MANSOURA JOURNAL OF Mathematics
Official Journal of Faculty of Science, Mansoura University, Egypt
E-mail: scimag@mans.edu.eg ISSN:2974-4946

A Modified LSB Technique Utilizing a Two-DimensionalChaotic Map for Image Steganography

Mahmoud Abdelhakm, Ahmed Salah, and A. A. Karawia

Mathematics Department, Faculty of Science Mansoura University, Mansoura 35516, Egypt
abibka@mans.edu.eg

Received:30/12/2023
Accepted: 6/2/2024

Abstract: This study addresses the concealment and protection of secret images through both image cryptographic and steganographic techniques. The proposed image cryptographic algorithm employs a logistic map for bit shuffling, followed by encryption using a two-dimensional piecewise chaotic map. Concurrently, an image steganographic algorithm is introduced, utilizing a two-dimensional economic map for pixel rearrangement and a modified least significant bit technique to embed the encrypted, rearranged secret image within a cover image. The cryptographic algorithm ensures the encryption of the secret image, while the steganographic algorithm conceals the encrypted image within the cover image. Experimental results confirm the security and robustness of the proposed algorithm, showcasing its efficacy in withstanding various attacks. The main contributions of this paper lie in the clarity of the proposed methods for image encryption and steganography, and the obtained results underscore the algorithm's security and performance relative to other published methods.

keywords: Encryption, Decryption, Chaos, LSB substitution, Steganography, Security analyses.

1.Introduction

The importance of secure communication in networked systems for sharing information is widely recognized in today's world. Various types of information are shared, including videos, audios, images, texts, and more. This paper focuses specifically on images, which play a crucial role in sensitive sectors such as the military and medicine. Consequently, safeguarding the transmission of these images is of utmost importance. To ensure their protection, the most commonly employed methods are watermarking, cryptography, and steganography [1]. A digital watermark is a hidden marker implanted within a signal that can tolerate noise, including audio, video, or image data. Its primary purpose is to establish the copyright ownership of the signal [2]. Cryptography is the art of encrypting information, making it unreadable, using encryption techniques. The author [3] initially published the encryption algorithm that relies on a chaotic map. Numerous authors have employed chaotic maps to create encryption algorithms for images following this development. Among these algorithms, a subset relied solely on permutation [4] some other algorithms, on the other hand, relied solely on substitution as seen in [5]. Furthermore, many authors have incorporated various types of chaotic maps to blend permutation and substitution in their image encryption algorithms [6, 7, 8]. Steganography algorithms are utilized for concealing confidential information inside the cover image. Subsequently, the cover image is transmitted to the recipient. Ultimately, the concealed data can be extracted from the cover image. Images [9], audio [10], text [11], video [12], compressed files [13], and other forms are frequently utilized as information carriers. Image steganography holds great importance in the field of information security due to the strong intuitive appeal and comprehensive characteristics of images.

Steganography in digital media has been applied since the 1990s. The earliest image steganography algorithm was suggested by Bender et al. [14], which involved altering the least significant bit (LSB) of pixels to embed secret data. Similarly, in the compression domain, algorithms like Jsteg [15] and F5 [16] hide information in the LSBs of quantized

discrete cosine transform (DCT) coefficients. There are two different approaches to conceal secret images using the cover image: spatial domain and frequency domain [17]. The cover image's intensity values are utilized in the spatial domain to conceal the secret information [18, 19]. On the other hand, the secret image pixels are concealed using the transformation coefficients of the pixels in the cover image within the frequency domain [20, 21]. Various suggested techniques exist for spatial domain steganography, including watermarking [22], LSB substitution [23, 24], modulus function [25], pixel value differencing technique [26], LSB matching [27], optimal pixel similarity [28], discrete wavelets technique [29], and deep learning [30]. LSB substitution-based hiding stands out as a straightforward and speedy technique within these methods, providing effective security. We employ the LSB substitution technique in our suggested algorithm to conceal the secret image [31]. The bits from the secret image are used to replace the LSBs of the cover image pixels in the LSB-based embedding algorithm. This process can be carried out sequentially or randomly. Randomly selecting pixels from the cover image for concealment provides the best security compared to the sequential approach [32]. The cover image pixels in our method are selected randomly through a chaotic sequence produced by the chaotic map.

Chaos denotes disorder, while in mathematics, a map represents a function showcasing chaotic behavior [33]. A map is also known as a discrete time dynamical system. Chaotic maps possess certain inherent characteristics [13], including: 1) initial conditions' sensitivity; 2) ergodicity; 3) determinism; 4) structural complexity. Various schemes for information hiding using chaotic sequences have been proposed [34]. In our proposed algorithm, we utilize a one-dimensional logistic map and two-dimensional chaotic maps, piecewise smooth nonlinear and mixed Bertrand duopoly chaotic maps to produce random chaotic sequences. To enhance the security of our proposed algorithm, we will encrypt the secret image using the permutation-substitution model. The algorithm proposal consists of three primary stages. At first, the secret image undergoes encryption utilizing a two-dimensional piecewise smooth nonlinear chaotic map. Next, the pixel locations of the cover image are randomly determined using a two-dimensional mixed Bertrand duopoly chaotic map. At the end, the encrypted image bits are inserted into randomly chosen pixels from the cover image's color channel. Our contributions are as follows:

i- The clarity of the proposed methods for image encryption and steganography, and the obtained results underscore the algorithm's security and performance relative to other published methods.

ii- None of the bits from the secret image are sacrificed or omitted.

The subsequent sections of the paper are structured in the following manner: Section 2 provides a concise explanation of the suggested chaotic maps. Section 3 presents the encryption, embedding, extracting algorithms. Section 4 discusses the experimental results and analysis. Finally, Section 5 declares the conclusions of the current paper.

2. The suggested maps

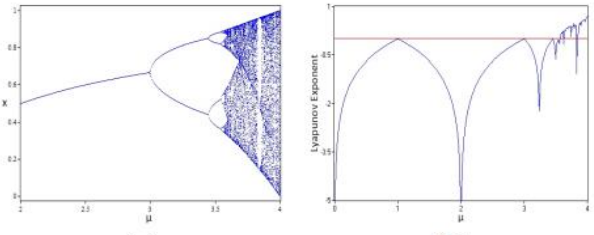
In this section, three different chaotic maps are presented. The first is the well-known chaotic map, the logistic map, which is represented as follows:

$$x_{t+1} = \mu x_t(1 - x_t),\ t = 0,1,2,\dots, x_t \in [0,1] \quad (1)$$

where $\mu \in (0,4)$ is the control parameter. The logistic map behaves chaotically when μ lies between 3.6 and 4.0. Figure 1 displays the bifurcation diagram and the Lyapunov exponent.

2.1 Piecewise smooth nonlinear chaotic map(PSNCM)

The current map has been proposed in [35]. It is represented as follows:



(a) (b)

Figure 1: (a) Bifurcation diagram of the logistic map and (b) Lyapunov exponent for the logistic map.

$$\left.\begin{aligned} w_{1,i+1} &= w_{1,i} + k_1 w_{1,i}[1 - 2(1+c_1)w_{1,i} - \theta w_{2,i}], \\ w_{2,i+1} &= \begin{cases} w_{2,i} + k_2 w_{2,i}[\theta(1 - w_{1,i} - 2w_{2,i}) - 2c_2 w_{2,i}] \\ \quad if \quad w_{1,i} \geq f(w_{2,i}), \\ w_{1,i} \quad if \quad w_{1,i} < f(w_{2,i}), \end{cases} \end{aligned}\right\} \quad (2)$$

where

$$f(w_{2,i}) = w_{2,i}[1 + \theta k_2 - 2k_2(c_2 + \theta)w_{2,i}]/(1 + \theta k_2 w_{2,i}).$$

The assumption made by the authors is that there are two firms in the market, and they offer different products. Each firm gives the market a non-negative real number w_i, which indicates the quantity of production the firm provides. The two parameters c_1 and c_2 represent the shift cost such that $c_1 > c_2$. The parameter θ refers to a fraction that customers can pay to purchase newly made goods such that $c_2 < \theta$ and $0 < \theta < 1$. The $k_1 q_1$ and $k_2 q_2$ functions are used to measure how quickly a company adjusts its output in response to changes in its marginal profit. The parameters: $c_1 = 0.55; c_2 = 0.30; \theta = 0.35; k_1 = 2.95; k_2 = 2.00$ and initial values $w_{1,0} = 0.0002; w_{2,0} = 0.0008$ show the chaotic behaviour of **PSNCM**. Figure 2 displays the bifurcation diagram and the Lyapunov exponent of **PSNCM** concerning k_1.

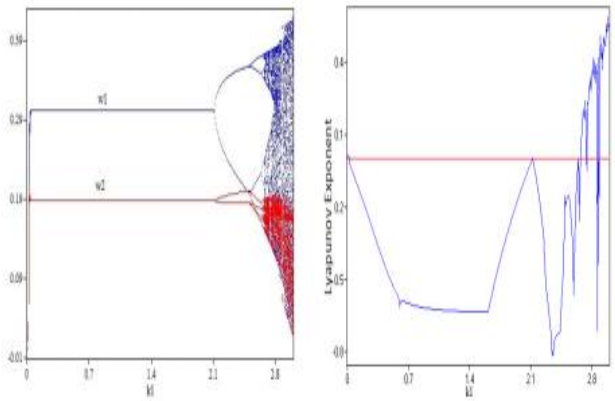


Figure 2: (a) Bifurcation diagram of the logistic map and (b) Lyapunov exponent for **PSNCM**.

2.2. Mixed Bertrand duopoly chaotic map (MBDCM)

Important security-related features are present in this map [36]. It exhibits a strong chaotic behaviour and a wide range of bifurcation parameters. It is defined as follows:

$$\left.\begin{aligned} p_1(t+1) &= p_1(t) + \frac{\alpha_1 p_1(t)}{1-d^2}[a(1-d) - (2-s_1)p_1(t) \\ &\quad + dp_2(t) - (1 + dm_2 - d)s_1 + m_1], \\ p_2(t+1) &= p_2(t) + \frac{\alpha_2 p_2(t)}{1-d^2}[a(1-d) - (2-s_2)p_2(t) \\ &\quad + dp_1(t) - (1 + dm_1 - d)s_2 + m_2] \end{aligned}\right\} \quad (3)$$

where $p_i = 0$, $i = 1,2$ denote the price of firm i's product, $-1 < d < 1$ denotes the degree of horizontal product differentiation, $a > 0$ denotes the size of the market demand, $m_i \geq 0$ represents the marginal cost, the speed adjustment of firm i is denoted by parameter $\alpha_i > 0$, while parameter $s_i \in (0,1)$ represents the extent of public ownership. The parameters: $a = 3$; $d = 0.1$; $s_2 = 0.4$; $m_1 = 1$; $m_2 = 2$; $\alpha_1 = 0.72$; $\alpha_2 = 0.4$; $s_1 \in (0,1)$and initial values $p_1(0) = 0.001; p_2(0) = 0.002$ show the chaotic behaviour of **MBDCM** with respect to s_1. Figure 3 displays the bifurcation diagram and the Lyapunov exponent of **MBDCM** concerning s_1.

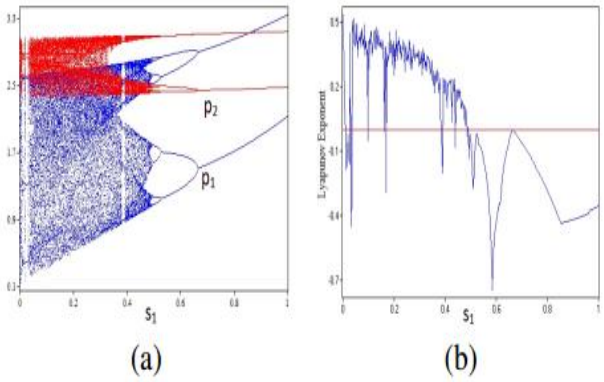


Figure 3: (a) Bifurcation diagram of the **MBDCM** and (b) Lyapunov exponent for the **MBDCM** with respect to the parameter s_1.

3. Proposed Algorithm

The proposed algorithm consists of three stages: encryption, embedding, and extraction.

3.1. Encryption Algorithm

In this stage, the secret image is encrypted using the chaotic map (**PSNCM**). Initially, the secret image's pixels undergo shuffling through the logistic map, which is known as the confusion step. Following that, the resulting shuffled matrix is diffused utilizing the chaotic map known as (**PSNCM**). The encryption algorithm may be described as follows:

Algorithm 1 Encryption Algorithm

Input: The secret image A of size $M \times N$, the initial value x_0 for the logistic map, the initial values $w_{1,0}$ and $w_{2,0}$ for **PSNCM**, the parameter μ for the logistic map, and the parameters k_1, k_2, c_1, c_2 and θ for **PSNCM**.
Output: The encrypted image E of size $M \times N$.
Step 1: Read the secret image A.
Step 2: Shuffle A using the logistic map (1), say $\mathbf{Sh_A}$.
Step 3: Reshape $\mathbf{Sh_A}$ to the size $1 \times MN$.
Step 4: Generate a sequence of random values using **PSNCM**,
say $W = [w_1, w_2, \cdots, w_{MN}]$.
Step 5: Preprocessing:
For $j = 1{:}\,MN$, compute
$w_j = mod\left(floor\left(w_j \times 10^{14}, 256\right)\right)$end
Step 6: Compute the bit-wise xor between $\mathbf{Sh_A}$ and W, say E $= bitxor(\mathbf{Sh_A}, \mathrm{W})$.
Step 7: Transform E into an $M \times N$ array.

3.2. Embedding Algorithm

The proposed embedding algorithm requires the selection of double pixels from the cover image to conceal a pixel from the encrypted medical image. **MBDCM** assists in the random selection of these pixels. To modify LSB embedding, the pixels in the encrypted image undergo a random shuffling of their 8 bits, as demonstrated in Figure 4. Let's say P_1 and P_2 denote two random pixels from the cover image, while the arrows indicate the updated locations of the bits in the encrypted image's pixel [37]. In [38], he researchers discovered that the red (R) and green (G) channels of an image are more susceptible to distortion caused by embedded bits compared to the blue (B) channel. Consequently, P_1 and P_2 each hide a pair of bits within their B channels. This algorithm may be described as in **Algorithm** 2.

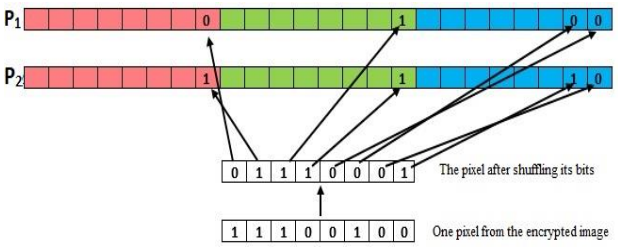


Figure 4: Conceal a single pixel of the encrypted image within double pixels of the cover image.

Algorithm 2 Embedding algorithm

Input: The secret image A of size $M \times N$, the cover image C of size $M_1 \times N_1 \times 3$, the initial value x_0 for the logistic map, the initial values $w_{1,0}$ and $w_{2,0}$ for **PSNCM**, the initial values $p_1(0)$ and $p_2(0)$ for **MBDCM**, the parameter μ for the logistic map, the parameters k_1, k_2, c_1, c_2 and θ for **PSNCM** and the parameters a, d, s_1, s_2, m_1, m_2, α_1, and α_2 for **MBDCM**.
Output: The stego image S with size $M_1 \times N_1 \times 3$.
Step 1: Utilize **Algorithm** 1 to encrypt the secret image. The result is the matrix **E**.
Step 2: Reshape **E** to the matrix $\mathbf{E_1}$ of size $MN \times 1$ and convert it to the binary with length 8 for each intensity value.
Step 3: Reshape **S** to the matrix $\mathbf{S_1}$ of size $M_1N_1 \times 1$ and convert it to the binary with length 8 for each intensity value.
Step 4: Generate random sequence of size $M_1N_1 \times 1$ using **MBDCM**, say
$\mathbf{G} = [g_1, g_2, \cdots, g_{M_1N_1}]$,
$X = [g_1, g_2, \cdots, g_{MN}]$ and
$Y = [g_{MN+1}, g_{MN+2}, \cdots, g_{2MN}], g_i \in \{1,2,3,\ldots,M_1N_1\}$.
Step 5: Shuffle the position of every pixel in $\mathbf{E_1}$, and denote the resulting order of bits as $\mathrm{B} = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\}$.
Step 6: For $i = 1$ to MN do
C(X(i),1)=bitset(**C**(X(i),1),1,$\mathbf{E_1}$(i,b_1));
C(Y(i),1)=bitset(**C**(Y(i),1),1,$\mathbf{E_1}$(i,b_2));
C(X(i),2)=bitset(**C**(X(i),2),1,$\mathbf{E_1}$(i,b_3));
C(Y(i),2)=bitset(**C**(Y(i),2),1,$\mathbf{E_1}$(i,b_4));
C(X(i),3)=bitset(**C**(X(i),3),1,$\mathbf{E_1}$(i,b_5));
C(X(i),3)=bitset(**C**(X(i),3),2,$\mathbf{E_1}$(i,b_6));
C(Y(i),3)=bitset(**C**(Y(i),3),1,$\mathbf{E_1}$(i,b_7));
C(Y(i),3)=bitset(**C**(Y(i),3),2,$\mathbf{E_1}$(i,b_8));
End
Step 7: Reshape C to the matrix S of size $M_1 \times N_1 \times 3$.
Step 8: S is the stego image.

3.3. Extraction Algorithm

During this stage, it is possible to extract the secret image **A** from the provided stego image **S** without relying on the original cover image **C**. Using the embedding algorithm sequence, the pixels in the stego image that contain the encrypted image bits are selected. By unshuffling and extracting the LSBs of the

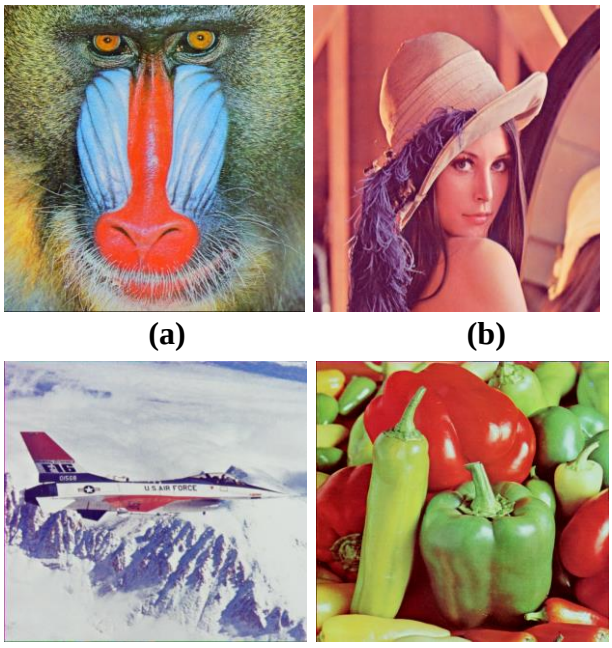
selected pixels, the encrypted image is reconstructed. Finally, the encrypted image is decrypted, and the secret image **A** is obtained using the inverse of **Algorithm** 1. The proposed extracting algorithm may be described as in **Algorithm** 3.

Algorithm 3 Extracting algorithm

Input: S(stego image) and the secret key.
Output: The secret image A of size $M \times N$.
Step 1: Reshape S to the matrix C of size $M_1N_1 \times 3$.
Step 2: Generate random sequence of size $M_1N_1 \times 1$ using **MBDCM**, say
$\mathbf{G} = [g_1, g_2, \cdots, g_{M_1N_1}]$,
$X = [g_1, g_2, \cdots, g_{MN}]$ and
$Y = [g_{MN+1}, g_{MN+2}, \cdots, g_{2MN}], g_i \in \{1,2,3,\ldots, M_1N_1\}$.
Step 3: Produce the shuffled locations of the extracting
bitsB $= \{b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8\}$.
Step 4: For $i = 1$ to MN do
$\mathbf{E_1}$(i,b_1)=bitand(C(X(i),1),1);
$\mathbf{E_1}$(i,b_2)=bitand(C(Y(i),1),1);
$\mathbf{E_1}$(i,b_3)=bitand(C(X(i),2),1);
$\mathbf{E_1}$(i,b_4)=bitand(C(Y(i),2),1);
$\mathbf{E_1}$(i,b_5)=bitand(C(X(i),3),1);
sbx=dec2bin(C(X(i),3),8);
$\mathbf{E_1}$(i,b_6)=bitand(str2double(sbx(7)),1);
$\mathbf{E_1}$(i,b_7)=bitand(C(Y(i),3),1);
sbx=dec2bin(C(Y(i),3),8);
$\mathbf{E_1}$(i,b_8)=bitand(str2double(sbx(7)),1); End
Step 5: Reshape $\mathbf{E_1}$ to the matrix E of size $M \times N$.
Step 6: The secret image, A, is obtained by decrypting **E** using the inverse of **Algorithm** 1.

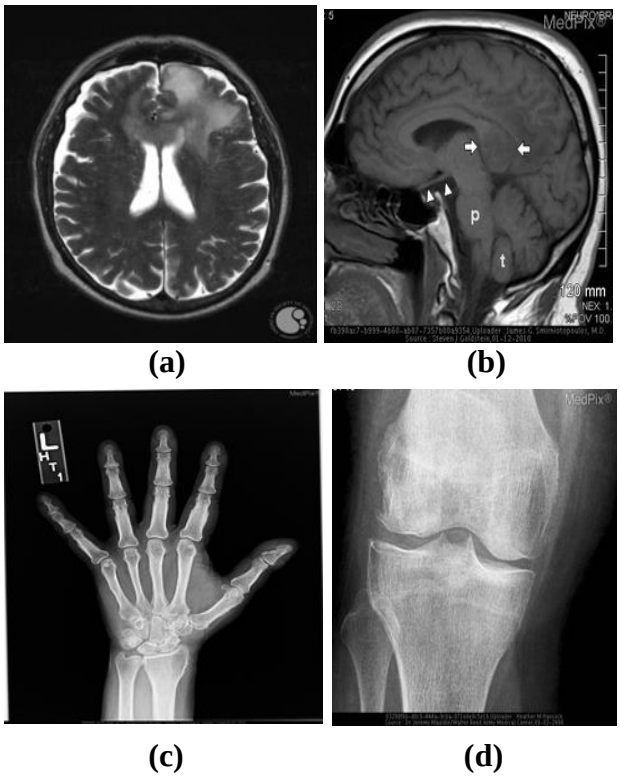
4. EXPERIMENTAL RESULT

We will now investigate the outcome of our proposed algorithm in the present section. All software applications are developed utilizing MATLAB R2023a, and the specifications of the laptop include an AMD A10-9620P RADEON R5, 2.50 GHz, along with 8 GB of RAM. All images are chosen from the USC-SIPI image database. Our proposed algorithm's stego-key consists of five initial values: $x_0 = 0.01$, $w_{1,0} = 0.0002$, $w_{2,0} = 0.0008$, $p_1(0) = 0.20$, $p_2(0) = 0.30$ and fourteen parameters: $\mu = 4$, $c_1 = 0.55$, $c_2 = 0.30$, $\theta = 0.35$, $k_1 = 2.95$, $k_2 = 2.00$, , $a = 3$, $d = 0.1$, $s_1 = 0.50$, $s_2 = 0.99$, $m_1 = 1$, $m_2 = 2$, $\alpha_1 = 0.72$ and $\alpha_2 = 0.40$. Figure 5 displays four images utilized as cover images, while Figure 6 displays the four samples of secret medical images (S_1, S_2, S_3, and S_4). Figure 7 exhibits the cover image (lena), S_1, the stego image, and the recovered S_1.



(a) **(b)** **(c)** **(d)**

Figure 5: 512 × 512 cover images: (a) Baboon, (b)Lena, (c) Airplane, and (d) Peppers



(a) **(b)** **(c)** **(d)**

Figure 6: 256× 256 secret medical images: (a) S_1, (b) S_2, (c) S_3, and (d) S_4

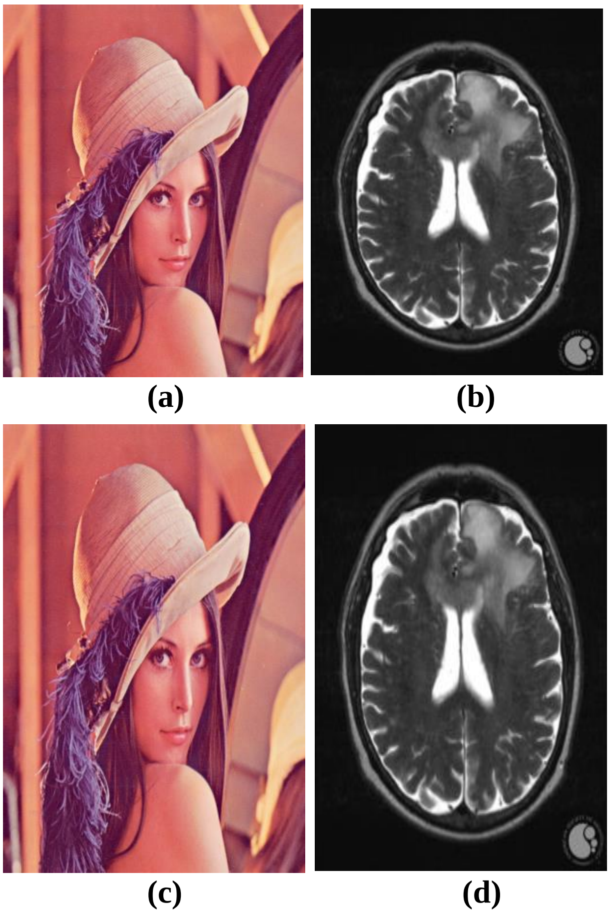


Figure 7: Result of the proposed algorithm: (a) Lena(cover image), (b) S_1, (c) Stego image, and (d) Restored S_1

4.1. Statistical analyses

arious statistical analyses were used to examine the effectiveness of the suggested algorithm.

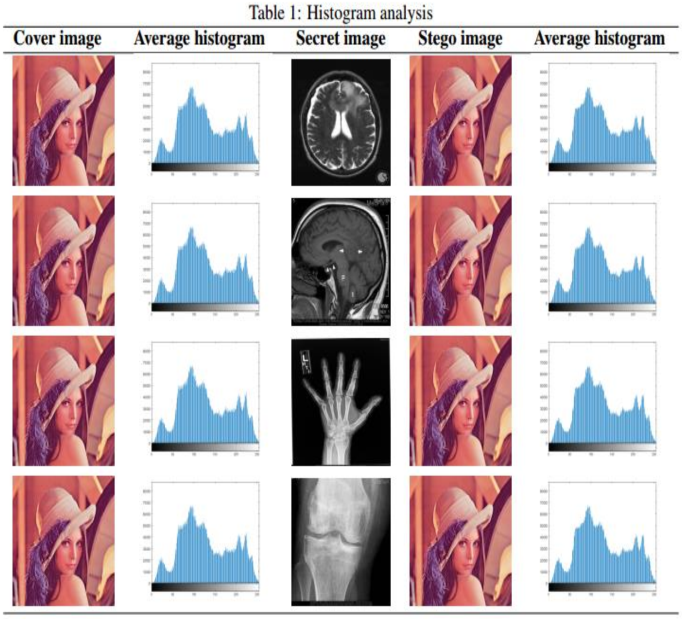
4.1.1. Histogram analysis

The histogram of an image shows a graphical representation of the pixel count for each unique intensity value present in the image. Therefore, it is regarded as a statistical method of attack. This attack allows the human eye to detect differences between the cover image and the stego image. Table 1 shows the average histograms of the cover image (Lena) and stego images, along with the four secret images. It can be observed visually that the average histograms of the stego images closely resemble those of the cover images.

4.1.2. PSNR analysis

The significance lies in how much the stego image deviates from the cover image, which is typically assessed using the peak signal to noise ratio (PSNR).

Table 1: Histogram analysis



$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right)(4)$$

where

$$MSE = \frac{1}{3MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{3}\left(C(i,j,k) - S(i,j,k)\right)^2, (5)$$

where **C** and **S** are the cover and stego images, respectively. Table 2 displays **MSE** and **PSNR**) between various cover images and their corresponding stego images, all containing the identical secret image. Moreover, Table 3 exhibits a comparative result between the current algorithm, algorithm [29], algorithm [24], and algorithm [30]. The evaluation employs lena (512 × 512) as the cover image. Considering that the **PSNR** value exceeds 50 dB, the effectiveness of the current algorithm is evident in achieving nearly perfect restoration outcomes comparable to the cover image.

4.1.3. Payload(capacity)

The embedding rate (**ER**) represents the proportion of concealed bits within all pixels of the cover image. It is represented by the equation:

$$ER = \frac{T}{M \times N \times 3}(bpp)(6)$$

where T is the number of embedding bits, and $M \times N \times 3$ represents the total number of pixels in the cover image. A higher **ER** value, as determined by the assessment of the embedding payload in steganography, signifies improved performance of the algorithm. This

means that a cover pixel can accommodate a greater number of concealed bits. The embedding rates are displayed in Table 4.

Table 2: **MSE** and **PSNR** values comparing cover images and stego images

Cover image	Secret image	MSE	PSNR(dB)
Baboon (512 x 512)	S_1	0.5850	50.4594
	S_2	0.5840	50.4666
	S_3	0.5871	50.4435
	S_4	0.5838	50.4682
Lena (512 x 512)	S_1	0.5835	50.4702
	S_2	0.5823	50.4794
	S_3	0.5837	50.4688
	S_4	0.5867	50.4464
Airplane (512 x 512)	S_1	0.5898	50.4239
	S_2	0.5884	50.4343
	S_3	0.5887	50.4322
	S_4	0.5904	50.4195
Peppers (512 x 512)	S_1	0.5995	50.3529
	S_2	0.5946	50.3882
	S_3	0.5980	50.3636
	S_4	0.5968	50.3726

Table 3: MSE and **PSNR** between proposed algorithm, algorithm [29], algorithm [24], and algorithm [30]

Algorithm	MSE	PSNR(dB)
Proposed algorithm	0.580	50.470
[29]	2.090	45.780
[24]	2.250	44.600
[30]	0.957	48.317

Table 4: Embedding payload

Cover image	Secret image	ER
Baboon (512 x 512)	S_1	0.6667
	S_2	
	S_3	
	S_4	
Lena (512 x 512)	S_1	0.6667
	S_2	
	S_3	
	S_4	
Airplane (512 x 512)	S_1	0.6667
	S_2	
	S_3	
	S_4	
Peppers (512 x 512)	S_1	0.6667
	S_2	
	S_3	
	S_4	

4.1.4. Image quality measures

The mathematical equations for measures of image quality can assess the connection between the human visual system and the display. Steganographic algorithms can utilize statistical indicators to gauge the concealed information within the stego image in relation to the cover image [37]. After inserting the secret image, the measurements calculate the similarity between the cover image and the stego image by summing all the bytes [39].

$$AD = \frac{1}{3MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{3}|\mathrm{C}(i,j,k) - \mathrm{S}(i,j,k)| \quad (7)$$

$$IF = 1 - \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{3}\mathrm{C}(i,j,k) - \mathrm{S}(i,j,k)}{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{3}\mathrm{C}^2(i,j,k)} \quad (8)$$

$$SC = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{3}\mathrm{C}^2(i,j,k)}{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{3}\mathrm{S}^2(i,j,k)} \quad (9)$$

$$CQ = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{3}\mathrm{C}(i,j,k)\mathrm{S}(i,j,k)}{\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{3}\mathrm{C}^2(i,j,k)} \quad (10)$$

where $C, S, AD, IF, SC,$ and CQ represent cover image, stego image, average absolute difference, image fidelity, structure content, and correlation quality, respectively. When $IF, SC,$ and CQ approach 1, it signifies a significant degree of similarity. Additionally,

AD gets closer to 0, the difference between the stego image and the cover image is negligible. Table 5 illustrates that IF, SC and CQ are near one, while AD is close to zero. Therefore, it can be concluded that there are no notable differences between the stego image and the cover image. In addition, Table 6 displays a comparison among the proposed algorithm, algorithm [31], and algorithm [39], illustrating the superior performance of the proposed algorithm over algorithm [39].

4.1.5. Relative entropy

In the context of image steganography, relative entropy is utilized to assess the statistical changes that occur in the cover image when embedding secret information. It is defined by the formula:

$$D(P_C||P_S) = \sum P_C log_2\left(\frac{P_C}{P_S}\right) \quad (11)$$

where D represents the relative entropy, P_C and P_S denote the probability distributions of the pixel values in the cover image and the stego image, respectively.

By evaluating the relative entropy, steganography algorithms can estimate the level of detectability or the extent to which the secret information embedded in the stego image deviates from the original cover image. Lower relative entropy values indicate that the steganographic changes are less noticeable, while higher values suggest more significant alterations in the probability distribution of

pixel intensities. Table 7 displays the relative entropy values for various cover and stego images containing secret information. According to the findings in Table 7, the relative entropies are nearly zero. In Table 8, a comparison is shown between our algorithm and the algorithms mentioned in [31] and [39]. Our algorithm demonstrates superior performance compared to the algorithm discussed in [39], and it exhibits similar performance to the algorithm described [31]. The data presented in Tables 7 and 8 enables us to derive conclusions regarding the similarity between the cover and stego images.

Table 5: Measures of image quality

Cover image	Secret image	AD	IF	SC	CQ
Baboon (512 x 512)	S_1	0.3754	1.0000	1.0000	1.0000
	S_2	0.3753	1.0000	1.0000	1.0000
	S_3	0.3764	1.0000	1.0000	1.0000
	S_4	0.3748	1.0000	1.0000	1.0000
Lena (512 x 512)	S_1	0.3747	1.0000	1.0000	1.0000
	S_2	0.3741	1.0000	1.0000	1.0000
	S_3	0.3749	1.0000	1.0000	1.0000
	S_4	0.3764	1.0000	1.0000	1.0000
Airplane (512 x 512)	S_1	0.3766	1.0000	0.9999	1.0000
	S_2	0.3763	1.0000	0.9999	1.0000
	S_3	0.3759	1.0000	0.9999	1.0000
	S_4	0.3775	1.0000	0.9999	1.0000
Peppers (512 x 512)	S_1	0.3792	1.0000	0.9999	1.0000
	S_2	0.3776	1.0000	0.9999	1.0000
	S_3	0.3789	1.0000	0.9999	1.0000
	S_4	0.3783	1.0000	0.9999	1.0000

Table 6: Comparing the image quality of our algorithm, algorithm [31] and the algorithm [39] using a baboon (512 x 512) as the cover image and a lena (256 x 256) as the hidden image.

Cover image	AD	IF	SC	CQ
Proposed algorithm	0.3750	1.0000	1.0000	1.0000
[31]	0.0067	1.0000	1.0000	1 .0000
[39]	0.0067	-5.6×10^{-5}	$--$	-8.22×10^{-6}

Table 7: Relative entropies between distinct cover and stego images.

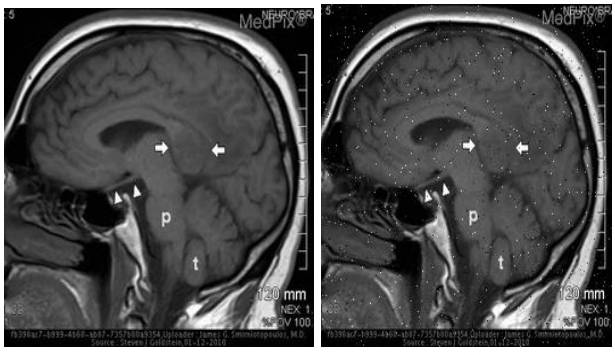
Cover image	Secret image	Red	Green	Blue	Gray
Baboon (512 x 512)	S_1	0.00035	0.00040	0.00100	0.00030
	S_2	0.00032	0.00028	0.00086	0.00026
	S_3	0.00036	0.00044	0.00100	0.00029
	S_4	0.00400	0.00047	0.00092	0.00028
Lena (512 x 512)	S_1	0.00057	0.00067	0.00098	0.00033
	S_2	0.00045	0.00062	0.00094	0.00032
	S_3	0.00049	0.00059	0.00110	0.00033
	S_4	0.00042	0.00073	0.00098	0.00033
Airplane (512 x 512)	S_1	0.00079	0.00160	0.00540	0.00100
	S_2	0.00098	0.00160	0.00540	0.00110
	S_3	0.00110	0.00160	0.00540	0.00100
	S_4	0.00082	0.00160	0.00540	0.00100
Peppers (512 x 512)	S_1	0.00042	0.01040	0.01590	0.00750
	S_2	0.00049	0.01060	0.01560	0.00760
	S_3	0.00046	0.01050	0.01550	0.00750
	S_4	0.00044	0.01060	0.01570	0.00760

Table 8: Comparing the relative entropy of our algorithm, algorithm [31] and the algorithm [40]

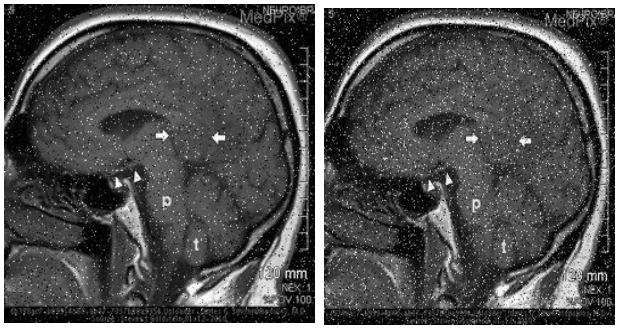
Cover image	Secret image	Proposed algorithm	Algorithm [31]	Algorithm [39]
Airplane (512×512)	Lena (256×256)	0.00030	0.00034	0.00225

4.1.6. Noise attack

In this section, our algorithm underwent testing against a noise attack. The stego image was corrupted by salt & pepper noise at densities of 0.01,0.05, and 0.1. The impact of the noise attack on the current algorithm was assessed using **PSNR**. Figure 8 shows the secret information extracted from the stego image following the inclusion of salt & pepper noise at densities of 0.01,0.05, and 0.1. Table 9 evaluates the **PSNR** between the secret and the extracted information. Lower density of salt & pepper noise produces better results for image extraction compared to higher density, as indicated in Table 9.



(a) **(b)**



(c) **(d)**

Figure 8: Result of noise attack on the cover image by introducing salt & pepper noise at densities of 0.01,0.05, and 0.1: (a) *S*1, (b) *S*1 obtained after applying salt & pepper noise (density: 0.01), (c) *S*1 obtained after applying salt & pepper noise (density: 0.05), and *S*1 obtained after applying salt & pepper noise (density: 0.1).

Table 9: Noise attack: **PSNR** values

Cover image	**Secret image**	**Noise**	**PSNR**
Baboon (512 x 512)	S_2	Salt & pepper (d=0.05)	27.5923
		Salt & pepper (d=0.01)	20.7954
		Salt & pepper (d=0.01)	17.8001

5. Conclusions

The current paper introduces an image steganographic algorithm that utilizes chaotic maps and modified LSB. The researchers employ three distinct chaotic maps: the logistic and the two-dimensional piecewise smooth nonlinear chaotic maps, which encrypt the secret information, while the third is a two-dimensional mixed Bertrand duopoly chaotic map, used to randomly select locations of the cover pixels. These chosen pixels serve as the storage locations for the cipher image bits. The algorithm ensures that no bits from the secret information are lost. Experimental findings indicate a strong similarity between the histogram of the stego image and the histogram of the cover image. Comparisons with other algorithms in the literature, based on PSNR, affirm that the proposed algorithm outperforms them. The image quality measures yield satisfactory results using our algorithm. The relative entropies among different cover and stego images are almost negligible. Thus, the new steganographic algorithm can be suitable for transmitting secret medical images through communication media. Finally, the quantum image steganography algorithm via **MBDCM** will be proposed to increase the current algorithm's security in the future.

References

1 A. Elghandour, A. Salah, and A. Karawia, (2022) "A new cryptographic algorithm via a two dimensional chaotic map," *Ain Shams Engineering Journal*, vol. **13**, no. 1, p. 101489,.

2 I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker, (2008) Digital watermarking and steganography. Burlington: Morgan Kaufmann, 2 ed.,.

3 R. Matthews, (1989) "On the derivation of a "chaotic" encryption algorithm," Cryptologia, vol. **13,** no. 1, pp. 29–42,.

4 M. Prasad and S. K.L., (2011) "Chaos image encryption using pixel shuffling," Computer Science & Information Technology, vol. **1**, p. 1217, 07.

5 K. Q. Chunhu Li, Guangchun Luo and C. Li, (2017) "An image encryption scheme based on chaotic tent map," Nonlinear Dyn, vol. **87**, pp. 127–133, 08.

6 M. Alawida, A. Samsudin, J. S. Teh, and R. S. Alkhawaldeh (2019), "A new hybrid digital chaotic system with applications in image encryption," Signal Processing, vol. **160**, pp. 45–58,.

7 J. I. Moreira Bezerra, V. Valduga de Almeida Camargo, and A. Molter, (2021) "A new efficient permutation-diffusion encryption algorithm based on a chaotic map," Chaos, Solitons & Fractals, vol. **151**, p. 111235,.

8 M. Alawida, (2023) "A novel chaos-based permutation for image encryption," *Journal of King Saud University - Computer and Information Sciences*, vol. **35,** no. 6, p. 101595,.

9 M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. Ho, and K.-H. Jung, (2018) "Image steganography in spatial domain: A survey," Signal Processing: Image Communication, vol. **65**, pp. 46–66,.

10 F. Djebbar, B. Ayad, H. Hamam, and K. Abed-Meraim, (2011) "A view on latest audio steganography techniques," in 2011 International Conference on Innovations in Information Technology, pp. 409–414,.

11 T. Fang, M. Jaggi, and K. Argyraki, (2017) "Generating steganographic text with LSTMs," in Proceedings of ACL 2017, Student Research Workshop, (Vancouver, Canada), pp. 100–106, Association for Computational Linguistics, **7**.

12 R. Balaji and G. Naveen, (2011) "Secure data transmission using video steganography," in 2011 IEEE International Conference on Electro/Information Technology, pp. 1–5,.

13 Z.-H. Wang, H.-R. Yang, T.-F. Cheng, and C.-C. Chang, (2013) "A high-performance reversible data-hiding scheme for lzw codes," *Journal of Systems and Software*, vol. **86**, no. 11, pp. 2771–2778,.

14 W. Bender, D. Gruhl, N. Morimoto, and A. Lu, (1996) "Techniques for data hiding," *IBM Systems Journal*, vol. **35**, pp. 313–336, 01.

15 A. Westfeld and A. Pftzmann, (2000) "Attacks on steganographic systems," in Information Hiding (A. Pftzmann, ed.), (Berlin, Heidelberg), pp. 61–76, Springer Berlin Heidelberg,.

16 A. Westfeld, "F5—a steganographic algorithm (2001)," in Information Hiding (I. S. Moskowitz, ed.), (Berlin, Heidelberg), pp. 289 302, Springer Berlin Heidelberg,.

17 R. Amirtharajan and J. B. Balaguru Rayappan, (2012) "An intelligent chaotic embedding approach to enhance stego-image quality," Information Sciences, vol. **193**, pp. 115–124,.

18 Y.-Y. Tsai, J.-T. Chen, and C.-S. Chan, (2014) "Exploring lsb substitution and pixel-value differencing for block based adaptive data hiding," *International Journal of Network Security*, vol. **16**, pp. 363–368, 09.

19 S. Vigila and K. Muneeswaran, (2015) "Hiding of confidential data in spatial domain images using image interpolation," *International Journal of Network Security*, vol. **17**, pp. 722–727, 01.

20 A. Seyyedi, V. Sadau, and N. Ivanov, (2016) "A secure steganography method based on integer lifting wavelet transform," *International Journal of Network Security*, vol. **18**, pp. 124–132, 01.

21 S. Arunkumar, V. Subramaniyaswamy, V. Vijayakumar, N. Chilamkurti, and R. Logesh, (2019) "Svd-based robust image steganographic scheme using riwt and dct for secure transmission of medical images," Measurement, vol. **139**, pp. 426–437,.

22 D. Essaidani, H. Seddik, and E. Braiek, (2016) "Asynchronous invariant digital image watermarking in radon feld for resistant encrypted watermark," *International Journal of Network Security*, vol. **18**, pp. 19–32, 01.

23 R. Amirtharajan, R. Subrahmanyam, J. Teja, K. Reddy, and J. B. B. Rayappan, (2013) "Pixel indicated triple layer: A way for random image steganography," *Research Journal of Information Technology*, vol. **5**, pp. 87–99, 02.

24 M. Khaled and A. H. Abu El-Atta, (2021) "Enhanced algorithms for steganography based on least significant bit and secret image compression," in 2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS), pp. 266–272,.

25 C. Chan, C. Chang, and Y. Hu, (2006) "Image hiding scheme using modulus function and optimal substitution table," Pattern Recognit. Image Anal., vol. **16**, p. 208–217, 04.

26 S. Manoharan and D. RajKumar, (2016) "Pixel value differencing method based on cmyk colour model," *International Journal of Electronics and Information Engineering*, vol. **5**, pp. 37–46, 09.

27 B. Jana, (2016) "Dual image based reversible data hiding scheme using weighted matrix," *International Journal of Electronics and Information Engineering*, vol. **5**, pp. 6–20, 09

28 S. Karakus and E. Avci, (2020) "A new image steganography method with optimum pixel similarity for data hiding in medical images," Medical Hypotheses, vol. **139**, p. 109691,.

29 K. Sashi Rekha, M. Joe Amali, M. Swathy, M Raghini, and B. Priya Darshini, (2023) "A steganography

embedding method based on cdf-dwt technique for data hiding application using Elgamal algorithm," Biomedical Signal Processing and Control, vol. **80**, p. 104212,.

30 B. Ray, S. Mukhopadhyay, S. Hossain, S. K. Ghosal, and R. Sarkar, (2021) "Image steganography using deep learning based edge detection," Multimed Tools Appl, vol. **80**, p. 33475–33503,.

31 A. A. Karawia, (2021) "Medical image steganographic algorithm via modified lsb method and chaotic map," IET Image Processing, vol. **15,** no. 11, pp. 2580–2590,

32 A. Pisarchik and M. Zanin, (2012) "Chaotic map cryptography and security*," International Journal of Computer* Research, vol. **19**, no. 1, pp. 49—78,.

33 M. Francois, T. Grosges, D. Barchiesi, and R. Erra, (2014) "Pseudo-random number generator based on mixing of three chaotic maps," Communications in Nonlinear Science and Numerical Simulation, vol. **19**, no. 4, pp. 887 895,.

34 Z. Liu and L. Xi, (2007) "Image information hiding encryption using chaotic sequence," in Knowledge-Based Intelligent Information and Engineering Systems, (Berlin, Heidelberg), pp. 202–208, Springer Berlin Heidelberg,.

35 S. Askar and A. Al-khedhairi, (2020) "The dynamics of a business game: A 2d-piecewise smooth nonlinear map," Physica A: Statistical Mechanics and its Applications, vol. **537**, p. 122766,.

36 A. Awad, S. Askar, and A. Elsadany, (2022) "Complex dynamics investigations of a mixed Bertrand duopoly game: synchronization and global analysis," Nonlinear Dyn, vol. **107**, pp. 3983–3999,.

37 M. Kutter and F. Petitcolas, (1999) "A fair benchmark for image watermarking systems," in Security and Watermarking of Multimedia Contents, vol. **3657**, pp. 226– 239, SPIE, 01.

38 S. Khan, T. Khan, M. Ismail, M. Zafar, R. Ashraf, and N. Ahmad, (2016) "5lsb steganogaraphy using monotonie rgb color image as cover medium," in *2016 Sixth* International Conference on Innovative Computing Technology (INTECH), 08.

39 A. Soria-Lorente and S. Berres, (2017) "A secure steganographic algorithm based on frequency domain for the transmission of hidden information," Security and Communication Networks, vol. **2017**, p. 5397082,