0.5 \end{cases} . \quad (1)$$

Specifies an iterative map by

$X_{n+1} = T(x_n)$.
Where $x_n \in [0, 1]$, $0 \leq u \leq 2$.

$$x(i + Q, j + 1 - (bQ)) = \begin{cases} u\, x\,(i,j), & \text{if } x\,(i,j)<0.5 \\ u\,(1-x(i,j)), & \text{otherwise} \end{cases} . \quad (2)$$

Where x (i, j) $\in$ [0, 1], i is the row number, j is the column number, b = sum of the image columns, $0 \leq u \leq 2$, Q= ⌊j/b⌋.

D. *Least Significant Bit Technique*

The LSB technique is a simple and fast spatial domain hiding technique [10]. It replaces the least significant bits of pixels in a cover file with bits from secret information, resulting in a stego image that resembles the cover image. The principle behind this technique is that the least Significant bits reflect only poor information in an image, and human eyes are incapable of detecting minor changes in those bits. [23].

LSB technique applies to the cover pixels corresponding to resulting locations of the chaotic map. We created an Eq. (3) to get the random bits that are hidden in (the last one, two, or three bits randomly) by multiplying each value from the chaotic map by 2, adding one, and then rounding the resulting value.

$$Num\_bits\ (i, j) = round\ (x\ (i, j) \times 2+1). \quad (3)$$

Where x (i, j) equals values resulting from the chaotic map.

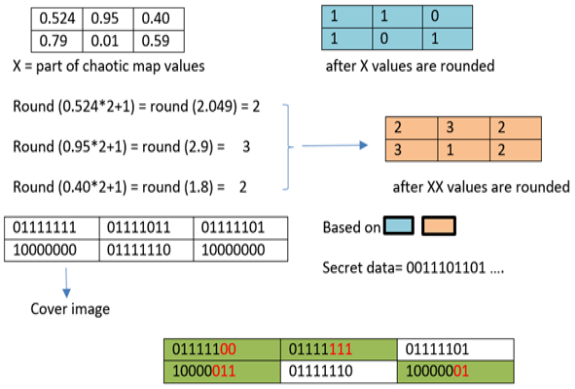


Fig. 5 Hide secret data.

In Fig. 5, Eq. (3) is applied to the chaotic map values before rounding to see which random bits of cover pixels are hidden inside. Based on the approximation of the values obtained from the chaotic map and the values obtained from Eq. (3), the secret bits are hidden using the LSB technique.

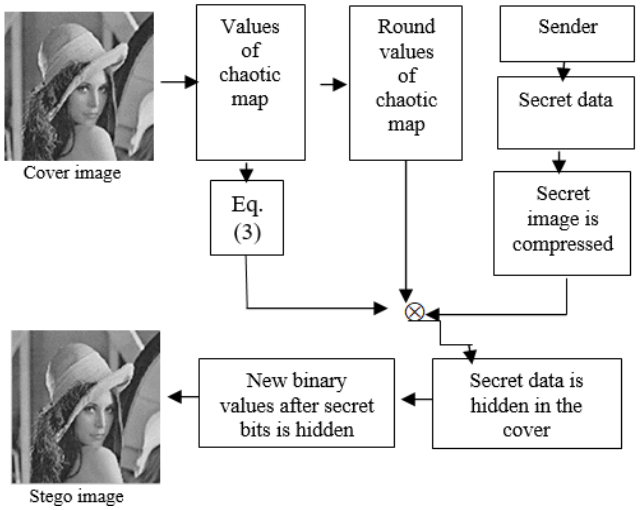


Fig . 6 Proposed algorithm processed stages.

In Fig 6, the chaotic map is applied, the resulting values are between 0 and 1, and the values are rounded. Equation (3) is applied to the chaotic map values before rounding to see which random bits of cover pixels are hidden inside. Secret information is concealed in cover pixels after converting the cover to a binary, get a stego image.

Algorithm of the embedding an image when using SIC

1. Converting cover image to binary.
2. The secret image is compressed by the SIC method in this paper, and binary bits stream is achieved.
3. Through the chaotic map (2), the secret data are hidden, then the average bits are hidden in the cover pixels corresponding to the chaotic map whose value is equal to one.
4. Applying Eq. (3) to hide a secret image in the last bit, 2bits or 3bits.
5. Get the stego image.

In this algorithm, transferring the image cover to a binary. The secret image is compressed to approximately half its size by dividing the image into blocks, calculating the average for each block, and 4 bits of MSB for each pixel is hidden inside the cover, the first 4 average bits are concealed in a cover after the secret bits are concealed, to get a stego image. Select the pixels in which we hide the secret bits using chaotic map, and randomly choose the bits inside each pixel by using Eq. (3).

Algorithm of the embedding an image when using HE

1. Converting cover image to binary.
2. Applying Eq. (3) to conceal the secret image within the last bit, 2bits or 3bits.
3. The hidden bits are concealed randomly in the last bit, two, or three of each pixel whose value is one from the chaotic map and the remaining secret bits are concealed within the last 2bits of each pixel with a value equals zero resulting from the chaotic map until the secret bits are hidden.
4. Get a stego image.

In this algorithm, an image is compressed by a Huffman encoding, this technique compresses the bits at a small rate, and the cover contains a high amount of secret bits. Because of the large size of secret information, the hidden bits are concealed randomly in the last bit, two or three of each pixel whose value is one from the chaotic map, and the remaining secret bits are concealed in the last 2bits of each pixel with a value equals zero resulting from the chaotic map until the hidden bits are concealed.

Algorithm of the extracting image when using SIC

1. Read the stego image.
2. Using the chaotic map, select pixel positions in the stego image.
3. Create a binary image from a stego image.
4. Looking for where the secret bits are hidden.
5. Retrieve bits of the secret data where the last 1024 bits are average bits and other bits are secret bits.
6. Repeat step 4 until you've obtained all of the secret bits.
7. Add all 4 bits together and then divide the data into blocks.
8. Combine the first 4 bits of a secret image with the first 4 bits of the average block to this pixel.

In this algorithm, the stego image is split into blocks, the pixels that have been hidden inside are defined by a chaotic map. The stego image is converted into a binary, the last 1024 bits in the bits extracted from the stego image are the average bits and the rest of the bits are the secret bits of the image.

Algorithm of the extracting image when using HE

1. Read the stego image.
2. Using the chaotic map, select pixel positions in the stego image.
3. Create a binary image from a stego image.
4. Search for the location of the hidden bits.
5. Repeat step 4 until you've obtained all of the secret bits.
6. Huffman decoding is executed.
7. The secret image is achieved.

IV. Experimental Results and Discussion

This section shows the effects of applying the proposed method of hiding secret data in various images and calculating the accuracy of the resulting images using the PSNR and Mean Square Error (MSE) measurements. To acquire the PSNR value, we first must calculate the MSE value of a stego image. Equation (4) [15] can be used to calculate the MSE of a stego image, and Eq. (5) can be used to calculate the PSNR of a stego image. In Eq. (4), M and N represent the image's rows and columns, respectively. The pixel intensities of the ijth position of a cover image and a stego image, respectively, are I(i, j) and S(i, j).

$$\mathrm{MSE} = \frac{1}{M\,N} \sum_{i=1}^{M} \sum_{j=1}^{N} (I(i,j) - S(i,j))^2. \quad (4)$$

$$\mathrm{PSNR} = 10\log 10 \frac{(255)^2}{\mathrm{MSE}}\ . \quad (5)$$

The Structural Similarity Index Measure (SSIM) is a mode of comparison metric for determining how similar two images are. It is determined as follows:

$$SSIM(I,S) = \frac{(2\mu_I\mu_S + c)(2\sigma_{IS} + c_2)}{(\mu^2{}_I + \mu^2{}_S + c_1)(\sigma^2{}_I + \sigma^2{}_S + c_2)}. \quad (6)$$

$c_{1=(k_1 l)^2}$, $c_{2=(k_2 l)^2}$

Where μ_I and μ_S are the average value of the intensity of I and S images (original and stego). $\sigma^2{}_I$ Is the variance of I, $\sigma^2{}_S$ is the variance of S and $\sigma^2{}_{IS}$ is the covariance of I and S. The two stabilizing parameters are c_1 and c_2 , L is the dynamic range of pixel values ($2^{\#bits\ per\ pixel}$ -1) and the contents $k_1 = 0.01$and $k_2 = 0.03$.

The proposed technique is implemented using Matlab R2013a. Dataset of standard test images is used in this paper to test the proposed method. In [12, 14] the authors used the cover images with size 256×256 and secret grayscale images with size 128×128. In [18] the authors used the cover images (512×512) and test pattern images as a secret image (128×128). We are comparing the proposed method using the same dataset.

TABLE I. PSNR Results for images with size 256 × 256 and secret grayscale image with size 128×128

Cover image	In [12]	In [14]	Proposed technique(HE method)	Proposed technique(SIC method)
Lena	44.53	41.598	44.5994	47.9698
Baboon	44.54	41.620	44.6312	47.9860
Elaine	44.53	41.593	44.6344	47.9779

TABLE II. MSE Results for images with size 256 × 256 and secret grayscale image with size 128×128

Cover image	In [12]	Proposed technique(HE method)	Proposed technique(SIC method)
Lena	2.28	2.25	1.03
Baboon	2.28	2.24	1.03
Elaine	2.28	2.24	1.03

TABLE I shows the PSNR result for the different used images with size (256× 256). Compared with the method was used in [12, 14] the result of PSNR in the proposed method is better than other studies. Our PSNR value is 47.9 and 44.6 dB when using SIC and HE, respectively. TABLE II shows the results of MSE and the lower value of the MSE, the better the result. The result of our proposed method is 1.03 and 2.25 when using SIC and HE, the method was used in [12] was 2.28 so our proposed technique is better for imperceptibility.

In the proposed algorithm, the value of PSNR when using the SIC method equals 47.98 is better than the PSNR value in the HE method that equals 44.6 because the number of secret bits that were hidden was less in the case of compression by the SIC method.

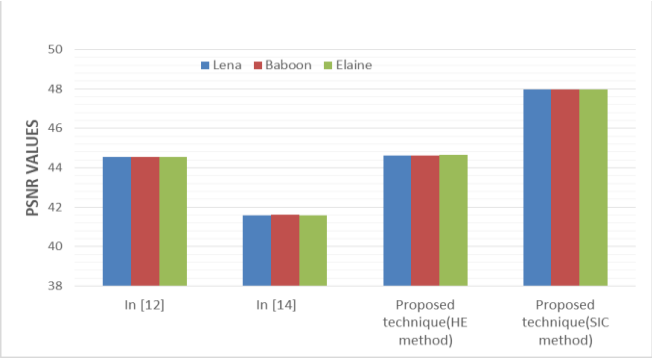


Fig. 7 RSNR values belong to TABLE I.

TABLE III. PSNR Results for images with size 512 × 512 and secret grayscale image (test pattern image was used in [18]) with size 256*256.

Cover image	In [18] at 2bpp	Proposed technique(HE method)	Proposed technique(SIC method)
Airplane	45.23	45.65	47.17
Mandrill	45.19	45.60	47.18
Pepper	45.34	45.60	47.17

TABLE IV. SSIM Results for images with size 512 × 512.

Cover image	In [18] at 2bpp	Proposed technique(HE method)	Proposed technique(SIC method)
Airplane	0.9963	0.9969	0.9975
Mandrill	0.9987	0.9990	0.9994
Pepper	0.9958	0.9973	0.9981

TABLE III shows the PSNR result for the different used images with size 512×512. Compared with the method was used in [18] the result of PSNR in the proposed method is better than other studies. Our PSNR value is 47.2 and 45.6 dB when using SIC and HE, respectively. TABLE IV shows the results of SSIM and the higher value of the SSIM, the better the result. The result of our proposed method is between 0.9975 and 0.9994 and the method was used in [18] was between 0.9969 and 0.9990 so our proposed technique is better.

In the proposed method, the value of PSNR when using the SIC method equals 47.2 is better than the PSNR value in the HE method that equals 45.6 because the number of secret bits that were hidden was less in the case of compression by the SIC method.

TABLE V. Experimental PSNR Results for images with size 512 × 512 and secret data bits (was used in [18]).

Cover image	[24]	[25]	In [18] at 3bpp	Proposed technique (HE method)	Proposed technique (SIC method)
Blonde	38.32	39.15	39.84	44.00	46.24
cameraman	38.31	39.27	39.82	44.01	46.25
Lena	38.25	39.38	39.88	43.98	46.23

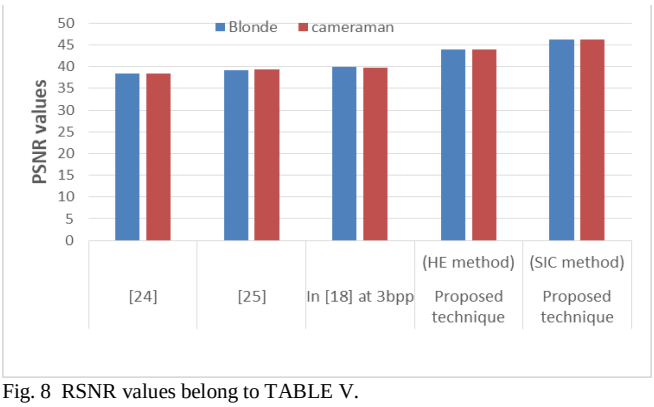


Fig. 8 RSNR values belong to TABLE V.

The proposed technique is better than the method used in [12, 14, 18] because the secret image is compressed first, and the cover image pixel location is selected at random using a tent chaotic map, the LSB bits for embedding the secret image bits are chosen randomly based on Eq. (3), it reducing the security risk and increasing the efficiency of the proposed

technique. In [12] 1D logistic map was used and the secret bits were divided into 2 bits together and embedded into the cover of the image, and in [14] each pixel's five or four LSBs are used for embedding, therefore, the rate of difference between cover and stego image is greater and leads to a decrease in the PSNR value. Without compression of the secret image in each [12, 14] and increasing the secret bits leads to an increase in the number of cover pixels used for hiding and a decrease in imperceptibility.

The method used in [18] is better than that used in [12, 14] because GA is used to get the optimal value of the parameters to arrange and update secret information. The method in [12] is better than [14] because secret data were hidden randomly in the cover.

Image histogram

An image histogram is a form of the histogram that operates on a digital image of the tonal distribution as a graphical representation. The majority of new digital cameras have histograms of images on them. Photographers may use it as an aid to display the distribution of captured colors and whether the brightness of the image has been lost due to blown-out highlights or black-out shadows. Tonal changes are defined by the horizontal axis of the graph, while the total pixel count in that particular tone is determined by the vertical axis [26].

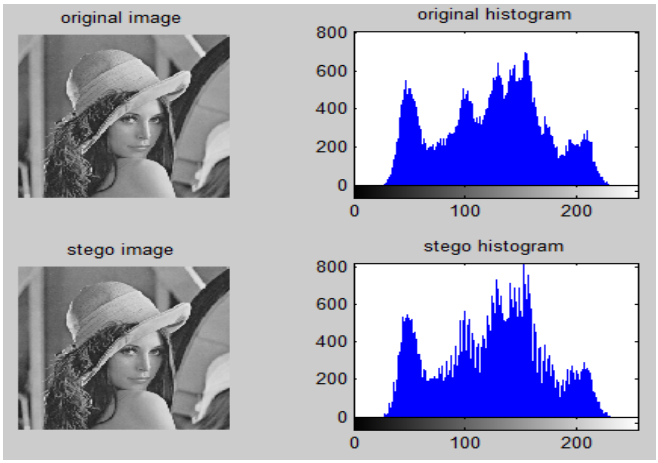


Fig . 9 Histogram of the Lena image.

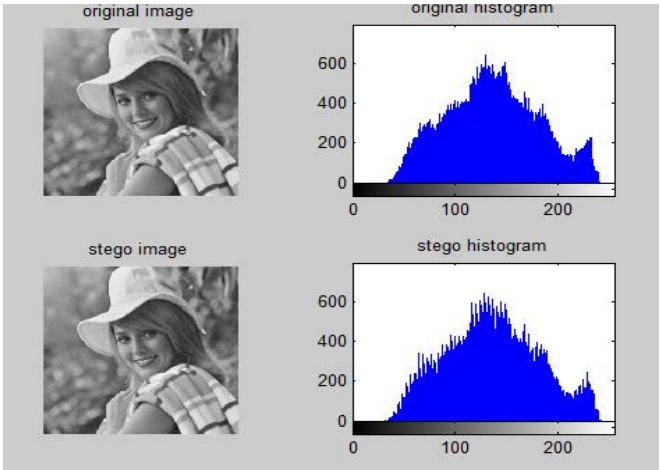


Fig. 10 Histogram of the Elaine image.

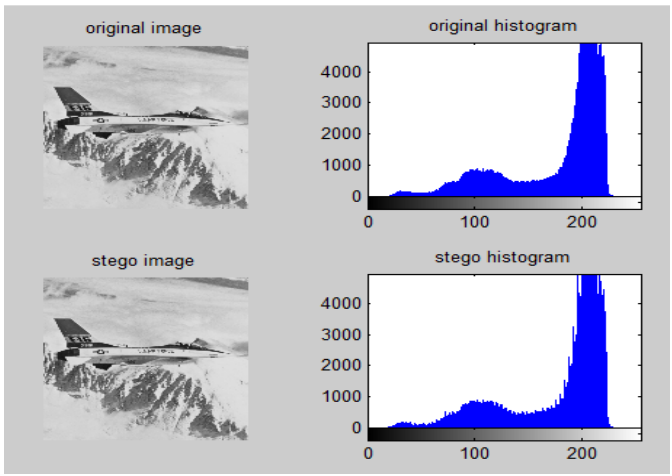


Fig. 11 Histogram of the Airplane image.

In image histogram we use Elaine and Lena images which are used in table.1 and the Airplane image which is used in table.3 when HE method is used. The histogram of the cover image and the stego image is shown in Figs 9, 10, and 11 and shows how close the histogram of both the cover and stego image is.

V. CONCLUSIONS

The secret image is embedded into the cover based on the chaotic map, LSB method, and image compression is which propose in this paper. Our algorithm incorporates the secret hidden data bits into the random bits for pixels of the cover image. Hiding data in a random way led to an increase in the confidentiality of information. Compressing the secret image using the SIC method increases capacity, robustness, and imperceptibility, so this method is better than the HE method. The test focused on PSNR, MSE, SSIM, histogram measures, and comparisons with other previous techniques using other methods. The current method increases the imperceptibility, the capacity of embedding and improves the quality of the stego image, according to experimental results. In future work, compress the secret image in another method to improve the quality of the image after compression. Find a method to select the hidden bits to be similar to the secret bits to improve the PSNR values.

ACKNOWLEDGMENT

The authors are grateful to the confidential referee for carefully reviewing the original manuscript and providing helpful comments that improved the presentation of the data and highlighted critical details.

REFERENCES

[1] M. Fateh, M. Rezvani, and Y. Irani, “A new method of coding for steganography based on LSB matching revisited,” *Secur. Commun. Networks*, vol. 2021, 2021.

[2] S. Farrag and W. Alexan, “A high capacity geometrical domain based 3d image steganography scheme,” in *2019 International Conference on Advanced Communication Technologies and Networking (CommNet)*, 2019, pp. 1–7.

[3] R. Bhardwaj and V. Sharma, “Image steganography based on complemented message and inverted bit LSB substitution,” *Procedia Comput. Sci.*, vol. 93, pp. 832–838, 2016.

[4] E. A. Abbood, R. M. Neamah, and S. Abdulkadhm, “Text in image hiding using developed LSB and random method,” *Int. J. Electr. Comput. Eng.*, vol. 8, no. 4, pp.

2091–2097, 2018, doi: 10.11591/ijece.v8i4.pp2091-2097.
[5] P. Rai, S. Gurung, and M. K. Ghose, "Analysis of image steganography techniques: a survey," *Int. J. Comput. Appl.*, vol. 114, no. 1, 2015.
[6] K. Muhammad, J. Ahmad, N. U. Rehman, Z. Jan, and R. J. Qureshi, "A secure cyclic steganographic technique for color images using randomization," *arXiv Prepr. arXiv1502.07808*, 2015.
[7] L. M. Marvel, C. T. Retter, and C. G. Boncelet, "A methodology for data hiding using images," in *IEEE Military Communications Conference. Proceedings. MILCOM 98 (Cat. No. 98CH36201)*, 1998, vol. 3, pp. 1044–1047.
[8] B. Li, J. He, J. Huang, and Y. Q. Shi, "A survey on image steganography and steganalysis," *J. Inf. Hiding Multimed. Signal Process.*, vol. 2, no. 2, pp. 142–172, 2011.
[9] O. S. Khalind, "New methods to improve the pixel domain steganography, steganalysis, and simplify the assessment of steganalysis tools." University of Portsmouth, 2015.
[10] R. Amirtharajan and J. B. B. Rayappan, "An intelligent chaotic embedding approach to enhance stego-image quality," *Inf. Sci. (Ny).*, vol. 193, pp. 115–124, 2012.
[11] S. Mukherjee and G. Sanyal, "A physical equation based image steganography with electro-magnetic embedding," *Multimed. Tools Appl.*, vol. 78, no. 13, pp. 18571–18593, 2019.
[12] S. Rajendran and M. Doraipandian, "Chaotic map based random image steganography using LSB technique," *Int. J. Netw. Secur.*, vol. 19, no. 4, pp. 593–598, 2017, doi: 10.6633/IJNS.201707.19(4).12.
[13] O. N. Kadhim and Z. M. Hussain, "Information Hiding using Chaotic-Address Steganography," *J. Comput. Sci.*, vol. 14, no. 9, pp. 1247–1266, 2018.
[14] D. Nashat and L. Mamdouh, "An efficient steganographic technique for hiding data," *J. Egypt. Math. Soc.*, vol. 27, no. 1, pp. 1–14, 2019.
[15] S. Sun, "A novel edge based image steganography with 2^k correction and Huffman encoding," *Inf. Process. Lett.*, vol. 116, no. 2, pp. 93–99, 2016.
[16] S. Dash, M. N. Das, and M. Das, "Secured image transmission through region-based steganography using chaotic encryption," in *Computational Intelligence in Data Mining*, Springer, 2019, pp. 535–544.
[17] N. Kar, K. Mandal, and B. Bhattacharya, "Improved chaos-based video steganography using DNA alphabets," *ICT Express*, vol. 4, no. 1, pp. 6–13, 2018.
[18] P. D. Shah and R. S. Bichkar, "Secret data modification based image steganography technique using genetic algorithm having a flexible chromosome structure," *Eng. Sci. Technol. an Int. J.*, 2021, doi: 10.1016/j.jestch.2020.11.008.
[19] S. S. M. Than, "Secure data transmission in video format based on LSB and Huffman coding," Int. J. Image, Graph. Signal Process, vol. 12, no. 1, pp. 10–17, 2020.
[20] R. S. Bhogal, B. Li, A. Gale, and Y. Chen, "Medical image encryption using chaotic map improved advanced encryption standard," IJ Inf. Technol. Comput. Sci., vol. 8, pp. 1–10, 2018.
[21] M. Jain, "Medical image steganography using dynamic decision tree, piecewise linear chaotic map, and hybrid cryptosystem," *Int. J. Appl. Eng. Res*, vol. 13, no. 15, pp. 12353–12363, 2018.
[22] C. Li, G. Luo, K. Qin, and C. Li, "An image encryption scheme based on chaotic tent map," *Nonlinear Dyn.*, vol. 87, no. 1, pp. 127–133, 2017.
[23] S. Gupta, A. Goyal, and B. Bhushan, "Information hiding using least significant bit steganography and cryptography," Int. J. Mod. Educ. Comput. Sci., vol. 4, no. 6, p. 27, 2012.
[24] C.-N. Yang, S.-C. Hsu, and C. Kim, "Improving stego image quality in image interpolation based data hiding," *Comput. Stand. Interfaces*, vol. 50, pp. 209–215, 2017.
[25] G. S. Yadav and A. Ojha, "Hamiltonian path based image steganography scheme with improved imperceptibility and undetectability," *Appl. Soft Comput.*, vol. 73, pp. 497–507, 2018.
[26] J. H. Park, J. J. Jung, and G. B. Kim, "A Feature Vector Generation Technique through Gradient Correction of an Outline in the Mouth Region," *J. Korea Multimed. Soc.*, vol. 17, no. 10, pp. 1141–1149, 2014.