International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 4, August 2021, pp. 3070~3083
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i4.pp3070-3083

Image encryption under spatial domain based on modify 2D LSCM chaotic map via dynamic substitution-permutation network

Rana Saad Mohammed[1], Khalid Kadhim Jabbar[2], Hussien Abid Hilal[3]
[1,2]Department of Computer Sciences, College of Education, Mustansiriyah University, Baghdad, Iraq
[3]Electronica Computer Center, Mustansiriyah University, Baghdad, Iraq

Article Info

Article history:

Received Sep 15, 2020
Revised Dec 25, 2020
Accepted Jan 13, 2021

Keywords:

Block cipher
Confusion-diffusion
Logistic-sine coupling map
Random chaotic map
S-box
Substitution-permutation

ABSTRACT

Image encryption has become an important application aspect of information security. Most attempts are focused on increasing the security aspect, the quality of the resulting image, and the time consumed. On the other hand, dealing with the color image under the spatial domain in this filed is considered as another challenge added to the proposed method that make it sensitivity and difficulty. The proposed method aims to encode a color image by dealing with the main color components of the red (R), green (G), and blue (B) components of a color image to strengthen the dependence of each component by modifying a two dimensional logistic- sine coupling map (2D-LSCM). This is to satisfy the statistical features and reduce time-consumption, and benefit from a mixing step of the second of advanced encryption standard (AES) candidates (serpent block cipher) and modified it to achieve in addition of confusion and diffusion processes. The experimental results showed that our proposed method had the ability to resist against statistical attacks and differential attacks. It also had a uniform histogram, a large key space, complex and faster, closer Shannon entropy to 8, and low correlation values between two adjacent pixels compared with other methods.

Corresponding Author:

Rana Saad Mohammed
Computer Science Department, College of Education
Mustansiriyah University
Baghdad, Iraq
Email: ranasaad2014@gmail.com, drranasaad@uomustansiriyah.edu.iq

1. INTRODUCTION

In one hand, color image is popularly used in the world in different applications in (e.g., politics, economy, and education), on the other hand; because of the friendly programs/applications that simplified the work of different attacks continued its work to penetrate an encryption algorithm of a color image. Several encryption algorithms used 1D, 2D, 3D, and 4D; of a chaotic map depending on the need of encryption algorithm. Furthermore, chaotic systems are one of an important tool that are adopted to generate a secret key which is used on the encryption stage and this is what has been adopted in our proposed method which represents an important aspect of data security. On one hand; it deals with images directly in order to complete the encryption stage makes it within the interest of information security.

From this fact, our contribution to the security of both information and data is due to the methodology that is used to maintain the security of the proposed method that is developed to encrypt the color image. There are different researches about image encryption in spatial domain by using a chaotic

map. This paper classifies them into four classes of related works. The first class uses chaotic maps to construct an S-box for image encryption, such as in [1-7]. The second class uses chaotic maps to get develop confusion-diffusion process such as [8-17]. The third class uses chaotic maps to develop a classical block cipher such as Data Encryption Standard DES, AES for image encryption such as in [18-23]. The fourth class employs chaotic maps in the permutation process as in [24] and another one for permutation to mix R, G, B components as in [25]. The above research used 1D, 2D, 3D, or 4D of chaotic maps depending on the need of the algorithm work.

Chaotic systems have some characteristics that attract the attention of information security researchers. These characteristics are deterministic, nonlinear and highly sensitive to any change in the initial value. So, it is suitable for image encryption [26-29]. However, the researchers still defect and try finding a novel method to encrypt the R, G, B components in mixing not independently by using appropriate confusion-diffusion architecture [25].

The objective of this paper is to mix the red (R), green (G), and blue (B) components of color image in parallel to strengthen the dependence of each component, to satisfy the confusion-diffusion properties and to reduce time consumption to mix them in parallel by modifying 2D-LSCM, in addition to modifying a mixing step of the 2nd AES candidates (serpent block cipher) to produce a dynamic permutation and a substitution network of these four sub-images to achieve the confusion and diffusion processes.

2. ADVANCED ENCRYPTION STANDARD (AES)

AES is a symmetric square code, which has capacity for encryption and unscrambling information in blocks as the states that changed over at each phase of encryption or unscrambling measure. Then again, the key is shown up as a square exhibit of a few bytes, where any key is comprehensive into a framework of keys that booked words. Each word has a size of four bytes, so; when utilizing 128-bits, the entire key of timetable is 44 words with the size of (128) bits by utilizing a few key scopes of (128) bits, (192) bits or (256) bits separately. Which is demonstrated as: AES-128, AES-192 or AES-256 as the key reach, AES tasks are done in two measurements exhibited with size of 4×4 bytes known as the states that changed over at each phase of encryption or unscrambling measure. Furthermore, the key is shown up as a square cluster of a few bytes, where any key is comprehensive into a grid of keys that booked words. Each word has a size of four bytes, so, when utilizing 128-bits, the entire key timetable is 44 words [22].

3. THE PROPOSED METHOD

3.1. Statement of the problem

The problem that is encountered here is how to deal with the main color components (red, green and blue) separately due to the naturalness of each sub-bands and how this is done with the algorithms that are used in our proposed method. On the one hand, the process of mixing these main color components is very important and sensitive to any error in the mixing stage such as mathematical calculations or encryption stage or others. On the other hand, the consuming time during the mixing and encryption stage is possibly exploited by attackers through statistical attacks or clear text recognition attacks to capture information about the clear image.

From this standpoint, our contribution in the proposed method is to develop a method that has to have the ability to encrypt/ decrypt the color image based on chaotic system by using several encryption algorithms by mixing the main color components R, G, and B through dealing with each sub-color in parallel after dividing the original image into 4×4 blocks and then mixing the main color components R, G, and B in order to increase the chaotic aspect of the 2D LSCM function, which leads to an increase in the dependence of the color components on each other to increase the complexity in order to make it difficult to break the security aspect of the proposed method, as a result that will produce a substitution-permutation network to achieve the statistical characteristics and reduce the time consumed. The color mixing and modification enhance the strength of the serpent encryption algorithm, which is the second candidate of the algorithms that used in the block encryption stage after AES. All these contributions are suggested in the proposed method in addition to the encryption and decryption stages.

3.2. Random chaotic map

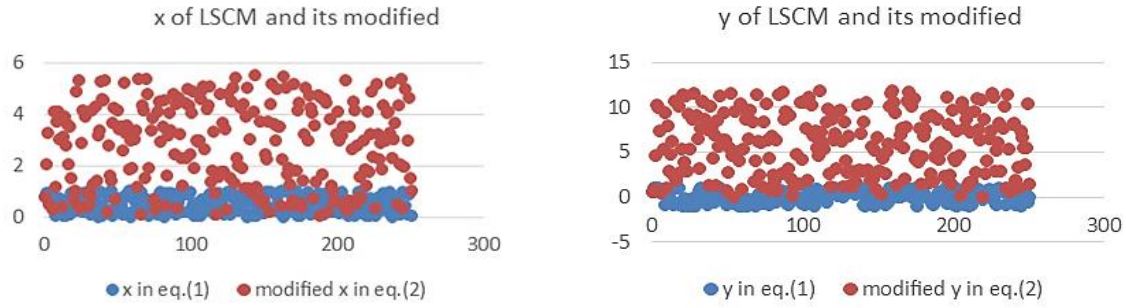
Our proposed method needs a 2D map in the proposed encryption method of color image. So, it modifies a 2D-LSCM that generates a sequence of random keys to improve a permutation and substitution for encryption a color image. The original 2D-LSCM is described in (1) as [28]:

$$\begin{cases} x_{i+1} = \sin(\pi(4\theta x_i(1-x_i) + (1-\theta)\sin(\pi y_i))) \\ y_{i+1} = \sin(\pi(4\theta y_i(1-y_i) + (1-\theta)\sin(\pi x_{i+1}))) \end{cases} \quad (1)$$

where: x_i,y_i,θ[0.1], where we modify of 2D-LSCM is described in (2) by applying a random variant generation (Uniform distribution) $X_i = a + (b-a)R_i$ on (1) and then computing a modulation of the average result random variantX_i. Where R_i is either x_{i+1} or y_{i+1} from (1):

$$\begin{cases} x_{i+1} = (a + (b-a)\sin(\pi(4\theta x_i(1-x_i) + (1-\theta)\sin(\pi y_i))))mod\ avx \\ y_{i+1} = (a + (b-a)\sin(\pi(4\theta y_i(1-y_i) + (1-\theta)\sin(\pi x_{i+1}))))mod\ avy \end{cases} \quad (2)$$

where: avx is average of result sequence of x_{i+1} in (2) and avy average of result sequence of y_{i+1} in (2). Figure 1, shows the difference of (x_{i+1},y_{i+1}) values in the modified 2D-LSCM compared with the original map, for example "i=520, x (1) =0.8, y (1) =0.5, θ =0.99, a=10, b=50".



(a) (b)

Figure 1. Modified LSCM and its original, (a) Equation of x, (b) Equation of y

When using a "Lambda (L)" measurement in the test [30] which is a dominant lyapunov exponent, if (L) is positive, this indicates the absence of randomness in chaotic map. On one hand; on the other hand, if (L) is negative, this indicates the presence of randomness in chaotic map. For the above equations (L= 1.12087905 of x in (1)) decrease to (L= -0.263533223 of x (2)) and (L= -0.243042098 of y in (1)) decrease to (L= -0.869752856 of y in (2)) that indicates a random sequence of our modified LSCM map in (2) compared with its original in (1). Our proposed method used (2) to generate the two key sequences according to the following steps.

3.3. Dynamic permutation and substitution stages

The important process in the cryptography is a permutation and substitution stages that are used to investigate the confusion and diffusion according to Shannon theory. Diffusion means the relationship between the plain image and cipher image is complex, while the confusion means the relation between the cipher image and the key is complex and difficult to be understood. Even if an attacker somehow obtains one plaintext corresponding to one cipher text-a known-plaintext attack, a chosen plaintext or chosen-cipher text attack-the confusion and diffusion make it difficult for the attacker to recover the key. Even if an attacker somehow obtains one plaintext corresponding to one cipher text-a known-plaintext attack, a chosen plaintext or chosen-cipher text attack-the confusion and diffusion make it difficult for the attacker to recover the key. Figure 2 shows the sample permutation of 100 values of the image block. Figure 2(b) shows the changes of 100-pixel positions after permutation compared with Figure 2(a) that shows the original positions before permutation.

```
Algorithm 1: Proposed Dynamic Permutation
Input   : Color image and padding it to get square size (n×n).
          Initial value of map (i, x (1), y (1), θ, a, b).
Output  : permuted image vector.
Begin
1. Read the plain image.
2. Save the vector of image block values with length (m×m×3).
```

```
3. Save the vector of a random sequence of the key from x or y in (2).
4. Permute the vector in step 2 by ascending or descending sort the vector in step 3.
5. Save the permuted values of the image. End;
```

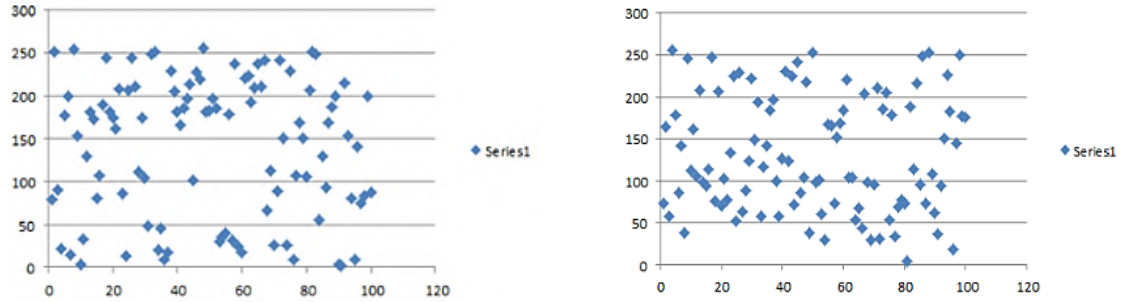


Figure 2. Dynamic proposed permutation example, (a) Sample 100 values from image block before permutation, (b) Sample 100 values from image block after permutation

While the dynamic Substitution box (S-box), illustrated in the algorithm, and it can be created based on the generated sequence key from (2) as in algorithm 2. Figure 3, shows the sample substitution of the previous permuted 100 values which are generated from algorithm 1, Figure 3(b) shows the values of 100 pixels which are changed after substitution compared with original values in Figure 3(a).

Algorithm 2: Proposed Dynamic Substitution box (S-box)

```
Input  : Permuted sub image.
Output : Substituted sub image.
Begin
1. Save the vector with length 256 of S-box index and with values (0 to 255) since the
   image is color.
2. Save the vector with length 256 of random sequence of key from x or y in (2).
3. Permute the vector in step 1 by ascending or descending sort the vector in step 2.
4. Save the permuted S-box index with length 256.
5. Read the permuted image;
   a. Get the value.
   b. finds the index of this value.
   c. Find the substitute value from the sbox index and save it in substitute vector.
6. Convert the substitute vector to image size (m×m×3). End;
```

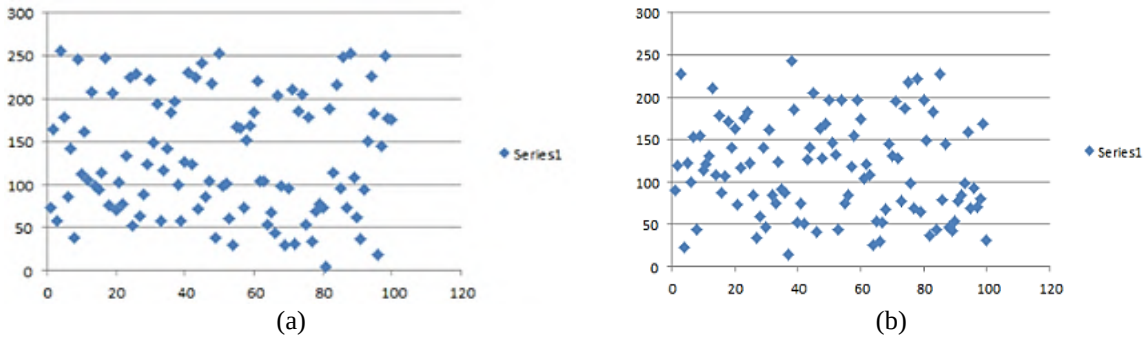


Figure 3. Dynamic proposed substitution, (a) Sample 100 values from image block before substitution, (b) Sample 100 values from image block after substitution

3.4. Mixing stage

The mixing stage in our proposed method is based on the AES, candidates (serpent block cipher) by using our proposed (2) to strengthen the dependence of each (R, G, and B) component on each other, satisfy the confusion-diffusion properties, and reduce time consumption. This step has other property that is simple, and it can be used in the modern processor with a minimum number of pipeline stalls [19]. Figure 4 shows a

process of proposed mixing step, where it applies the following operations (XOR, round shift to left, and round to the nearest integer) with the generated key sequences X_i and Y_i from (2).

Algorithm 3: The proposed mixing stage

```
Input  : four blocks each with a same size.
Output : new four blocks each with a same size.
Begin
1.  Apply round to left by one bit of 1st block.
2.  Apply round to left by two bits of 3rd block.
3.  Get a new 2nd block by doing XOR operation for old 2nd block with the resulted blocks
    from steps 1 and 2.
4.  Generate a sequence of random key from x in (2).
5.  Get a new 4th block by doing XOR operation for old 4th block with the resulted blocks
    from steps 2 and 4.
6.  Apply round to left by three bits of resulted block from steps 3.
7.  Apply round to left by four bits of resulted block from steps 5.
8.  Get a new 1st block by doing XOR operation for blocks in steps 1, 5, and 7.
9.  Generate a sequence of random key from y in (2).
10. Get a new 3rd block by doing XOR operation for blocks in steps 2, 7, and 9.
11. Apply round to left by five bits of resulted block from steps 8.
12. Apply round to left by six bits of resulted block from steps 10. End;
```

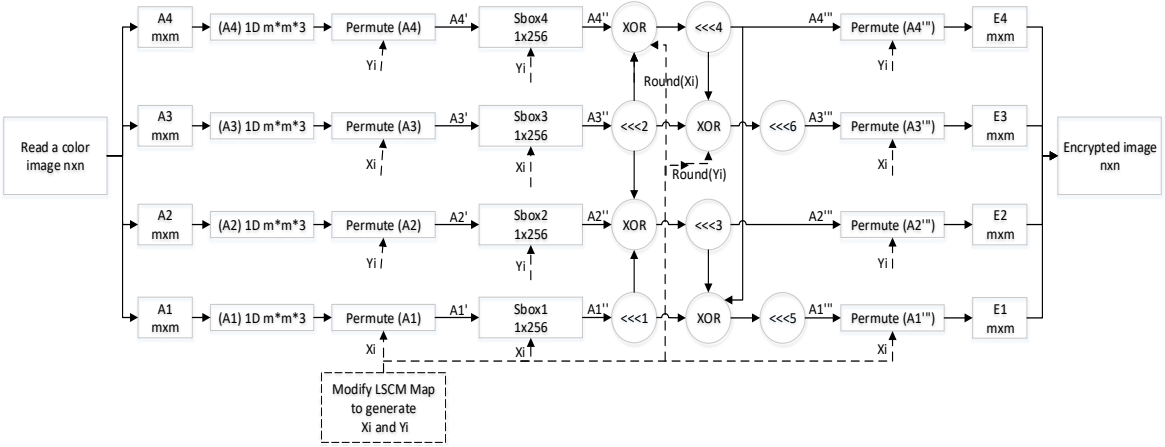


Figure 4. General block diagram of proposed method

3.5. Encryption stage

The general diagram of our proposed encryption method illustrated in Figure 4 after an agreement between the sender and receiver about the initial values of the map, the sender generates the two key sequences by using (2). At first, read a color image size of (256×256) and divide it into four blocks each size of (64×64), then apply initial permuted and substituted of these blocks using algorithm 1 and 2, respectively to produce permuted and substituted blocks. After that, perform a proposed mixing step in algorithm 3 and also apply a final permuted in algorithm 1 to produce the encrypted image, as in the encryption stage illustrated in the following algorithm, while Figure 4 shows the block diagram of encryption stage:

Algorithm 4: Encryption Stage

```
Input  : Color image with initial value of map (i, x (1), y (1),θ, a, b)
Output : Cipher image.
Begin
1.  Read the plain image and padding it with zero to get a square image (n×n) size.
2.  Divide the original image into four sub-images (A1, A2, A3, and A4) each one with size of
    m×m.
3.  Reshape these sub-images into 1D each one with a size of (m × m × 3).
4.  Generate a sequence of the key with length (m × m ×3) from Xi and Yi of a modified LSCM
    chaotic map from (2).
5.  Using algorithm 1 to initial Permute an odd sub-image (A1 and A3) by sort a random
    sequence key Xi and permute an even sub-images (A2 and A4) by sort a random sequence key
    Yi.
6.  Using algorithm 2 to create four dynamic Substitution-Box (8 × 8 S-box) to substitute
    the resulted permuted sub-images (A'1, A'2, A'3, and A'4).
```

a. To create an odd S-box ($Sbox_1$) by permuting an index from (0 to 255) by using an ascending sort of a random sequence key X_i to substitute the elements of A'_1,
b. Another side; to create ($Sbox_3$) by permuting an index from (0 to 255) by using a descending sort of a random sequence key X_i to substitute the element of A'_3.
c. To create an even S-box ($Sbox_2$) by permuting an index from (0 to 255) by using an ascending sort of a random sequence key Y_i to substitute the element of A'_2,
d. Another side to create ($Sbox_4$) by permuting an index from (0 to 255) by using a descending sort of a random sequence key Y_i to substitute the element of A'_4.

7. Using algorithm 3 to apply a proposed mixing step between the outputs from step 6 ($A''_1, A''_2, A''_3, and\ A''_4$):
$A'''_1 = A''_1$ <<< 1
$A'''_3 = A''_3$ <<< 2
$A'''_2 = A''_2 \oplus A'''_3 \oplus A'''_1$
$A'''_4 = A''_4 \oplus A'''_3 \oplus$ Round (X_i)
$A'''_2 = A'''_2$ <<< 3
$A'''_4 = A'''_4$ <<< 4
$A'''_1 = A'''_1 \oplus A'''_2 \oplus A'''_4$
$A'''_3 = A'''_3 \oplus A'''_4 \oplus$ Round (Y_i)
$A'''_1 = A'''_1$ <<< 5
$A'''_3 = A'''_3$ <<< 6
8. Using algorithm 1 again to final permute an odd output of step 7 (A'''_1 and A'''_3) by sort a random sequence key X_i. And permute even outputs (A'''_2 and A'''_4) by sort a random sequence key Y_i.
9. Reshape each of the permuted outputs into 2D each with size of (m × m) to get four sub-encrypted images ($E_1, E_2, E_3, and\ E_4$).
10. Merge sub-encrypted images ($E_1, E_2, E_3, and\ E_4$) into one encrypted image. End;

3.6. **Decryption** sta**ge**

On the other hand, the receiver generates the same two key sequences by using (2) that is generated by the sender that needed in decryption stage after reversing the processes in the encryption algorithm. At first, read an encrypted image (256×256) and divide it into (4×4) blocks each size of (64×64), then apply re-final permuted of these blocks by using the reverse of algorithm 1 to produce a re-permuted block. After that, perform a proposed mixing step by the reverse of algorithm 3, then apply the reverse of algorithm 2, and 1, respectively to produce re-initial permuted and re-substituted blocks to get an original image. The following algorithms that are described in Figure 4. Where the symbols ($A_1, A_2, A_3, and\ A_4$) are the sub-images of a color image; the symbols (E_1, E_2, E_3, $and\ E_4$) are sub-images of encrypted image; the symbol (1D m×m×3) means concatenate R, G, and B components into one dimension.

Algorithm 5: Decryption Stage

Input : A cipher image with initial values of the map (i, x (1), y (1), θ, a, b).
Output : Original image.
Begin

1. Read the cipher image.
2. Divide the cipher image into four sub-images ($E_1, E_2, E_3, and\ E_4$) each with size m x m.
3. Reshape these sub-images into 1D each with the size of (m × m × 3).
4. Generate a sequence of the key with length (m × m × 3) from X_i and Y_i of the modified LSCM chaotic map from (2).
5. Using algorithm 1 to re-final permute an even outputs of step 3 (A'''_2 and A'''_4) by reverse the sort of a random sequence key X_i, and permute an odd output (A'''_1 and A'''_3) by reverse sort a random sequence key Y_i.
6. Using algorithm 3 in reverse to apply a reverse of proposed Mixing step as follows to get the outputs (A''_1 , A''_2 , A''_3 , $and\ A''_4$):-
$A''_3 = A'''_3$ >>> 6
$A''_1 = A'''_1$ >>> 5
$A''_3 = A''_3 \oplus A'''_4 \oplus$ Round (Y_i)
$A''_1 = A''_1 \oplus A'''_2 \oplus A'''_4$
$A''_4 = A'''_4$ >>> 4
$A''_2 = A'''_2$ >>> 3
$A''_4 = A''_4 \oplus A''_3 \oplus$ Round (X_i)
$A''_2 = A''_2 \oplus A''_3 \oplus A''_1$
$A''_3 = A''_3$ >>> 2
$A''_1 = A''_1$ >>> 1
7. As a step 6 of encryption algorithm, using algorithm 2 to create four dynamic Substitution-Box (8 × 8 S-box) to substitute the resulted Mixed sub-images to get another sub-images (A'_1 , A'_2 , A'_3 , $and\ A'_4$).
8. Using algorithm 1 to re-initial permutes an odd sub-image (A'_1 and A'_3) by reverse sort of a random sequence key X_i, and permute even sub-images (A'_2 and A'_4) by reverse sort of a random sequence key Y_i.

9. Reshape these sub-images into two-dimension (2D) each with size of (m × m) to get four sub-original images (A_1, A_2, A_3, $and\ A_4$).
10. Merge sub-original images (A_1, A_2, A_3, $and\ A_4$) into one original image. END;

4. PERFORMANCE AND SECURITY ANALYSIS

This section shows the security and statistical analysis of the proposed method and comparison with other methods. This section uses R, G, and B histogram, entropy, correlation with adjacent pixel, NPCR with UACI for differential attack test, key sensitivity, key space size against brute force attack, and time complexity measurement. The results below show the results that are compared with other methods.

4.1. RGB histogram

A good encryption method has a histogram graph like a noise. So, our encryption algorithm tested with five different samples of color image (it takes into consideration image texture). These images are different in R, G, and B components as shown in Figure 5, it gives the R, G, and B histogram for the original image and its encrypted. The R, G, and B histogram of encrypted images in all experiments has a uniform form, as in Figure 6, however, this shows the confidence and strength of our proposed method and it can hide the original image information from the attacks.

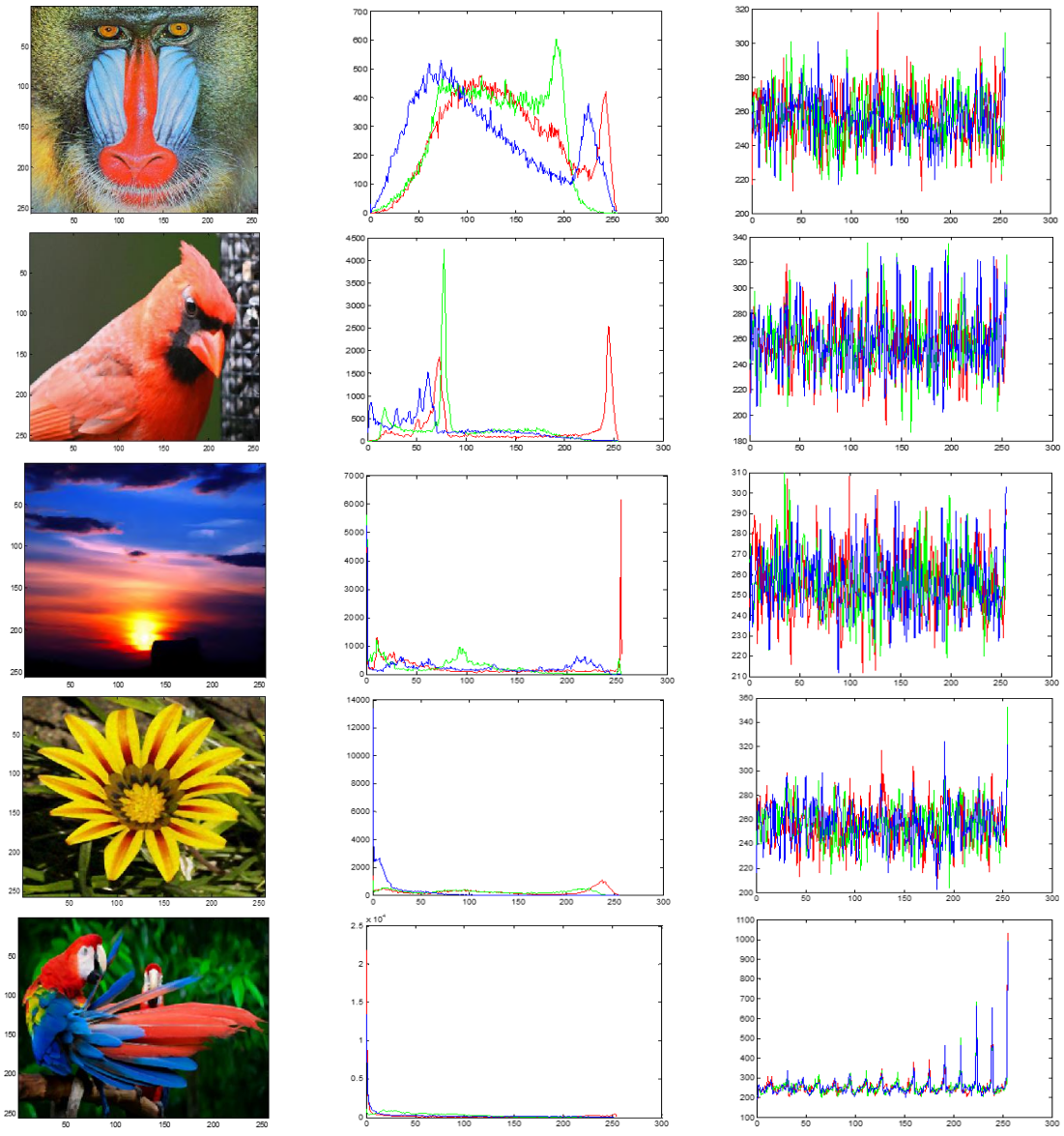


Figure 5. R, G, and B histograms of original images and its encryption

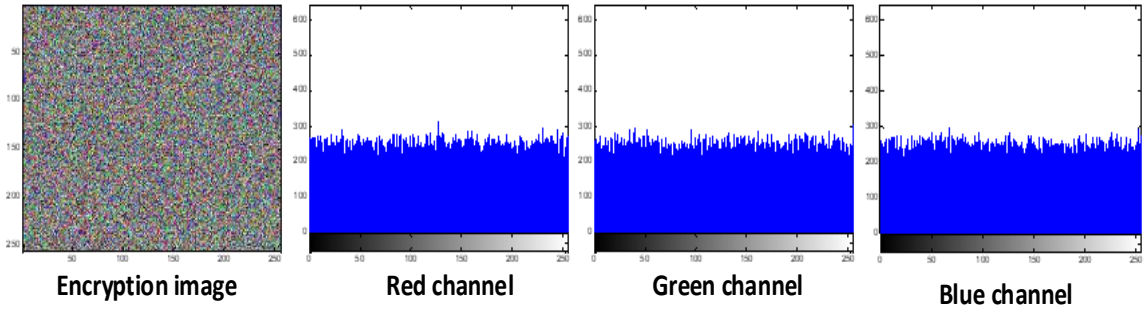


Figure 6. Uniform R, G, and B histograms of encryption image

4.2. Shannon entropy

The entropy test is a measure of information that closes to a random source. This test can be computed as (3).

$$H(s) = \sum_{i=0}^{M-1} p(m_i) log \frac{1}{p(m_i)} \quad (3)$$

Where M is the number of a symbol, $m_i \in$ s, p (m_i) is the probability of occurrence of symbolm_i, a log is denoting as base 2 logarithm. An ideal standard information entropy value is 8 [26]. From Table 1, note that our proposed method is close to 8 (range from 7.96823 to 7.99907) and it is secure against entropy attack, thus the entropy comparison assessed between our proposed method and the other methods is illustrated in Table 2.

Table 1. Shannon entropy test of the proposed method

Sample	Shannon entropy
Baboon	7.99898
Canary	7.99492
Sun Set by pakkano	7.99907
Sunshine ron	7.99843
Parrot	7.96823

Table 2. Comparison results of Shannon entropy test with other methods

Method	Shannon Entropy Test	
	Baboon	Lena
[1]	7.9581	7.9965
[3]	7.8853	7.99932
[10]	7.7564	7.9963
[18]	7.8704	7.9040
[25]	7.999.3	7.9992
Our Proposed Method	7.99898	7.99981

4.3. Correlation analysis of two adjacent pixels

In the plain image, the pixel has high correlate with its adjacent pixel in the horizontal (H), vertical (V), and diagonal (D) location while in the encryption image the correlation is decreased. The correlation between adjacent pixels is measured according to (3), and the results are illustrated in Table 3.

$$r_{xy} = \frac{cov(x.y)}{\sqrt{D(x)D(y)}} \quad (4)$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x_i))^2 \quad (5)$$

$$cov(x.y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x_i))(y_i - E(y_i)) \quad (6)$$

Table 3, shows that the correlation of plain image is high while in the encryption image is decreased, also; Figure 7 shows this correlation for each of the five samples. While Figure 8 shows the correlation of encryption, image is similar of all samples that are used, and Table 4 shows the correlation analysis (H, V, and D) with other methods where the image sample is "Lena", because it is smoothed (standard). From Table 4, our proposed method has low correlation values (with sample Lena) that compared with other methods, while Table 5 shows the results of correlation analysis (x, y) between our proposed method and others, where image sample "Baboon" is used (high texture).

Table 3. Correlation analysis of the two adjacent pixels of proposed method

Image sample	Baboon		Canary		Sun Set by pakkano		Sunshine ron		Parrot	
Direction	Original	Encrypt.	Original	Encrypt.	Original	Encrypt.	Original	Encrypt.	Original	Encrypt.
Horizontal	0.810	- 0.002	0.980	-0.002	0.998	0.001	0.979	0.009	0.972	-0.004
Vertical	0.760	0.001	0.986	0.004	0.994	-0.004	0.975	-0.002	0.971	0.001
Diagonal	0.747	0.002	0.967	-0.002	0.993	-0.001	0.964	-0.002	0.949	0.00060

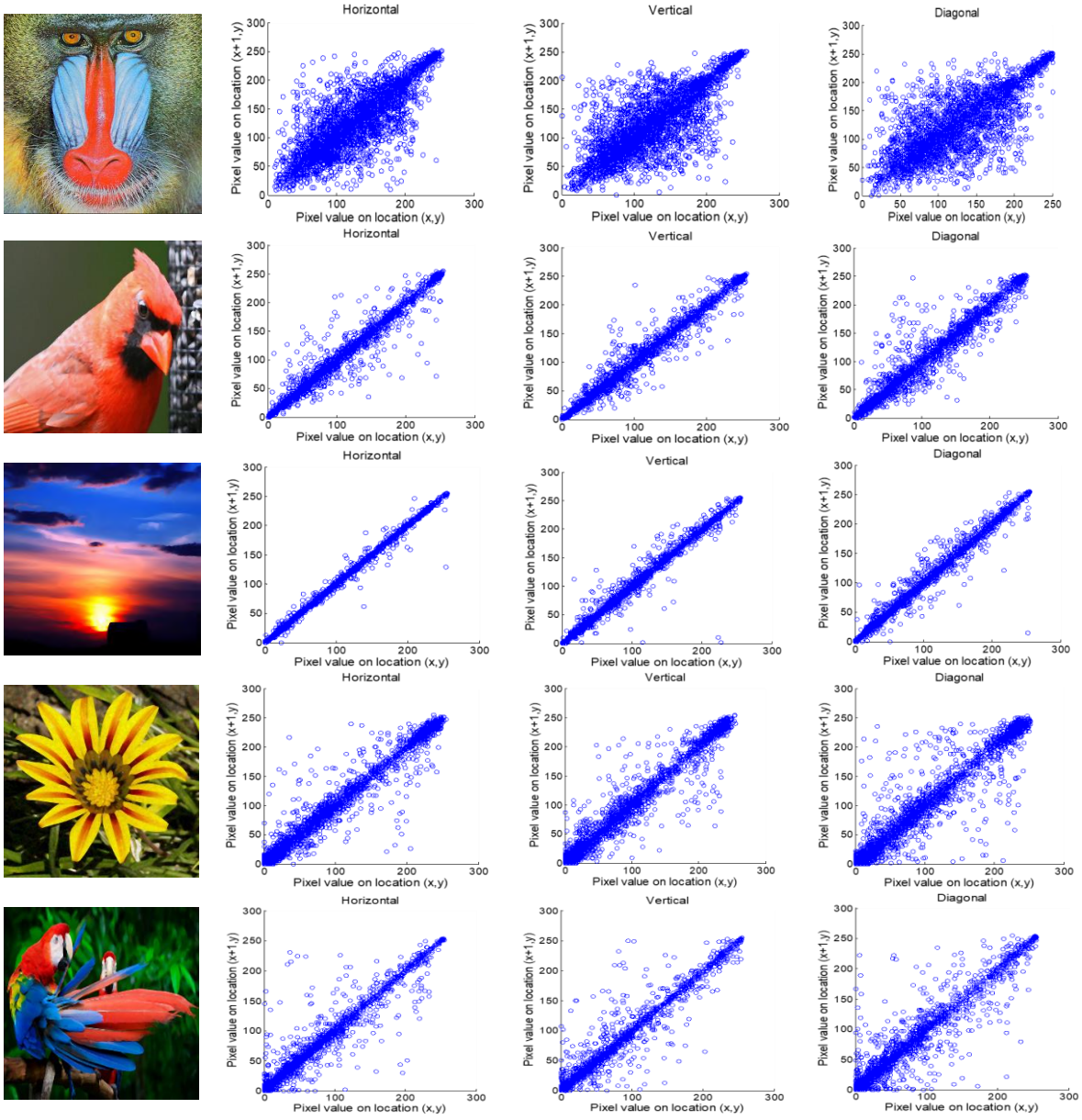


Figure 7. Correlation analysis of original images

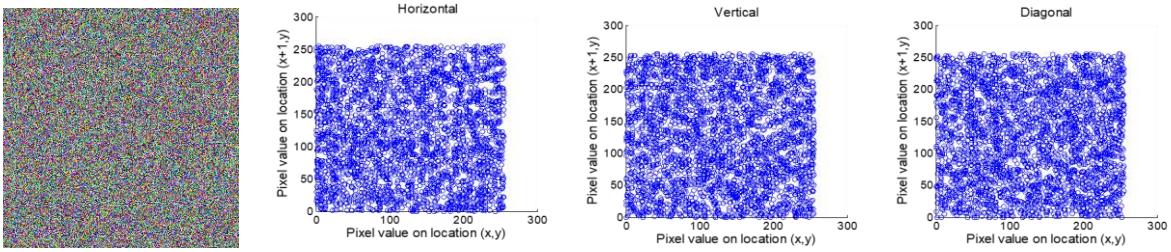


Figure 8. Correlation analysis of encryption image

Table 4. Comparison of correlation analysis (H, V, and D) with other methods

Encryption Method	Horizontal (H)	Vertical (V)	Diagonal (D)
[3]	- 0.0010	- 0.0010	- 0.0004
[10]	- 0.0021	0.0143	0.0085
[25]	0.0064	0.0045	0.0057
Our proposed Method	- 0.0042	0.0027	- 0.0061

Table 5. Comparison of correlation analysis (x, y) with other methods

Encryption Method	Correlation (x ,y)
[1]	- 0.0031
[18]	0.0315
Our Proposed Method	- 0.0044

4.4. Differential attack

This type of attack is one of the common attacks in the cryptography. It is the comparison between the encryption of plain image and the encryption of changing one pixel or more (even one bit) of the plain image. NPCR and UACI are the two standard methods to measure this property as (7) and (8), respectively [2].

$$NPCR(C_1.C_2) = \frac{1}{M \times N} \sum_{I=1}^{M} \sum_{J=1}^{N} D(i.j) \quad (7)$$

where $D(i.j) = \begin{cases} 0 & C_1(i.j) = C_2(i.j) \\ 1 & C_1(i.j) \neq C_2(i.j) \end{cases}$

$$UACI(C_1.C_2) = \frac{1}{M \times N} \sum_{i=}^{M} \sum_{i=1}^{N} \frac{|C_1(i.j) - C_2(i.j)|}{255} \quad (8)$$

The standard value of NPCR is ≃99 and UACI≃33, Table 6 shows that our proposed method is survived under different attacks, while Table 7 presents the comparative value of NPCR and UACI that test with the other method.

4.5. Key sensitivity

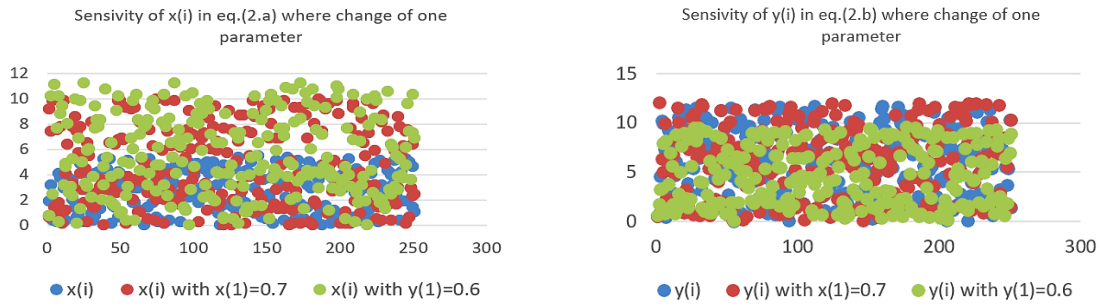
Encryption algorithm should be sensitive to a secret key that means that encrypted method cannot be decrypted correctly if there is only a slight change in the secret key. The proposed method has a key sensitivity when changing only one parameter of the secret keys and keeping all other keys unchanged. The secret keys in our modified (2) are (i, x (1), y (1), a, b, θ). Figure 9 shows the sensitivity of x and y in the developed (2) with the change of one parameter (e.g., x (1)=0.7 or by (1)=0.6), since this property enables our proposed method to immune against the cipher text and plaintext attack.

Table 6. NPCR, UCAI of proposed method

Sample	NPCR	UACI
Baboon	99.7231	33.5562
Canary	99.5862	33.5213
Sun Set by pakkano	99.6531	33.7189
Sunshine ron	99.8586	33.7400
Parrot	99.7156	33.4628

Table 7. Comparison results of NPCR, UACI with other methods

Method	Sample			
	Baboon		Lena	
	NPCR	UACI	NPCR	UACI
[1]	98.995	30.2783	97.012	28.112
[3]	99.6040	33.978	99.121	33.4614
[10]	99.6293	32.456	98.112	15.4133
[25]	99.6376	33.5149	99.6403	33.4968
Our proposed Method	99.7231	33.5562	99.6596	33.4898



(a) (b)

Figure 9. Sensitivity of proposed method where change of one parameter, (a) Sensitivity of x in (2), (b) Sensitivity of y in (2)

4.6. Brute force attack

Key space size is the total number of different keys that can be used for the encryption algorithm; it should be large enough to resist the brute force attack. The precision of initial values and the parameter of the proposed method are (10^{16}). The key space size of secret keys is(10^{16})^k. The number of initial values and parameters is k=8 (i, x (1), y (1), a, b, θ, avx, avy). Then the key space size is the sum of two sequences generated of (2) (X_i and Y_i), respectively ((10^{16})^8+(10^{16})^8), therefore; at least the key space size of our proposed method is (10^{256}), it is large enough against brute force attack.

Another calculation of key space size in binary; for the most used double-precision floating-point format which usually refers to binary64, the key space (as specified by the IEEE 754 standard) is 2^{52} [25]. The number of initial values and parameters is k = 8 (i, x (1), y (1), a, b,θ, avx, avy), then the key space size is the sum of two sequences generated of equation 2 (X_i,Y_i), respectively ((2^{52})^8 + (2^{52})^8) so that the at least key space size of our proposed is (2^{832}). The key space should be greater than the $2^{200}\approx10^{30}$ [7] to withstand or be resistant to the brute-force attack. From these calculations; our proposed method has larger key space that it is enough and resistant against brute force attack.

4.7. Time complexity

In our proposed method, the computational cost consists of generating 2D-LSCM key sequence, initial and final permutation, substitution and mixing steps in parallel. A two key sequences (X_i,Y_i) with length M×N are generated. So, the computational cost is Θ (2×M×N).

The cost of initial and final permutation processes is Θ (2×M×N). The cost of substitution process is Θ(M×N). The cost of mixing steps (XOR, <<< and round to nearest integer) is Θ (3×M×N). The total time complexity in our encryption and decryption stages is Θ (8×M×N). The time complexity of the proposed algorithm is lower. Hence the speed of our encryption scheme is faster compared with the related work [25] that has Θ (9×M×N).

5. CONCLUSION

In this paper, a color image encryption algorithm using a chaotic system based on modified 2D LSCM under the spatial domain has been proposed. Our proposed method deals with a set of main and important axes in addition to the proposed contributions to achieve the desired goal of the proposed method, as dealing with the color image is a major challenge in this area due to the variation and difference in the nature of the colored images, on the other hand, the encryption stage is subject to a set of standards of the

task that is adopted, such as the security aspect, the time consumed, the quality of the resulting image, in addition to the fact of dealing with the chaotic system in this area requires us to achieve a set of determinants on the basis of which chaotic systems are built. All of this adds to the task of harnessing all the tools and algorithms and the proposed contributions for encryption the color image under spatial domain. The process of dealing with the contrast and difference in the nature of color images is exceeded in the proposed method of taking advantage of the characteristics of the color mixing algorithm and dealing with the main color components R, G, and B in parallel.

On the other hand, the benefit from the characteristics of the chaotic system is evident through the development of modified 2D LSCM chaotic map, which was tested and found to meet the criteria for chaotic systems of randomness, and complexity, in addition to using known and approved encryption algorithms in order to achieve the goal of the research. It is accomplished in parallel by using 2D LSCM to generate a random key sequence that are used to create a dynamic substitution-permutation network to satisfy the confusion-diffusion properties and to reduce the time consumption. From the map that is presented, a dynamic S-box (substitution stage) is proposed, and the mixing stage of the second AES candidates (serpent block cipher) is modified to strengthen the dependence of each R, G, and B component on another and achieved in addition confusion and diffusion stages.

The results that are arrived at show the success of the proposed method in encoding the color image while maintaining the confidentiality and security of the proposed method in addition to obtaining a high-quality encrypted image. The time factor has been controlled, and sometimes the time taken has been reduced due to the image texture. It makes the process of mixing the main colors less complicated, such as standard image. As for high texture image or painted images, the time taken is more, but in general it is less than others, and it has been controlled by the proposed method, which is another contribution to us. Our proposed method is sensitive to any change in the initial value of chaotic map. Furthermore, it has a better statistical feature that immune against statistical and differential attacks, the large key space against brute-force attack is obtained with low correlation values between two adjacent pixels. Our proposed method has a uniform histogram against chosen-plaintext and cipher text attacks; it has complexity and it is faster after compared with other methods.

The proposed method dealt with an important aspect of information security in terms of dealing with the color image, and on the other hand, the development that was addressed in the mechanism with the use of the chaotic system and the mathematical equations that were modified and used are considered as a contribution to the aspect of data security, it has been tested under several well-known tests in this field, which is considered an important criterion in knowing the strength and success of the proposed method, and the results show the possibility of the proposed method to pass all tests and succeed in them, as detailed in the results of the section.

ACKNOWLEDGEMENTS

The author(s) would like to thank Mustansiriyah University (www.uomustansiriyah.edu.iq) Baghdad-Iraq for its support in the present work.

REFERENCES

[1] S. Farwa, "An Image Encryption Technique based on Chaotic S-Box and Arnold Transform," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 6, pp. 360-364, 2017.
[2] Y. Pascal, K. Nkandeu, and A. Tiedeu, "An Image Encryption Algorithm Based on Substitution Technique and Chaos Mixing," *Springer Science*, 2018. [Online]. Available: 10.1007/s11042-018-6612-2.
[3] F. M. A. Ben, "An Image Encryption Scheme Based on a New Hybrid Chaotic Map and Optimized Substitution Box," *Springer Nature B.V., Nonlinear Dyn*, 2019.
[4] A. Kumar and N. Raghava, "Selective Colour Image Encryption Using Hénon Chaotic System with a Keyless Substitution Cipher," *Engineering and Applied Science Research*, vol. 47, no. 1, pp. 66–76, 2020.
[5] Prajwalasimha S. N. and Basavaraj L., "Performance Analysis of Transformation and Bogdonov Chaotic Substitution Based Image Cryptosystem," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 188–195, 2020.
[6] F. E. Canales, "Pseudo-Random Bit Generator Using Chaotic Seed for Cryptographic Algorithm in Data Protection of Electric Power Consumption," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 2, pp. 1399–1409, 2019.
[7] C. R. Revanna and C. Keshavamurthy, "A New Partial Image Encryption Method for Document Images Using Variance Based Quad Tree Decomposition," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 786–800, 2020.
[8] Z. Hua, "2D Logistic-Sine- Coupling Map for Image Encryption," *Signal Processing*, pp. 148-161, 2018, doi: 10.1016/j.sigpro.2018.03.010.

[9] S. Sharma, "Improved Method for Image Security Based on Chaotic-Shuffle and Chaotic-Diffusion Algorithms," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 9, no. 1, pp. 273–280, 2019, doi: 10.11591/ijece.v9i1.pp.273-280
[10] X. Wang, S. Liny, and Y. Li, "A Chaotic Image Encryption Scheme Based on Cat Map and MMT Permutation," *Modern Physics Letters B*, vol. 33, no. 27, 2020.
[11] Prajwalasimha S. N., A. V. Kumar, and Arpitha C. R., "On the Sanctuary of A Combined Confusion and Diffusion Based Scheme for Image Encryption," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 9, no. 1, pp. 3258-3263, 2019.
[12] G. Jiao, X. Peng, and K. Duan, "Image Encryption with The Cross Diffusion of Two Chaotic Maps," *KSII Transformation on internet and information systems*, vol. 13, no. 2, pp. 1064–1079, 2019.
[13] Q. Cai, "A Secure Image Encryption Algorithm Based on Composite Chaos Theory," *International Information and Engineering Technology Association*, vol. 36, no. 1, pp. 31–36, Feb. 2019.
[14] Y. Liu, "A Multidimensional Chaotic Image Encryption Algorithm Based on the Region of Interest," *Springer Science, LLC, part of Springer Nature*, 2020.
[15] B. Mondal, P. K. Behera, and S. Gangopadhyay, "A Secure Image Encryption Scheme Based on a Novel 2D Sine–Cosine Cross Chaotic (SC3) Map," *Journal of Real-Time Image Processing, Springer-Verlag GmbH Germany, part of Springer Nature*, 2020.
[16] A. Susanto, "Triple Layer Image Security Using Bit-Shift, Chaos, and Stream Encryption," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 9, no. 3, pp. 980–987, 2020.
[17] D. F. Chalob, "A New Block Cipher for Image Encryption Based on Multi Chaotic Systems," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 18, no. 6, pp. 2983–2991, 2020, doi: 10.12928/TELKOMNIKA.v18i6.13746.
[18] Ü. Çavu, "A Novel Hybrid Encryption Algorithm Based on Chaos and S-AES Algorithm," *Springer Science Business Media B.V., Nonlinear Dyn*, 2018.
[19] I. Hussain, "Image Encryption Based on Chebyshev Chaotic Map and S8 S-Boxes," *Optical Applicator*, vol. XLIX, no. 2, 2019.
[20] R. Rimani, "Image Encryption by AES Algorithm Based on Chaos-Permutation," *Malaysian Journal of Computing and Applied Mathematics*, vol. 2, no. 2, pp. 14–24, 2019.
[21] C. Yang and Y. Chien, "FPGA Implementation and Design of a Hybrid Chaos-AES Colour Image Encryption Algorithm," *Symmetry*, vol. 12, p. 189, 2020.
[22] S. A. Mehdi, K. K. Jabbar, and F. H. Abbood, "Image Encryption Based on the Novel 5D Hyper-Chaotic System via Improved AES Algorithm," *International Journal of Civil Engineering and Technology*, vol. 9, no. 10, pp. 1841-1855, 2018.
[23] S. R. Maniyath and Thanikaiselvan V., "A Novel Efficient Multiple Encryption Algorithm for Real Time Images," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 2, pp. 1327–1336, 2020, doi: 10.11591/ijece.v10i2.pp1327-1336.
[24] D. Herbadji, "A New Image Encryption Scheme Using an Enhanced Logistic Map," *International Conference on Applied Smart Systems (ICASS'2018)*, pp. 1-6, 2018.
[25] G. Cheng, C. Wang, and H. Chen., "A Novel Colour Image Encryption Algorithm Based on Hyperchaotic System and Permutation-Diffusion Architecture," *International Journal of Bifurcation and Chaos*, vol. 29, no. 9, pp. 1–17, 2019.
[26] S. Fadhel, M. Shafry, and O. Farook, "Chaos Image Encryption Methods: A Survey Study," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 6, no. 1, pp. 99–104, 2017.
[27] S. D. Putra, "Power Analysis Attack against Encryption Devices: A Comprehensive Analysis of AES, DES, and BC3," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 17, no. 3, pp. 1282–1289, 2019.
[28] X. Wang and C. Tu, "A Chaos-Based Medical Image Encryption Method," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 19, no. 3, pp. 1316–1324, 2020.
[29] E. L. Mohaisen and R. S. Mohammed, "Stream Cipher Based On Chaotic Maps," *IEEE CAS*, 2020.
[30] Bensa Ida, "A Practical Test for Noisy Chaotic Dynamics," *ELSEVIER*, 2015.

BIOGRAPHIES OF AUTHORS

Rana Saad Mohammed received a B.Sc. in computer science from Mustansiriyah University, Baghdad, Iraq in 2006, M.Sc. in computer science - data security from University of Technology, Baghdad, Iraq in 2008, and Ph.D. in computer science from Babylon University, Hilla, Iraq in 2015. I'm currently an assistant professor in computer science department at the Education College in the University of Mustansiriyah. She was a member of IEEE. My main research interests in Information security, stream cipher, block cipher, pseudorandom number generator, voice encryption, chaos theory, steganography, data mining privacy, Security of Internet of things (IOT), and Cloud computing.

Khalid Kadhim Jabbar, Baghdad-Iraq, B.Sc. in Computer Science - Baghdad University - Baghdad-Iraq, 2001. HI-DUP in Data Security - ICCI, Baghdad-Iraq, 2002. M.Sc. in The Science of Software Engineering - ICCI, 2012. Asst. Pro. In Computer Science Department-Collage of Education-Mustansiriyah University. Scientific reviewer, Designated Reviewer, Editorial Board Member, and member of the Technical Program Committee in local and international journals (Scopus). My scientific interests include the majors of: Information/ Data Security, Image Processing, Chaotic System, Steganography, AI, Software Engineering, Digital Watermarking, and Tamper Detection.

Huseein Abed, Baghdad-Iraq. B.Sc. Alsadiq university.HI-DUP in web site design - University of technology, 2011. M.Sc. University of technology, 2015. Assist lecturer- Mustansiriyah University - Programing and development in computer center. Skills in project development, programming language such as: Mat lab, Visual C++, VP.net, ASP.net, Visual Basic.