STEGANOGRAPHIC SUBSTITUTION OF THE LEAST SIGNIFICANT BIT DETERMINED THROUGH ANALYSIS OF THE COVER IMAGE AND THE ENCRYPTED MESSAGE

Martha Angelica Garcia-Villa, Ricardo Francisco Martinez-Gonzalez
Juan Francisco Mejia-Perez, Miguel Valerio-Canales
and Yesenia Isabel Moreno-Pavan

Tecnologico Nacional de México/IT Veracruz, Veracruz, Mexico

ABSTRACT

The present workproposes to perform an analysis of the similarities between the least significant two bits of the cover image and multiple series of two-bit-length encrypted frames, all of them from the crypto-message. After finding the most similar frame, we proceed to substitute it into the cover image; nevertheless, to provide a proof of the improvement from using itor the least similar one, the statistics from both cases are obtained.Providing information that the more similar the frame is, the better statistics the stego-image has. Moreover, the statistics obtained from our work are also compared with other works, finding that we provide a good scheme for hiding information.

KEYWORDS

Steganographic scheme, Information encryption, Substitution of the least significant bits.

1. INTRODUCTION

Communication is essential for daily life; unfortunately, there are problems on transmitting messages, given that the information can be intercepted by unwanted readers to whom it was not originally sent. For this reason, developers work in hiding the transmission and reception of information by different methods; one of them is the term "data encryption", that it was coined with the arrival of the digital era. Which consists of rendering the information illegible, and it can only be read by a specific key holder [1].

There are several techniques of information concealment, being the steganography an art that consists in the application of different techniques to hide messages into a medium [2], with the purpose that the information embedded in the cover medium cannot be possible to decrypt, and also remaining its main characteristics to reduce detection possibility. Even though there are multiple types of cover medium; in the present manuscript, we put focus on images used to insert the information to be hidden [3].

The advantage of using a digital image as a cover medium is the digital processing, because it helps on the purpose of studying steganography by replacing the least significant bit (LSB) [4],

DOI: 10.5121/ijcsit.2018.10604

given that it allows access to information and location of each pixel in the image. The numerical value contained in each pixel is responsible for varying brightness and contrast, and they are put together within a matrix to form the image. This fact is a great advantage within the steganography since you can vary the information of the pixels in a minimal way to insert the hidden message in it without causing a great impact on the image information.

Steganography is a fairly noble weapon when it comes to the concealment of messages since it has several attributes that allow the medium to maintain its quality and have a distortion not perceptible by the human senses, which are the following:

• Confidentiality is an attribute of steganography which ensures that only authorized persons, with the secret key, can access the information embedded in the cover medium.

• Integrity, the attribute that provides the information of when the transmitted message has been modified.

• Non-repudiation is the attribute that allows verifying when a sender or receiver denies the transmitted information.

• Authentication allows confirming sender and receiver identity inside communication processes.
Figure 1 shows the methodology in the development of steganography [5], the proposed method basically follows it, but with different techniques of encryption and reading of the data. In almost every method, each author has his/her own method to provide security and confidentiality in data transmission.

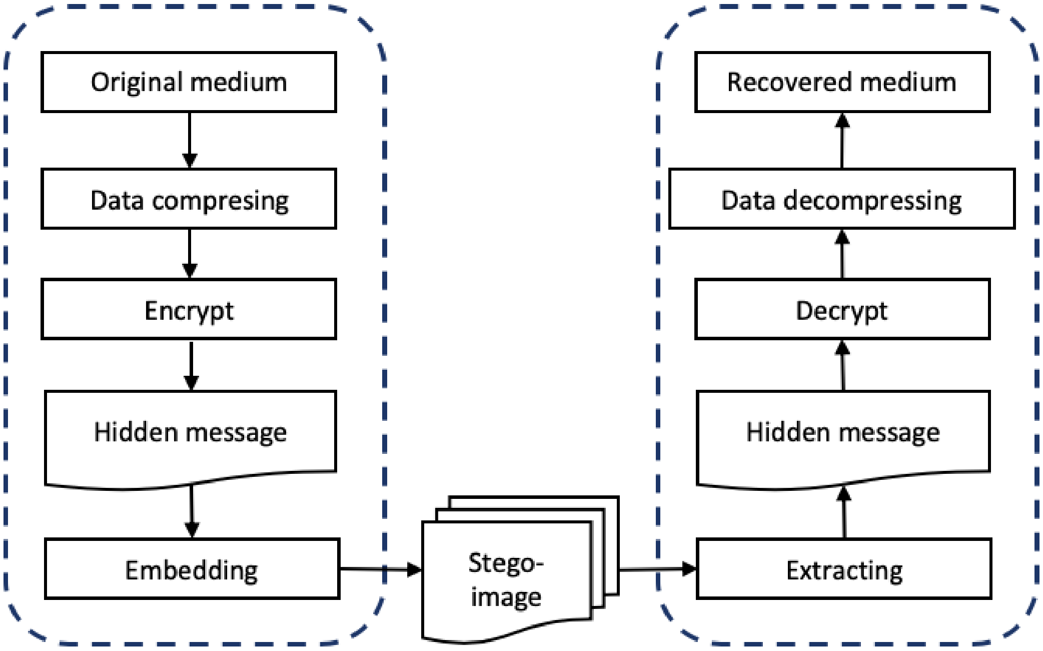

Figure 1. General model for steganographic scheme.

According to some authors, the term stegoimage refers to the image that already has an embedded information in it, and it can be considered as the output of a coding function; where such function takes on the original image, the data and a secret encryption key. The decoding function aims to reconstruct the hidden data after processing the stegoimage and secret decoding key [6].

To insert the information within a cover medium, it is necessary to statistically generate digital signals that provide high security and performance [7]; whereby, we use the Bernoulli map. This map works perfectly with digital implementation, and it is defined as an iterated map of a piece-wise linear function (PWL) and using the appropriate values for the variables, the Bernoulli map behaves alike a chaotic dynamic system [8], and its mathematical representation is expressed in Eq. 1. Besides our current implementation, in the future we do not discard the possibility to implement a digital system that hides information into "innocent" packages, so developing ideas with a map easy to implement in digital systems is a milestone for our research.

$$x_{n+1} = \begin{cases} 2\mu x_n, & 0 \leq \ x_n < 0.5 \\ 2\mu x_n - 1, & 0.5 \leq \ x_n < 1 \end{cases} \qquad \text{Eq. 1}$$

where:

x_n is the result of the iteration in the current time
x_{n+1} result of the iteration in future time.
μ represents the feedback factor and is in the interval [0,1].

Each pixel in the cover image is in a binary coding, so it is necessary to modify the domain of the Bernoulli map, to be able to work only with integer values in [0, 255] range. To make the adjustment, Eq. 2 is applied to the map output,

$$Y = floor(X * 2^{bits}) \qquad \text{Eq. 2}$$

where:

X is the resulting output of the iterations of Eq. 1.
Y is the set of well-ranged integer valuesobtained from *X*.
floor is a function that truncates decimals to only get integer-part of a real number.

2. METHODS

Traditionally, steganographic methods based on the least-significant bit substitution are limited to replace a number of leastsignificant bits in pixels [9]. The number is determined by the information that is pretended to send in a hidden manner. In other proposals, the author introduces another padlock when encrypting the hidden information, thereby increasing security by preventing possible threats. Another advantage of the last method relies on the fact that the encrypted information usually has a shape similar to the apparent randomness of the last bits of an image. Causing the statistics associated with the image not to be seriously affected by the information substitution, and inherently increasing the degree of imperceptibility.

Even with the advantages of using the data encryption, its apparently low level of impact on the cover image, it is not possible to ensure because of the lack of a study prior to the substitution. In the present paper, we propose a methodology in which the cover image is firstly analyzed, and then the least significant bits in the image are replaced by the closest encrypted frame.

2.1 Presentation of The Proposed Methodology

The proposed method begins by the cover image reading after that such image is processed to extract the two least significant bits in each pixel; they are analyzed, obtaining the statistical analysis of their occurrence. After the analysis, we proceed to obtain the to-be-hidden message. It is important to note that the maximum information that can be embedded with the proposed methodology depends on the cover image size, and such limit is the image size divided by four.

The next step is to encrypt the message. For such activity, a sequence generated by Bernoulli map limited to 32 bits length. It is taken from scaling the double-precision numbers to 32-bit integers. Because of employing the double precision sequence, the presence of the dynamic system degradation is greatly reduced from being represented by finite systems, or in this case integer-only representation.

The double-precision complete sequence must consist of the number of dibits in the to-be-hidden information. In other words, if you want to hide one kilobyte of information, you will need a sequence of four thousand elements. The 32-bit scaled sequence is divided into dibits, leaving a total of sixteen sequences, each of the two bits long.

Despite the statistical similarity between them, the sixteen sequences are not equal, and they do not have the same occurrence. Moreover, they deliver encrypted messages with certain characteristics, and we need to choose the more similar to the last two bits in the cover image. Therefore, the next step is to use the sixteen sequences to encrypt the to-be-hidden message, giving a total of sixteen encrypted messages.

Every encrypted message is analyzed to know their values occurrence, and later to compare their analysis with the analysis of the last two bits of the cover image. On next, the encrypted message whose occurrence is more similar to cover image’s one is selected. Finally, the selected message is embedded in the corresponding cover image pixel. The complete scheme of the aforementioned is shown in Figure 2.

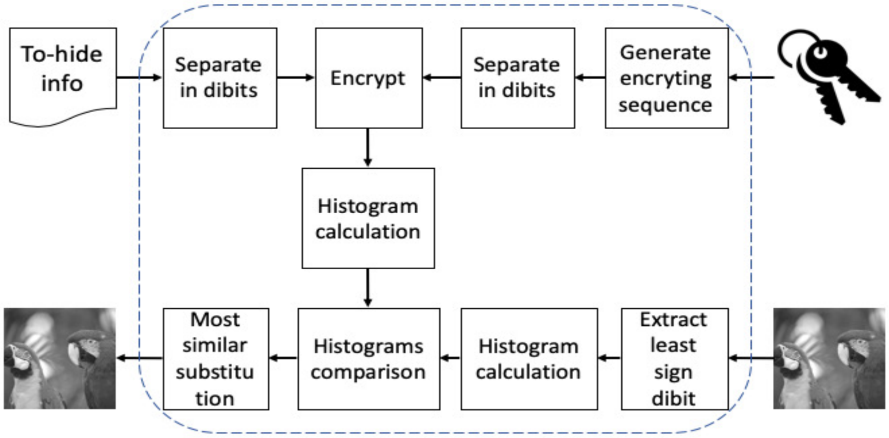

Figure 2. Proposed steganographic scheme

3. Implementation of the Proposal Using Matlab

The implementation of the steganographic scheme was done using Matlab, where the first step is the use of the command *imread*, that allows to acquire any type of image and generate an array where each of the pixels is represented in color layer in a 3D matrix of 8-bit integers; nonetheless, in the case of grayscale images, the representation of the image lies on a 1D matrix. For the present proposal, the image used as the covering is the grayscale Macaw.

The next step is to read the information file, which in this case is a one-thousand-character Lorem-ipsum file. For such, we have the set of functions integrated by *fopen*, *fread* and *fclose*. The first one allows to call a file and reserve it for being used by Matlab. *fread* captures file content that is currently reserved, and its content is assigned to a variable for later handling. And *fclose* function release the resource back to the operating system so that it can be used by any program.

After having the information in a variable, the double-precision sequence that will serve to generate the cipher sequences is obtained. It is important to notice that feedback factor (u) must be into [0.75, 1] range to exhibit a pseudorandom behavior [10]; meanwhile, the initial value is not so important as it is in [0,1] range. This sequence must be four times the length of the information to be hidden, so this value is taken from a function named length and multiplied by four. The calculated value will be used to limit the cycle of the double-precision sequence generation. The road followed until this point is described in Pseudocode 1.

Pseudocode 1. Carry image and to-hide message acquisition, as well as primary ciphering sequence generation.

```
get carry image from file (im)
open to-hide message
read to-hide message file
store message file in (mg) variable
close to-hide message file
initialize (u, lm, aux, aux1, b, c) double-precision variable
lms<= calculate mg length
initialize (s1) vector of double-precision withlms length
```

```
s1[1]<= 0.8
initialize a for cycle from 2 to lms with a variable
    aux <= divide s1[a-1] by 2
    s1[a]<= extract decimal part from aux
end for cycle
```

The primary floating-point sequence is scaled and limited truncate operation, generating a sequence that only allows integers in the range of zero to 2^{32}. Once the sequence is scaled, it is chunked in dibits. For such activity, the two least significant bits are taken by the mod function with 2; then the sequence divided by 2 and the decimal part is eliminated using the floor function. The process is repeated sixteen times to cover the complete 32-bit sequence.

After having obtained the sixteen two-bit sequences, the proposed algorithm orders the information to be encrypted, so that it coincides in word width with the two-bit ciphering sequences. To do so, the whole message is taken and separated by characters, placing them in rows in one array; after that, the characters are divided in dibits, and each one is ordered in columns of the same row. The last step is to vectorize the matrix, letting the last dibit of the first characters next to the first dibit of the second character, and so on. The complete scheme is presented in Figure 3.

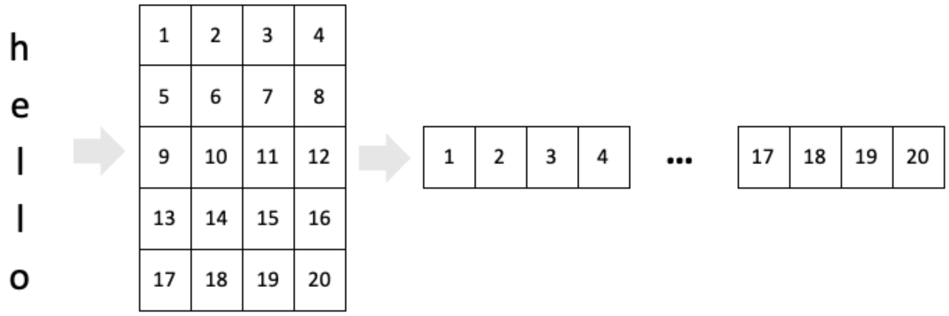

Figure 3. Vectorization scheme of the container matrix with the to-hide message characters

To finish the second section of the proposed algorithm, the two-bit ciphering sequences are used to encrypt the segmented to-hide information, resulting in a total of sixteen encrypted frames. After the encryption, the frames are analyzed to obtain their histograms, using a range of zero to three. The whole process can be seen in Pseudocode 2.

Pseudocode 2. Scaling and division of the primary encrypting sequence and to-hide message processing

```
s1<=s1 multiplied by (2 powered by 32)
s1<=truncate s1
initialize (s2) vector of double-precision with lms length
initialize a for cycle from 16 to 1 with decrements of a variable
    aux<= divide s1 by 4
    s2[a,:]<= extract decimal part from aux
    s1<= truncate aux
end for cycle

initialize (mg1) as a matrix of double-precision with lms length and 16 width
initialize a for cycle from 1 to 4 with increments of a variable
    aux<= divide mg by 4
    mg1[a,:]<= extract decimal part from aux
    s1<= truncate aux
end for cycle

mg1 <= vectorize mg1
mg1 <= transpose mg1

initialize (c1) as a matrix of double-precision with lms length and 16 positions
initialize a for cycle from 1 to 16 with increments of a variable
    c1[a,:]<= mg1 xor-bitwise s2[a,:]
end for cycle
initialize (range) vector of double-precision with 3 positions
b <= calculate length of mg1

initialize (bc) as a matrix of double-precision with lms length and 16 positions
initialize a for cycle from 1 to 16 with increments of a variable
    aux <= calculate histogram of c1[a,:] according to range
    bc[a,:]<= aux divided by b
end for cycle
```

With the second part of the algorithm explained, it is time to work with the cover image. Firstly, it is necessary to extract the least significant two bits, by using the modulus operation between the pixel information and the number two is possible to extract and store them in a vector for their further analysis; which consists of finding their occurrence. The occurrence for the last two bits in cover image is compared with those obtained from the encrypted frames of information. The comparison method is the mean square error calculation between the occurrence in the cover image and each of the sixteen encrypted frames, calculating sixteen mean square errors. According to our proposal, the minimum mean square error represents to the most similar pair image - encrypted frame. The last two bits of the image are replaced with the most similar two-bit encrypted frame. The last part of the proposed steganographic mechanism is presented in the Pseudocode 3.

Pseudocode 3. Extraction and evaluation of the histograms, and the most similar frame embedding

```
aux <= divide im by 4
im1<= extract decimal part from aux
aux<= im1
im1<= truncate aux
aux <= 0
initialize a for cycle from 1 to 16 with increments of a variable
    aux <= 0
    initialize a for cycle from 1 to 16 with increments of b variable
            aux1 <= subtract bim1[b] from bc[a,b]
            aux1 <= absolute value of aux1
            aux1 <= aux1 powered by 2
            aux <= aux added to aux1
    end for cycle
ec[a] <= aux divided by 4
end for cycle
c <= 1
im2 <= im

initialize a for cycle from 1 to 37 with increments of 4 in a variable
    initialize a for cycle from 1 to 13 with increments of 4 in b variable
            aux<= divide im2[a,b] by 4
            im2[a,b] <= truncate aux
            im2[a,b] <= im2[a,b] mulpilied by 4
            im2[a,b] <= im2[a,b] + c1[c]
            c <= increment c
    end for cycle
end for cycl
```

4. RESULTS OBTAINED FROM THE IMPLEMENTATION

As it was mentioned, the cover image used to evaluate the proposal is known as Macaw, and it was modified in grayscale. The grayscale image was created from a color one using the rgb2gray function, available in Matlab; such conversion was done previous to its usage as the carry image, but it does not form part of our proposal. Figure 4 shows the original color image and the gray-scaled version.

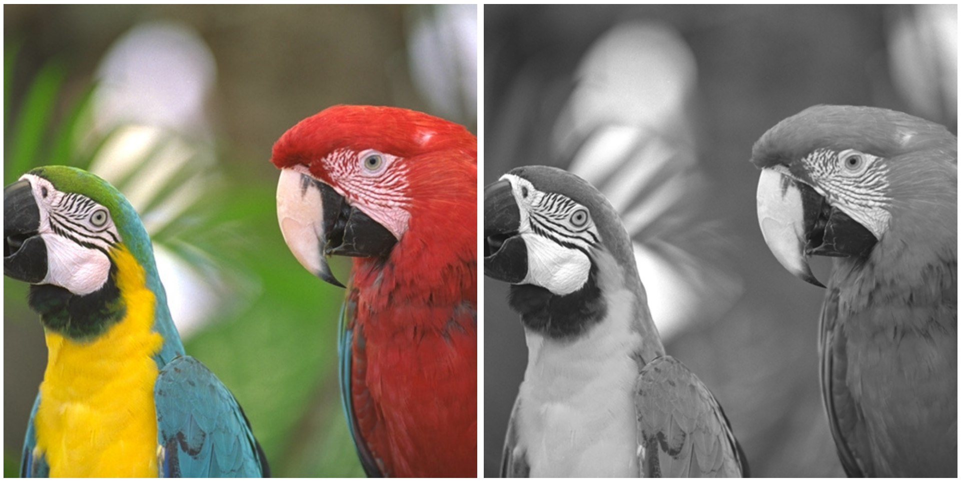

Figure 4. Macaw image in its color (left) and gray-scaled (right) version.

After obtaining the cover image, the primary encrypting sequence was generated. It is in [0, 1] range, and thenit is scaled to [0,232] range, allowing only-integer numbers. The 32-bit sequence is divided, generating sixteen two-bit frames, which were analyzed for obtaining their histograms. The results are presented in Figure 5.

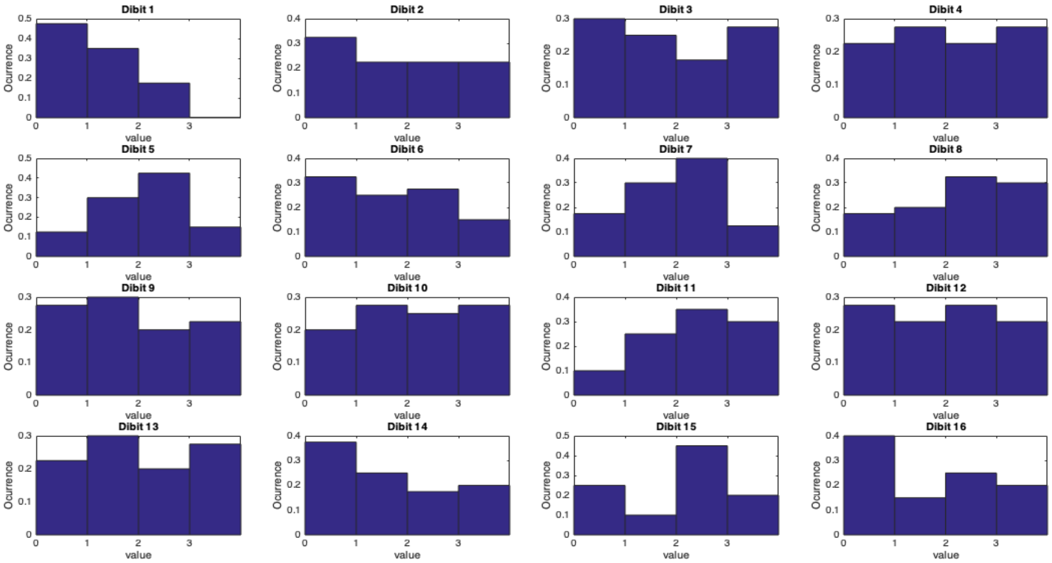

Figure 5. Histograms corresponding to the division in sixteen of the sequence scaled to 32 bits.

After the histograms corresponding to the encrypting sixteen frames, each of them is used to encrypt the to-hide message; generating a total of sixteen encrypted messages. Then we proceed to extract the last two bits of the cover image for analyzing and comparing it with the encrypted frames. To make the a fairly comparison, we proceed to calculate the histogram of the last two bits in the cover image and determine the similarities by the mean square error between it and each of the encrypted frames. The use of the mean square error is necessary, because all histograms look very similar, as is seen in Figure 6.

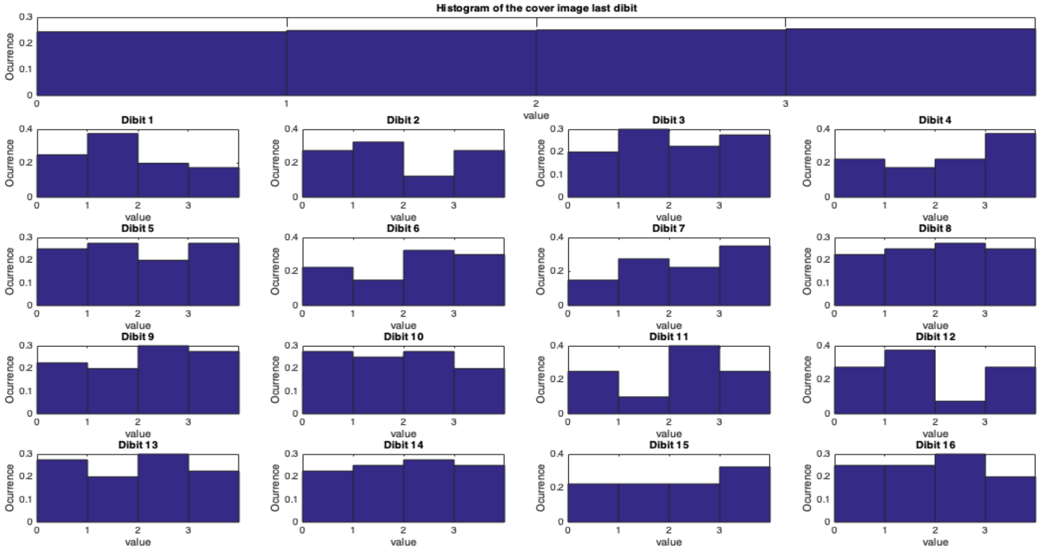

Figure 6. Histograms of the last two bits of the cover image and the encrypted frames.

When the most similar encrypted frameis determined, it is embedded in the last two bits of the cover image. However, in order to ratify our proposal, we perform an additional comparison.We proceed to repeat the embedding with two different encrypted frames, the most and the least similar ones. Table 1 shows the mean square error (MSE), as well as the peak signal to noise ratio (PSNR) between the aunatural cover image and the images with the embedded information.

Table 1. Comparison between the metrics of the images embedded with the most and the least similar encrypted frames

	least similar	most similar
PSNR	67.9913	68.4405
MSE	0.0103	0.0093

5. Discussion

In order to compare the proposed scheme, some related works were used to provide a reference, they are detailed on next. Miri and Faez present a work that is a novel approach for the data concealment in the frequency domain with the use of a genetic algorithm. In the beginning, the carrier image is mapped to a domain of the appropriate frequency using the concepts of adaptive wavelet transform and genetic algorithms. The information in the proper space is encrypted to be later embedded in the frequency coefficients that represent the edges of the image in the space domain; in such a way that the cover image is minimally changed, and it has the maximum compatibility with the human vision system [11].

On the other hand, Kumar and Kumar present a work where their technique is based on the Discrete Wavelet Transformation, together with the union of two concepts: Secret key computing, which will make the method more robust and resistant to steganalysis. And the concept is known as In-Blocks, whose purpose is to ensure the least variation in the cover image [12].

And the last comparison work is the one presented by [13]. They present a work based on three-dimensional chaotic mapping Cat and discrete wavelet transforms. In their work, they use the irregular outputs of the Cat mapping to embed the secret message in the cover image, while the discrete Wavelet transforms are used to provide robustness to the scheme.

Finally, with the established reference framework, the comparison between the three works and our proposal is presented in Table 2.

Table 2. Comparison between the current work and the reference framework

Scheme	Carry image	To-hide message	PNSR
Miri y Faez	multiples, and they present mean values as results	6.3 Kbno-specified type message	64.76 dB
Kumar y Kumar	Lena	Cameraman image, no-specified length, or any other characteristic.	44.84 dB
Ghebleh y Kanso	Lena	39 Kbrandom message	61.128 dB
Our proposal	Baboon	8 Kb, Lorem-ipsummesage	68.44 dB

According to the previous comparison, the proposed work has good features, and if we used the results obtained in Table 1 for the least similar encrypted frame. The fact is maybe because of dividing the encrypting sequence in two-bit frames has a positive influence, causing smaller distortions in the carrier image; however, there is not enough evidence to affirm it. On the other hand, the obtained results from our proposal have an opportunity to be compared again other similar methods, and also to continue in our pursuit to develop a digital system able to transmit data in a hidden from using steganographic schemes.

6. CONCLUSIONS

With the present work, it is demonstrated that in the least significant bit substitution algorithms is possible to enhance its performance, if an analysis of the cover image is previously made to the substitution. To refine our proposal some tests are needed to be applied to other carrier images, other types of message.

As it was presented in the discussion, the effect of dividing the coding frame must be characterized more deeply and finding a relationship that is easily exploited in the future. Since not only can it contribute a reduction in the impact on the carrier image, but the robustness of the scheme is favored, by presenting a very complicated plot to determine its origin.

7. ACKNOWLEDGMENTS

Tecnologico Nacional de Mexico is thanked for the support provided with the funded project "Generación de un esquemaesteganográficocaótico de bajo impacto para la imagen portadora" with reference number 6622.18P.

REFERENCES

[1] Brandao, A. S., & Jorge, D. C. (2016). Artificial neural networks applied to image steganography. IEEE Latin America Transactions, 14(3), 1361-1366.

[2] Fridrich, J., Goljan, M., & Du, R. (2001). Detecting LSB steganography in color, and gray-scale images. IEEE multimedia, 8(4), 22-28.

[3] Isaza, G. A., Espinosa, C. A., & Ocampo, S. M. (2018) Análisis de técnicasesteganograficas y estegoanalisisencanalesencubiertos, imágenes y archivos de sonido.

[4] Divya, S. S., & Reddy, M. R. M. (2012). Hiding text in audio using multiple LSB steganography and provide security using cryptography. International journal of scientific & technology research, 1(6), 68-70.

[5] Jamal A. Othman.(2014). Steganographic scheme to avoid statistical Steganalysis. JOURNAL OF THE COLLEGE OF EDUCATION FOR WOMEN, 25(1), 249-256.

[6] Wu, H. Z., Wang, H. X., & Shi, Y. Q. (2016). Can Machine Learn Steganography?-Implementing LSB Substitution and Matrix Coding Steganography with Feed-Forward Neural Networks. arXiv preprint arXiv:1606.05294.

[7] Anees, A., Siddiqui, A. M., Ahmed, J., & Hussain, I. (2014). A technique for digital steganography using chaotic maps. Nonlinear Dynamics, 75(4), 807-816.

[8] Li, S., Chen, G., &Mou, X. (2005). On the dynamical degradation of digital piecewise linear chaotic maps. *International journal of Bifurcation and Chaos*, *15*(10), 3119-3151.

[9] Zhang, T., & Ping, X. (2003). A new approach to reliable detection of LSB steganography in natural images. *Signal processing*, *83*(10), 2085-2093.

[10] Al-Fadhel, T. A. (2007). Gauss map vs Bernoulli shift. Applied Mathematics and Computation, 194(2), 520-526.

[11] Miri, A., &Faez, K. (2017). Adaptive image steganography based on transform domain via genetic algorithm. Optik-International Journal for Light and Electron Optics, 145, 158-168.

[12] Kumar, V., & Kumar, D. (2017). A modified DWT-based image steganography technique. Multimedia Tools and Applications, 1-30.

[13] Ghebleh, M., &Kanso, Ax. (2014). A robust chaotic algorithm for digital image steganography. Communications in Nonlinear Science and Numerical Simulation, 19(6), 1898-1907.