

## Enhanced Vigenere encryption technique for color images acting at the pixel level

Abdelhakim Chemlal<sup>1</sup>, Hassan Tabti<sup>2</sup>, Hamid El Bourakkadi<sup>1</sup>, Hicham Rrghout<sup>1</sup>, Abdellatif Jarjar<sup>1</sup>, Abdelhamid Benazzi<sup>1</sup>

<sup>1</sup>MATSI Laboratory, High School of Technology, Mohammed First University, Oujda, Morocco

<sup>2</sup>LSIA Laboratory, Faculty of Science and Technology, Sidi Mohamed Ben Abdellah University, Fez, Morocco

### Article Info

#### Article history:

Received Mar 1, 2024

Revised Jul 24, 2024

Accepted Aug 14, 2024

#### Keywords:

Broadcast function

Chaotic map

Diffusion

Enhanced Vigenere

S-Box

### ABSTRACT

The pixel unit is an essential component of many encryption schemes. In the beginning two substitution tables, separately constructed from chaotic maps namely, the logistic map, slanted tent map, and the AJ map, which has a very high Lyapunov exponent and is very sensitive to start factors, are used to make modifications at the pixel level. These S-Boxes have a maximum period and are produced from several linear congruential generators. This approach uses newly developed confusion and diffusion functions connected to the recently built substitution tables to perform a refined Vigenere strategy. The purpose of this chaining is to defend the system from differential assaults. Extensive simulations on a variety of image formats and sizes confirm our process's robustness against identified dangers.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



### Corresponding Author:

Hassan Tabti

LSIA Laboratory, Faculty of Science and Technology, Sidi Mohamed Ben Abdellah University

Fez-30000, Morocco

Email: hassan.tabti1@usmba.ac.ma

## 1. INTRODUCTION

Information security investigations during network transmission are now a well-known field of study. One important participant in this context is encryption technology, which uses both symmetric and asymmetric encryption techniques in the field of cryptography [1]. Symmetric encryption, which is known for its effectiveness, increased security, and quick encryption speed thanks to a big key, depends on the safe storage of the ciphering key. This algorithm entails minimal computation, ensuring a high level of protection and swift encryption when employing an extended key. This method's data transfer security depends on keeping the encryption key safe. Conversely, asymmetric ciphering is appropriate for limited data encryption, such as passwords, even if it offers great security due to its large encryption and decryption time. In this instance, the key and the algorithm are both necessary for data transfer security. Kirchhoff's principle states that the ciphering key, not the algorithm, has a major role in the key system's security. Many image ciphering techniques use symmetry theory ideas to overcome these difficulties, and this study takes a similar approach based on similar kinds of algorithms.

A number of picture encryption approaches have been successfully cracked, despite researchers' ongoing efforts to strengthen the security of ciphering methods [2], [3]. Numerous academics have resorted to multi-round encryption techniques in order to increase security [4], [5]. But there is a major time penalty associated with this strategy. Certain authors have suggested encryption techniques that target particular aspects of the images [6], [7]. A ciphering architecture for medical images based on an optimal game theory method was introduced by Ping *et al.* [7], demonstrating flexibility and dependability in protecting medical

images from attacks. A hybrid image encryption architecture using affine and substitution algorithms with an expanded logistic map was proposed in [8], [9], and it showed good security performance in experiments. Based on these discoveries, we suggest using current technology to recognize the area of the face contour while encrypting photographs of people, enabling the specific encryption of these areas. An extra layer of encryption is applied to the complete image once the face portion has been encrypted. This method demonstrates higher encryption effectiveness as compared to conventional one-round encryption. Private features like the face are unrecognizable and unrecoverable even in the face of algorithmic attacks. These methods entail faster and shorter times for both data encryption and decryption than multi-round encryption approaches.

Researchers have methodically examined the properties of pseudo randomness and sensitivity to beginning values as chaos theory develops [10], [11]. Diverse domains, including genetic algorithms [12]–[16], 3D chaotic maps [16], one-time keys [17], compressive sensing [18], [19], and perceptron-like networks [20], have given rise to a variety of chaotic picture encryption schemes. In order to handle problems with dynamic degradation, Qobbi *et al.* [21] created an encryption architecture in their work that uses replacement-diffusion processes with standard and logistic maps. Akraam *et al.* [22] showed how algorithmic security could be jeopardized by using Arnold's chaotic sequences as keys. Wang and Liu [23] addressed worries regarding dynamic degradation in chaotic cryptography by presenting theoretical proof of the security of chaotic flowchart cipher systems, demonstrating that algorithmic security is not endangered by chaotic sequences. Additionally, by integrating space, domains, and time, Kang *et al.* [24] suggested an encryption architecture that enhances security by combining a logistic map with a novel reality-preserving parameter of fractional transform. Zhang and Wei [25] proposed a novel architecture in their paper for encrypting color images. This architecture combines the Lorenz chaotic system with DNA computing, creating a chaotic system with spatial and temporal parameters [26], [27].

Most traditional systems are still susceptible to statistical and frequency attacks. Additionally, if there is no chaining between the cipher and the subsequent plain blocks, most systems are also susceptible to differential attacks. Moreover, this method works especially well for encrypting big datasets with a lot of redundancy and significant correlations. Our contribution is to elaborate a procedure for creating fresh substitution tables with varying sizes, achieved through the utilization of multiple (LCG). Furthermore, it will introduce a genetic crossover method tailored for encrypting extensive datasets. This method incorporates an enhanced Vigenere technique version, integrating the newly generated substitution tables to bolster the preservation of the plain image's integrity. The recommended encryption architecture significantly reduces encryption time compared to those suggested by other scholars. The control parameters and keys are generated rapidly from numerous linear congruent generators based on chaotic maps. The algorithm exhibits outstanding security, as evidenced by the simulation, security, and comparative results with other algorithms, showcasing its ability to withstand typical attacks.

The present paper is divided into separate pieces for the remainder of it. These comprise an introduction to the theoretical background, introducing the basis of chaotic sequences and genetic operators procedures; an outline of the suggested methodology, explaining the details of the ciphering and their reverse procedures; an experimental results section, presenting research results, comparisons, and discussions with their counterpart methods; and a conclusion section recapitulating the results.

## 2. METHOD

Our technique consists of developing two large substitution tables. These two tables will be used to implement a new enhancement of the traditional Vigenere algorithm. This technique applies confusion functions attached to SBoxes and is composed from the following axes.

### 2.1. Selected chaotic sequences

The three chaotic maps that we have selected to guarantee the effective operation of our system are the skew tent map, the AJ map, and the logistics cards. These are the cards that are used the most in the cryptography domain. This choice was made because of their outstanding reactivity to beginning conditions and ease of configuration.

#### 2.1.1. The logistic map

This map ( $U_n$ ) [28] is a sequence expressed by a simple second-degree polynomial defined recurrently by (1). It has high sensitivity to initial conditions and is easy to configure in any cryptosystem. This characteristic is confirmed by the calculated value of the Lyapunov exponent.

$$\begin{cases} u_0 \in ]0,5 \ 1[ \quad , \quad \mu \in [3,75 \ 4] \\ u_{n+1} = \mu u_n (1 - u_n) \end{cases} \quad (1)$$

### 2.1.2. The skewed tent map (SKTM)

This map ( $V_n$ ) [29] is a sequence expressed by a simple second-degree polynomial defined recurrently by (2). It has high sensitivity to initial conditions and is easy to configure in any cryptosystem. This characteristic is confirmed by the calculated value of the Lyapunov exponent.

$$\begin{cases} v_0 \in ]0 \ 1[ \quad p \in ]0.5 \ 1[ \\ v_{n+1} = \begin{cases} (p)^{-1} v_n & \text{if } 0 < v_n < p \\ (1-p)^{-1} (1 - v_n) & \text{if } p < v_n < 1 \end{cases} \end{cases} \quad (2)$$

### 2.1.3. A.J. map

This map ( $W_n$ ) [28] is a sequence expressed by a first-degree polynomial defined recurrently by (3). It has high sensitivity to initial conditions and is easy to configure in any cryptosystem. This characteristic is confirmed by the calculated value of the Lyapunov exponent.

$$\begin{cases} w_0 \in [(1+p)^{-1} \ p \ (1+p)^{-1}] \quad p \in [1.47 \ \varphi] \\ f(w_n) = w_{n+1} \begin{cases} p^2 w_n & \text{if } 0 \leq w_n \leq (1+p)^{-1} \\ p - p w_n & \text{if } (1+p)^{-1} \leq w_n \leq 1 \end{cases} \end{cases} \quad (3)$$

The hybridization employment of three chaotic maps has the advantage of deriving all essential parameters required for our innovative architecture efficiency and effectiveness. Based on the construction of multiple substitution tables using pseudorandom linear congruential generators and a genetic crossover acting at the bit level under the control of a crossover table constructed from the chaotic maps used. Our method is described in the next sub-sections.

## 2.2. Subkeys design

It is necessary to generate many pseudo-random vectors in order for the encryption and decryption process to function properly. Utilizing these vectors, an algorithm is created that can handle any known assault. The following stages are taken in this construction.

### 2.2.1. Constructing ciphering parameters

In this phase, an advanced Vigenere cipher is applied at the pixel level, requiring the creation of specific parameters to ensure robust encryption. These include i) XT confusion and diffusion process tables to manipulate pixel data and enhance security, ii) BT binary tables for controlling and guiding the encryption process, and iii) WS1 and WS2 substitution tables for effective data substitution and transformation. Together, these components work in concert to bolster the cipher's effectiveness and improve overall data protection.

### 2.2.2. (XT) Table creation

The purpose of the table (XT) of size (3 nm; 5) with coefficients in ( $G_{256}$ ) is to function as diffusion and aliasing on the level of the original image pixels. Algorithm 1 describes how to build such a table. Every individual column within the table (XT) signifies an independent pseudo-random vector distinct from the remaining vectors.

Algorithm 1. (XT) design

1. For  $i = 1$  to  $3 \text{ nm}$
2.  $XT(i; 1) = \text{mod}(E(|u(i) - v(i) * w(i)| * 10^{12}), 252)) + 3$
3.  $XT(i; 2) = \text{mod}(E((u(i) + w(i)) * 10^{10}), 254)) + 1$
4.  $XT(i; 3) = \text{mod}(E(\text{Sup}(u(i); v(i)) * 10^{11}), 254)) + 1$
5.  $XT(i; 4) = \text{mod}(E\left(\left(\frac{u(i)+2*v(i)+w(i)}{4} * 10^{11}\right), 253\right) + 2$
6.  $XT(i; 5) = \text{mod}(E((v(i) * 10^6 + w(i) * 10^7), 253)) + 2 : \text{Next } i$

### 2.2.3. (BT) Binary tables design

The table (XT), sized (3 nm; 5) and composed of coefficients in  $G_{256}G\{256\}G_{256}$ , is designed to facilitate diffusion and aliasing at the pixel level of the original image, enhancing the encryption process. Algorithm 1 outlines the method for constructing this table, ensuring that each column within (XT) represents a unique pseudo-random vector, distinct from the others. This approach guarantees that the diffusion and aliasing effects are applied effectively, contributing to the overall security of image encryption.

**Algorithm 2. (BT) design**

- |                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> <li>1. For <math>i = 1</math> to <math>3nm</math><br/>  // First column</li> <li>2. if <math>u(i) &gt; \text{Inf}(v(i); w(i))</math> Then</li> <li>3. <math>BT(i; 1) = 0</math> Else <math>BT(i; 1) = 1</math> : End if</li> </ol> | // Second column <ol style="list-style-type: none"> <li>4. if <math>XT(i; 1) \geq XT(i; 5)</math> Then</li> <li>5. <math>BT(i; 2) = 0</math> Else <math>BT(i; 2) = 1</math></li> <li>6. end if : Next <math>i</math></li> </ol> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**2.2.4. Substitution matrix computation**

To enhance the design of new Boxes, we will offer key mathematical reminders that leverage various linear congruence generators. These generators play a crucial role in producing pseudo-random sequences that can be utilized for constructing effective cryptographic tables. By applying these reminders, the design process will benefit from improved randomness and security in the generated boxes.

**2.2.5. Mathematics reminder**

A linear congruence generator is a real sequence defined by three initial parameters and given by (4):

$$LCG(s_0; a; b; m) \begin{cases} s_0 = u \text{ (initial condition)} \\ s_{n+1} = \text{mod}(as_n + b; m) \end{cases} \quad (4)$$

$(s_0; a; b; m)$  is called a linear congruential generator. A good (LCG) is a periodic sequence, with a maximum period equal to  $(m)$ . This period is reached under the conditions of the theorem of Hull and Dobell which offers a necessary and sufficient condition for a (LCG) to present the pseudo-random attribute.

**a. Hull-Dobell theorem (1962)**

All linear congruence generators (LCGs) that adhere to the previously outlined theorem demonstrate maximum periodicity, as established by Algorithm 3. This algorithm details the conditions under which LCGs achieve their longest possible sequence length before repeating. By following these guidelines, one can ensure that the LCGs used will provide optimal pseudo-randomness and period duration for cryptographic applications.

**Algorithm 3. Hull and Dobell terms**

1.  $s_0$  unspecified integer in  $\llbracket 0 ; m - 1 \rrbracket$
2.  $b \wedge m = 1; a \wedge m = 1$
3. if there exists a prime divisor  $p$  of  $m$ , then
  - a.  $p$  divides the quantity  $a - 1$ .
  - b. if  $4/m$  then  $4/a - 1$

**b. Particular case ( $m = 2^k$ )**

A very interesting particular case in the field of color image cryptography is  $m = 2^k$ .

$$LCG(s_0; a; b; 2^k): \begin{cases} s_0 = u \in G_{2^k} \\ s_{n+1} = \text{mod}(as_n + b; 2^k) \end{cases} \quad (5)$$

This generator has a period of  $2^k - 1$ , if and only if:

$$LCG(s_0; a; b; 2^k): \begin{cases} s_0 = u \in G_{2^k} \\ b = \text{mod}(2h + 1; 2^k) \\ a = \text{mod}(4k + 1; 2^k) \end{cases} \quad (6)$$

These generators are used to create two substitution tables (WS1) and (WS2), whose sizes (256; 256).

**2.2.6. (LG1) Parameter table design**

An (LCG) is recurrently determined by the system (7), satisfying the criteria outlined in the Hull and Dobell theorem.

$$(LCG): \begin{cases} s_0 = u \\ s_{n+1} = \text{mod}(as_n + b; 256) \\ s_0 \in \llbracket 0 ; 255, \rrbracket \\ b \equiv 1 [2] \quad b = 2k + 1 \\ a \equiv 1 [4], \quad a = 4k + 1 \end{cases} \quad (7)$$

The three parameters ( $s_0, a, b$ ) of (LCG) used in our algorithm for the development of two substitution tables (SW1) and (SW2) will be retained in the array (LG1) of size (3;3 nm), by the subsequent steps: i) the initial value of the generator in ( $G_{256}$ ), denoted as ( $s_0$ ), will be present in the first line as a pseudo-random seed; ii) the second line will store the multiplier parameter values ( $a = 4k + 1$ ); and iii) the third one will store the bias parameter values ( $b = 2k + 1$ ). Figure 1 depict an example to clarify the distribution of the (LCG).

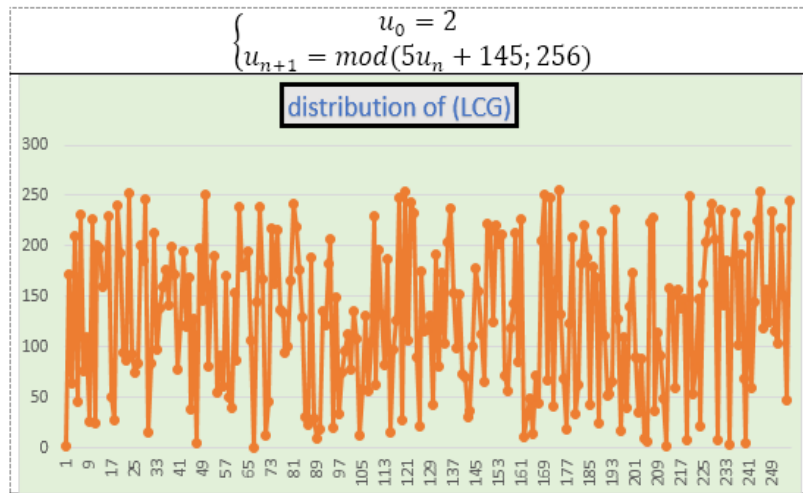


Figure 1. Example of the (LCG) distribution

These generators, which satisfy the Hull and Dobell theorem, produce non-linear permutations of size 256, stored in two tables (WS1) and (WS2) each of size (256×256), and used as S-boxes. First, we will place the parameters of each (LCG) in a table (LG1) of size (3×256).

- The first row contains the values of the parameters ( $s$ ) which constitute the seed of the generator.
- The second row contains the values of the multiplier ( $a$ ).
- The third row stores the values of the parameter ( $b$ ).
- The table (LG1) utilized for the development of the S-box is given by the Algorithm 4.

Algorithm 4. (LG1) Table design

1. For  $j = 1$  to  $3nm$
2. If  $BT(j; 2) = 0$  Then
3.  $LG1(1, j) = XT(j; 3)$
4.  $LG1(2, j) = mod(1 + 4 * XT(j; 2), 256)$
5.  $LG1(3, j) = mod(1 + 2 * XT(j; 3), 256)$
6. Else
7.  $LG1(1, j) = XT(j; 2)$
8.  $LG1(2, j) = mod(1 + 4 * XT(j; 1), 256)$
9.  $LG1(3, j) = mod(1 + 2 * XT(j; 4), 256)$
10. Next  $j$

The guided structure by pseudorandom vectors, exhibits high sensitivity to alterations in any constituent of a private key; this reinforces the robustness of a system. To give the pseudo-random aspect to the S-Boxes construction; we are going to use the tables (TC1) of size (256;4). The S-Box (WS1) is entirely governed by ( $G_{256}$ ) rearrangements of the table (TC1) of size (256,4). An example is given in Figure 2.

Rank	1	2	3	4	5	6
Row	2	4	1	3	6	5
$s$	1	3	10	11	1	2
$a$	2	5	3	13	11	5
$b$	2	13	5	4	9	6

Figure 2. Example of (LG1) in G16

a. (TC1) Table creation

The table (TC1) of size (4×256), is constructed for the selection of generators for the development of the two S-Boxes (SW1).

- The first row of table (TC1) is the permutation (P1) obtained by a descending sort of the first 256 values of the vector (XT (:4)), used to select the row where the chosen generator will be placed.
  - The second line of the table (TC1) is the permutation (P2) obtained by an ascending sorting of the first 256 values of the vector (XT (:2)), used to select the index of the value of the (s).
  - The third line of the table (TC1) is the permutation (P3) obtained by an ascending sorting of the first 256 values of the vector (XT (:2)), used to select the index of the value of the multiplier (a).
  - The fourth line of the table (TC1) is the permutation (P4) obtained by an increasing sorting of the first 256 values of the vector (XT (:3)), used to select the index of the value of the parameter (b).
- An example of (TC1) table in G16 is depicted in Figure 3.

<b>Rank</b>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
<b>Row</b>	7	2	5	12	11	8	9	3	6	13	15	10	8	1	16	4
<b>s</b>	0	2	5	10	2	3	5	7	8	10	6	9	13	10	14	5
<b>a</b>	1	5	5	9	13	5	1	13	9	5	13	1	9	5	9	13
<b>b</b>	3	15	11	3	5	7	9	7	3	5	7	11	13	9	11	1

Figure 3. Example of (TC1) table in G16

b. (WS1) Table computation

The construction of the substitution matrix (WS1) is guided by tables (TC1) and (LG1), with the process meticulously detailed in Algorithm 5. These tables provide the necessary parameters and controls for generating WS1, ensuring its effectiveness in cryptographic applications. By following the steps outlined in Algorithm 5, one can accurately construct WS1 to enhance the overall security and functionality of the encryption scheme. An example about how to compute the (WS1) table is depicted in Figure 4.

Algorithm 5. (WS1) S-Boxes computation

1. For  $j = 1$  to 256
2.  $a = LG1(TC1(j,2))$
3.  $s = LG1(TC1(j,1))$
4.  $b = LG1(TC1(j,3))$
5. For  $i = 1$  to 256
6.  $x = mod(a * s + b; 256)$
7.  $WS1(TC1(j,4),i) = x$
8.  $s = x : Next i, j$

<b>Rank</b>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\begin{cases} u_0 = 10 \\ u_{n+1} = mod(5 * u_n + 5; 16) \end{cases}$	10	7	8	13	6	3	4	9	2	15	0	5	14	11	12	1
$\begin{cases} u_0 = 0 \\ u_{n+1} = mod(5 * u_n + 15; 16) \end{cases}$	0	1	4	3	14	5	8	7	2	9	12	11	6	13	0	15
$\begin{cases} u_0 = 6 \\ u_{n+1} = mod(9 * u_n + 3; 16) \end{cases}$	6	9	4	7	2	5	0	3	14	1	12	15	10	13	8	11
$\begin{cases} u_0 = 5 \\ u_{n+1} = mod(13 * u_n + 13; 16) \end{cases}$	5	14	3	4	1	10	15	0	13	6	11	12	9	2	7	8
$\begin{cases} u_0 = 0 \\ u_{n+1} = mod(5 * u_n + 15; 16) \end{cases}$	0	15	10	1	4	3	14	5	8	7	2	9	12	11	6	13
$\begin{cases} u_0 = 2 \\ u_{n+1} = mod(13 * u_n + 7; 16) \end{cases}$	2	1	4	11	6	5	8	15	10	9	12	3	14	13	0	7

Figure 4. Example of (WS1) computation

c. (TC2) Design

The table (TC2) of size (4×256), is constructed for the selection of generators for the development of the two S-Boxes (SW2).

- The first row of table (TC2) is the permutation (Q1) obtained by a descending sort of the first 256 values of the vector (XT (:3)), used to select the row where the chosen generator will be placed.
- The second line of the table (TC2) is the permutation (Q2) obtained by an ascending sorting of the first 256 values of the vector (XT (:1)), used to select the index of the value of the ( ).

- The third line of the table (TC2) is the permutation (Q3) obtained by an ascending sorting of the first 256 values of the vector (XT (:4)), used to select the index of the value of the multiplier ( $a$ ).
- The fourth line of the table (TC2) is the permutation (Q4) obtained by an increasing sorting of the first 256 values of the vector (XT (:5)), used to select the index of the value of the parameter ( $b$ ).

#### Algorithm 6. (TC2) Computation

1. For  $i = 1$  to 256
2.  $TC2(i, 1) = Q1(i)$
3.  $TC2(i, 2) = Q2(i)$
4.  $TC2(i, 3) = Q3(i)$
5.  $TC2(i, 4) = Q4(i)$
6. Next  $i$

#### d. (WS2) Computation

Under the control of tables (TC2) and (LG1) the construction of the substitution matrix (WS2) is given by Algorithm 7. The construction of the table (WS2) is similar to that of (WS1) by considering the tables (TC) and (LG1).

#### Algorithm 7. (WS2) S-Box design

1. For  $i = 1$  to 256
2.  $h = LG1(TC2(i, 1))$ ;
3.  $c = LG1(TC2(i, 2))$ ;
4.  $d = LG1(TC1(i, 3))$
5. or  $j = 1$  to 256
6.  $y = \text{mod}(c * h + d; 256)$
7.  $WS2(TC2(i, 4), j) = y + 1$
8.  $h = y$ ; : Next  $j, i$

#### e. Pixel transcription based on S-boxes

The two substitution tables will be employed together to transform the pixel values of the original image using the (VG) function. This transformation process, which converts the  $i^{\text{th}}$  pixel  $X(i)X(i)$  into  $Y(i)Y(i)$ , is precisely defined by Algorithm 8. By applying Algorithm 8, the (VG) function ensures that each pixel's value is accurately and securely altered, leveraging the combined effects of the substitution tables.

#### Algorithm 8. $X(i)$ pixel encryption

```

VG(X(i)) = Y(i)
if BT(i; 2) = 0 then
    : Y(i) = WS1(XT(i; 2), WS2(XT(i; 3), X(i) ⊕ XT(i; 4))) ⊕ XT(i; 1)
else
    : Y(i) = WS2(XT(i; 3), WS1(XT(i; 1), X(i) ⊕ XT(i; 5))) ⊕ XT(i; 2) :
end if

```

This stage of the encryption process uses two nested S-Boxes and the decision vector  $BT(i)$  to increase the complexity of the substitution and confusion functions. A unique ciphering and distribution expression will arise from even a little modification to a private key parameter, producing a unique cipher picture. While  $Y(i)$  returns the transformation of the pixel  $X(i)$ .

### 2.3. Preparation of the image for ciphering

Prior to encryption, it is necessary to prepare any initial image using the following steps as the foundation for this technique. The steps are plain image vectorization and initialization constant design. The following are the details of each step.

#### 2.3.1. Plain image vectorization


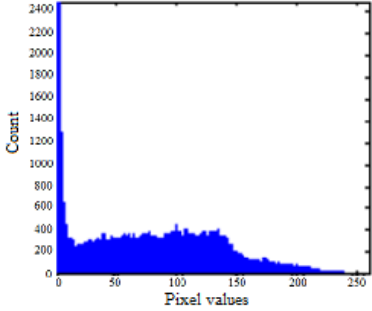
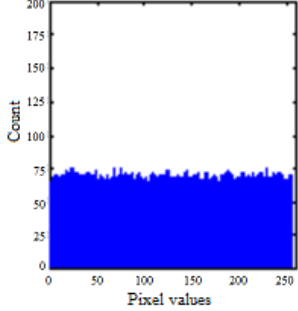
After extraction of the three red, green, and (RGB) color channels and their transformation into size vectors ( $Vr$ ), ( $Vg$ ), ( $Vb$ ), each of dimension (1, nm), a concatenation is performed, leading to confusion with the vector ( $V3$ ). This process generates then vector  $X(x_1, x_2, \dots, x_{3nm})$ , as outlined in Algorithm 9.

#### Algorithm 9. Transitioning to vector (X)

- a. for  $i = 2$  to  $nm$
- b. If  $BT(i; 1) = 0$  Then
- c.  $X(3i - 2) = Vb(i) \oplus XT((3i - 2); 4)$
- d.  $X(3i - 1) = Vr(i) \oplus XT((3i - 1); 3)$
- e.  $X(3i) = Vg(i) \oplus XT((3i); 1)$
- f. Else
- g.  $X(3i - 2) = Vb(i) \oplus XT((3i - 2); 1)$
- h.  $X(3i - 1) = Vr(i) \oplus XT((3i - 1); 4)$
- i.  $X(3i) = Vg(i) \oplus XT((3i); 3)$
- j. End if: Next  $i$

The decision vector (CR) governs this transition to vector notation. While the high correlation between pixels is somewhat lessened in this stage, the encryption is still strong enough to withstand statistical and brute force attacks as depicted in Table 1. This is demonstrated by the figure that follows. To defend our system from differential assaults, we will employ the encryption functions, the broadcast functions, and the two S-Boxes.

Table 1. First stage of image ciphering process

Image	Histograms		Entropy	
	Plain	Cipher	Plain	Cipher
			4.5687	7.9996

**2.3.2. Initialization constant design**

Using Algorithm 10, a constant (IV) is computed from the plain image in order to change the value of the seed pixel and start the ciphering process. It is noteworthy to note that the calculated constant is closely related to the vector (BT (:2)) and the unprocessed image. Any small change to the plain image or any of the private key’s parameters will produce a unique constant and, in turn, a new cipher picture. This demonstrates how sensitively and entirely dependent the encryption procedure is on the input variables.

Algorithm 10. First initialization value calculation

1.  $IV = 0$
2. For  $i = 2$  to  $3nm$
3. If  $BT(i; 2) = 0$  Then
4.  $IV = X(i) \oplus IV \oplus XT(i; 3)$
5. Else:  $IV = X(i) \oplus IV \oplus XT(i; 5)$
6. Next  $i$

**2.4. Encryption system process**

The computed constant serves solely to alter the seed pixel’s value and initiate the ciphering process. Figure 5 depicts the improved encryption procedure involving S-boxes and P-boxes. In (8) provides the advanced broadcasting function  $\Pi(Y(i))$  for this encryption stage, which is based on nested S-Boxes (WS1) and (WS2).

$$\Pi(Y(i)) = WS1(XT(i; 1), WS2(XT(i; 2); Y(i) \oplus X(i + 1))) \tag{8}$$

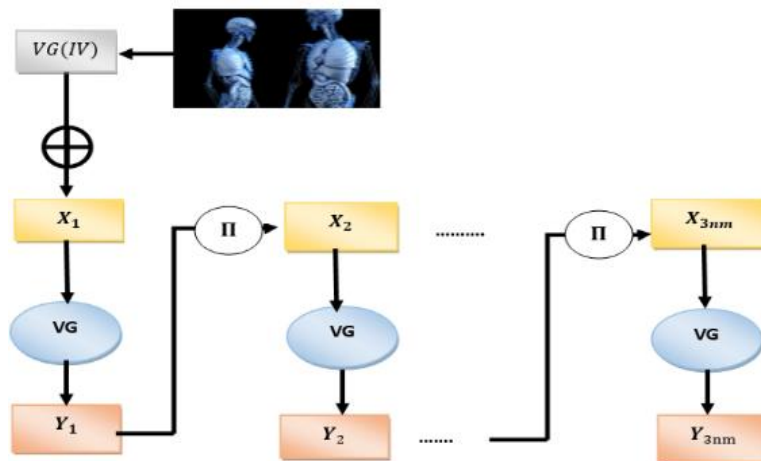


Figure 5. First stage of image ciphering process

This operation can be given by Algorithm 11. The obtained vector (Y) that signifies the encrypted image is a set of coordinates corresponding to nucleotides. The first-round encryption time is given in Table 1. This operation can be given by Algorithm 12. The resulting vector (Y) representing the encrypted image is a set of nucleotide coordinates.



**Algorithm 11. First stage of image ciphering algorithm***First pixel encryption*

1.  $Y(1) = VG(XT(1;1) \oplus X(1)) \oplus VG(IV)$
2.  $\Pi(X(i)) = VG(XT(i;1) \oplus Y(i-1)) \oplus X(i)$
3. *If*  $BT(i;2) = 0$  *Then*
4.  $Y(i) = VG(\Pi(X(i))) \oplus XT(i;3)$
5. *else*
6.  $Y(i) = VG(\Pi(X(i))) \oplus XT(i;2)$ : *Next*  $i$

**Algorithm 12. First stage of image ciphering algorithm***First pixel encryption*

1.  $Y(1) = VG(XT(1;1) \oplus X(1)) \oplus VG(IV)$
2. *For*  $i = 2$  *to*  $3nm$  // *Next Pixel Encryption*
3.  $\Pi(X(i)) = VG(XT(i;1) \oplus Y(i-1)) \oplus X(i)$
4. *If*  $BT(i;2) = 0$  *Then*
5.  $Y(i) = VG(\Pi(X(i))) \oplus XT(i;3)$
6. *else*
7.  $Y(i) = VG(\Pi(X(i))) \oplus XT(i;2)$ : *next*  $i$

**2.5. Decryption procedure**

In our method, we use a broadcast implementation as part of a symmetric encryption system. Consequently, as part of the decryption process, decryption functions are used, starting with the last block. Every operation used in our method is reversible, which ensures that a decryption function is available. The various stages in the decryption procedure are described: i) vectorization of the encrypted image; ii) initial round of decryption for the reciprocal generation of the Vigenere matrix; and iii) reciprocal of Vigenere matrix construction.

**2.5.1. Vigenere matrix reciprocal**

The inverse Vigenere transformation needs to be applied using Algorithm 13. We can derive the substitution traditional function reciprocal shown in (9), by applying the same logic as Vigenere's conventional technique.

$$\text{if } z = VG(y, x) \text{ Then } x = GV(y, z) \quad (9)$$

**Algorithm 13. Vigenere reciprocal***for*  $i = 1$  *to* 256*for*  $i = 1$  *to* 256

$$GV(i, VG(i, j)) = j ; DV((i, VD(i, j)) = j ;$$

*Next*  $j, i$ **2.5.2. Vigenere's reverse formula**

Algorithm 14 outlines the Vigenere transformation, a method used to encode plaintext into ciphertext using a designated key. Conversely, it provides the means to reverse this transformation, decrypting ciphertext back into its original plaintext form. This inverse process is integral for securely recovering plaintext from encrypted data, ensuring confidentiality and data integrity in cryptographic applications. This algorithm is fundamental in comprehending the bidirectional nature of the Vigenere cipher, facilitating both encryption and decryption operations effectively.

**Algorithm 14. Vigenere reciprocal**

$$GV(Y(i)) = X(i)$$

*if*  $BT(i;2) = 0$  *then*:

$$X(i) = SW2(XT(i;3), SW1(XT(i;2), Y(i) \oplus XT(i;1))) \oplus XT(i;4) :$$

*else*:

$$X(i) = SW1(XT(i;1), SW2(XT(i;3), Y(i) \oplus XT(i;2))) \oplus XT(i;5) :$$

*end if***2.5.3. Reverse diffusion**

The equation (10) specifies the inverse of the diffusion function used in our scheme, which plays a central role in spreading the input data to enhance security. This function ensures that changes in any part of the input are propagated throughout the output, which is crucial for maintaining robustness against cryptographic attacks. Understanding and implementing (10) is essential for efficiently reconstructing the original data from its diffused form, thereby completing the encryption-decryption cycle with accuracy. Integrating the inverse of the diffusion function is essential for achieving reliable data recovery and maintaining the integrity of sensitive information in our cryptographic framework.

$$\Pi^{-1}(X'(k)) = GV(CL(k), DV(KR(k), X'(k))) \oplus X(k-1) \quad (10)$$

### 3. RESULTS AND DISCUSSION

In this section, a significant number of randomly selected images from a large database will be the subject of an evaluation of our innovative algorithm. We will compare the performance results obtained with those of alternative algorithms to highlight the effectiveness and superiority of our approach. This evaluation aims to demonstrate the effectiveness of the algorithm in handling diverse image data and to show its competitive advantage over existing methods.

#### 3.1. Key-space analysis

In cryptanalysis, a brute force attack involves systematically testing every possible combination in order to uncover the encryption key. The feasibility of such attacks decreases significantly for larger encryption keys, as the sheer number of possible combinations makes exhaustive testing impractical. In our algorithm, the secret key is designed to be larger than  $2^{128}$ , ensuring robust resistance to brute force attacks, as noted in references [29]. In this cryptographic approach, a secret key is generated from the combined parameters of three widely used chaotic maps, with a total of six parameters encoded in 32 bits each, resulting in a robust key size of  $(2^{6 \times 32}) = (2^{192}) \gg (2^{100})$ .

#### 3.2. Analysis of the sensitivity of a secret key

Within our approach, each chaotic map exhibits a high sensitivity to initial conditions, ensuring that even small changes in parameters during key regeneration result in completely different keys and divergent chaotic vectors. This sensitivity plays a crucial role in our novel technology, which is visually illustrated in Figure 6, which shows the distinct and different results produced by different initial conditions and parameter perturbations within the chaotic maps. This feature underscores the robustness and unpredictability of our cryptographic approach, enhancing security by generating unique keys that are resistant to replication or prediction. We observe that a disturbance of a single variable of the order of  $(10^{-7})$  is not sufficient to precisely reconstruct the plaintext image.

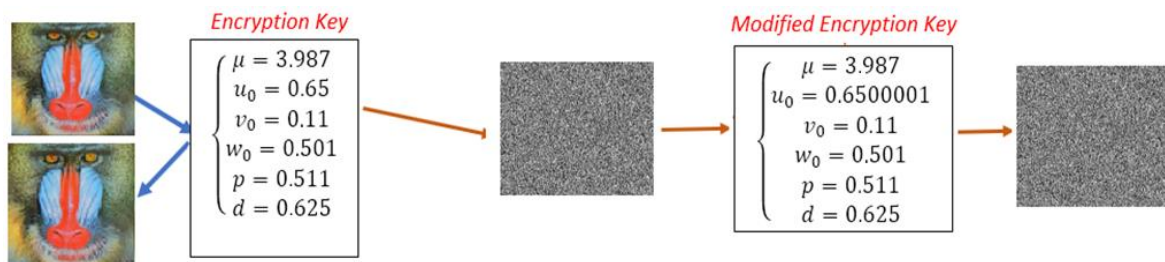


Figure 6. Sensitivity of ciphering key

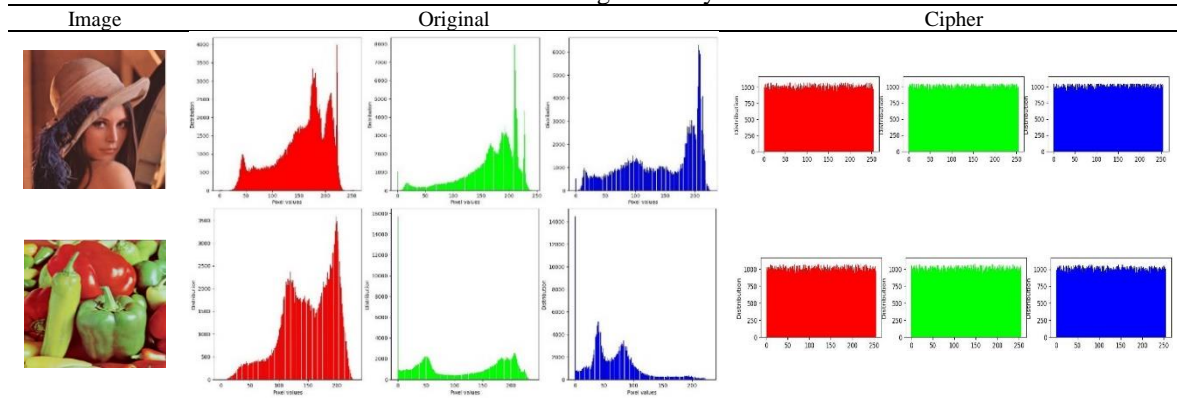
#### 3.3. Statistical attack security

To validate the resilience of our novel encryption technique for medical and color images against statistical attacks, extensive testing has been conducted, focusing on several notable experiments. These tests were designed to assess the algorithm's ability to withstand various statistical analyses commonly used in cryptanalysis and image processing. The results highlight key findings that demonstrate the effectiveness of our encryption method in maintaining image security and integrity under rigorous examination.

##### 3.3.1. Analysis of histograms

The flowchart illustrates the frequency distribution of pixels sharing identical grey levels, where each vertical bar along the x-axis represents occurrences ranging from zero to 255 levels within the image. From a cryptographic standpoint, analyzing the color distribution in encrypted images is critical as it can unveil insights into the original content. Conversely, a uniformly distributed histogram in encrypted images suggests a robust encryption process that obscures any identifiable patterns from the original data. Table 2 displays the simulation outcomes of our system, providing empirical evidence of its performance and effectiveness in protecting image integrity and confidentiality. This study demonstrates that our method consistently generates encrypted images with smoothed histograms across various test scenarios. The uniformity observed in these histograms provides strong defense against potential attacks that aim to exploit histogram characteristics for decryption or analysis purposes.

Table 2. Histogram analysis





**3.3.2. Entropy analysis**

In (11) defines the entropy linked to the pixel distribution within an image, capturing the degree of uncertainty or randomness present in its data. Table 3 showcases the entropy measurements obtained from images processed using our encryption technique, highlighting how our approach affects the information content and distribution uniformity within encrypted images. These entropy values serve as quantitative indicators of the cryptographic strength and effectiveness of our method in preserving image confidentiality while minimizing predictability and vulnerability to statistical attacks.

$$\begin{cases} H(MC) = \frac{1}{t} \sum_{i=1}^t -\pi(i) \log_2(\pi(i)) \\ \pi(i) \text{ represents the probability of the occurrence of level (i) in the original image.} \end{cases} \tag{11}$$

Table 3. Correlation, normalized pixel changes rate (NPCR), uniformity of average change intensity (UACI), and entropy analysis for Lena and Peppers encrypted images

Image N°	Correlation			NPCR	UACI	Entropy
	Horizontal	Vertical	Diagonal			
	0.0087	-0.0068	7.7766e-04	99.61	33.42	7.9997
	0.0034	-1.0498e-06	0.0026	99.60	33.43	7.9996

**3.3.3. Analysis of correlation**

In scientific contexts, correlation is used to measure pixel displacements between an image and a reference image, formulated by (12). Table 2 illustrates the correlation among pixels in a selection of encrypted images across three directional axes. Notably, these correlation values approach zero, indicating heightened resistance against correlation-based attacks. In an original image, pixels typically exhibit significant correlation with their neighbors along horizontal, vertical, or diagonal directions, revealing patterns exploitable by attackers to reconstruct the image. Effective encryption systems aim to minimize this correlation as much as possible, ideally approaching zero, to safeguard against such vulnerabilities. This metric of correlation serves as a critical benchmark for assessing the efficacy of the encryption system in maintaining image confidentiality and security.

$$r = \frac{cov(x,y)}{\sqrt{V(x)}\sqrt{V(y)}} \tag{12}$$

**3.4. Analysis of differential constants**

In (13) defines the mathematical analysis of normalized pixel change rate (NPCR) for an image, while (14) specifies the uniformity of average change intensity (UACI). The differential values computed for

reference images using our innovative technology conform to established standards, as depicted in Table 3. Specifically, the NPCR value approaches 99.99%, and the UACI value exceeds 34.65%. These findings validate the robustness of our encryption system against differential attacks, underscoring the efficacy of our initial round implementation. Table 4 presents a comparative analysis with alternative methodologies, further substantiating the effectiveness and reliability of our approach.

$$\left\{ \begin{array}{l} NPCR = \left( \frac{1}{nm} \sum_{i,j=1}^{nm} D(i,j) \right) * 100 \text{ With } D(i,j) = \begin{cases} 1 & \text{if } C_1(i,j) \neq C_2(i,j) \\ 0 & \text{if } C_1(i,j) = C_2(i,j) \end{cases} \\ UACI = \left( \frac{1}{nm} \sum_{i,j=1}^{nm} Abs(C_1(i,j) - C_2(i,j)) \right) * 100 \end{array} \right. \quad (13)$$

Table 4. Examination of correlation and differential constants

Image N°	Lena			Peppers		
	Ours	[30]	[31]	Ours	[30]	[31]
Correlation	0.00032	-0.0016	0.0036	-0.0025	-0.0125	0.0040
NPCR	99.97	99.6017	99.617	99.87	99.618	99.61
UACI	34.68	28.137	29.932	34.96	29.168	29.049

#### 4. CONCLUSION

The efficient implementation and updating of Vigenere functions were demonstrated using two newly generated substitution tables created via linear congruence pseudo-random generators. These generators allowed us to develop non-linear replacement tables for confusion and diffusion functions, enhancing the system's protection against known attacks. This technique shows great promise in advancing image encryption methods. Additionally, the efficiency and encryption speed of this process suggest the feasibility of extending this method to video and audio encryption.




#### REFERENCES

- [1] T. Li, W. Yan, and Z. Chi, "A new image encryption algorithm based on optimized Lorenz chaotic system," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 13, Jul. 2022, doi: 10.1002/cpe.5902.
- [2] J. Chen, L. Chen, and Y. Zhou, "Cryptanalysis of a DNA-based image encryption scheme," *Information Sciences*, vol. 520, pp. 130–141, May 2020, doi: 10.1016/j.ins.2020.02.024.
- [3] Y. Chen, C. Tang, and R. Ye, "Cryptanalysis and improvement of medical image encryption using high-speed scrambling and pixel adaptive diffusion," *Signal Processing*, vol. 167, Feb. 2020, doi: 10.1016/j.sigpro.2019.107286.
- [4] M. Rawat, A. S. Bafila, S. Kumar, M. Kumar, A. Pundir, and S. Singh, "A new encryption model for multimedia content using two dimensional Brownian motion and coupled map lattice," *Multimedia Tools and Applications*, vol. 82, no. 28, pp. 43421–43453, Apr. 2023, doi: 10.1007/s11042-023-14841-z.
- [5] X. Wang, S. Lin, and Y. Li, "Bit-level image encryption algorithm based on BP neural network and gray code," *Multimedia Tools and Applications*, vol. 80, no. 8, pp. 11655–11670, Jan. 2021, doi: 10.1007/s11042-020-10202-2.
- [6] P. Murali and V. Sankaradass, "An efficient ROI based copyright protection scheme for digital images with SVD and orthogonal polynomials transformation," *Optik*, vol. 170, pp. 242–264, Oct. 2018, doi: 10.1016/j.ijleo.2018.04.050.
- [7] P. Ping, X. Zhang, X. Yang, and Y. A. A. Hashems, "A novel medical image encryption based on cellular automata with ROI position embedded," *Multimedia Tools and Applications*, vol. 81, no. 5, pp. 7323–7343, Jan. 2022, doi: 10.1007/s11042-021-11799-8.
- [8] H. Çelik and N. Doğan, "A hybrid color image encryption method based on extended logistic map," *Multimedia Tools and Applications*, vol. 83, no. 5, pp. 12627–12650, Jul. 2024, doi: 10.1007/s11042-023-16215-x.
- [9] H. El Bourakkadi, A. Chemlal, H. Tabti, M. Kattass, A. Jarjar, and A. Benazzi, "Improved Vigenere approach incorporating pseudorandom affine functions for encrypting color images," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 3, pp. 2684–2694, Jun. 2024, doi: 10.11591/ijece.v14i3.pp2684-2694.
- [10] X. Wang and S. Gao, "Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory," *Information Sciences*, vol. 507, pp. 16–36, Jan. 2020, doi: 10.1016/j.ins.2019.08.041.
- [11] S. Mazloom and A. M. Eftekhari-Moghadam, "based on coupled nonlinear chaotic map," *Chaos, Solitons and Fractals*, vol. 42, no. 3, pp. 1745–1754, Nov. 2009, doi: 10.1016/j.chaos.2009.03.084.
- [12] H. El Bourakkadi, A. Chemlal, H. Tabti, M. Kattass, A. Jarjar, and A. Benazzi, "Improved vigenere using affine functions surrounded by two genetic crossovers for image encryption," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 3, pp. 1787–1799, Jun. 2024, doi: 10.11591/ijeecs.v34.i3.pp1787-1799.
- [13] H. Tabti, H. EL Bourakkadi, A. Chemlal, A. Jarjar, K. Zenkouar, and S. Najah, "Genetic crossover at the RNA level for secure medical image encryption," *International Journal of Safety and Security Engineering*, vol. 14, no. 1, pp. 201–216, Feb. 2024, doi: 10.18280/ijssse.140120.
- [14] F. A. Khan, J. Ahmed, and S. A. Alsuhibany, "A new multi chaos-based compression sensing image encryption," *Computers, Materials and Continua*, vol. 76, no. 1, pp. 437–453, 2023, doi: 10.32604/cmc.2023.032236.
- [15] M. Dua, A. Wesanekar, V. Gupta, M. Bholra, and S. Dua, "Differential evolution optimization of intertwining logistic map-DNA




- based image encryption technique,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 9, pp. 3771–3786, Nov. 2020, doi: 10.1007/s12652-019-01580-z.
- [16] S. Du and G. Ye, “IWT and RSA based asymmetric image encryption algorithm,” *Alexandria Engineering Journal*, vol. 66, pp. 979–991, Mar. 2023, doi: 10.1016/j.aej.2022.10.066.
- [17] H. Liu and X. Wang, “Color image encryption based on one-time keys and robust chaotic maps,” *Computers and Mathematics with Applications*, vol. 59, no. 10, pp. 3320–3327, May 2010, doi: 10.1016/j.camwa.2010.03.017.
- [18] X. Huang, Y. Dong, G. Ye, and Y. Shi, “Meaningful image encryption algorithm based on compressive sensing and integer wavelet transform,” *Frontiers of Computer Science*, vol. 17, no. 3, Sep. 2023, doi: 10.1007/s11704-022-1419-8.
- [19] A. Jin, X. Li, and Q. Xiong, “Security compressed sensing image encryption algorithm based on elliptic curve,” in *Communications in Computer and Information Science*, vol. 1879, Springer Nature Singapore, 2023, pp. 350–360, doi: 10.1007/978-981-99-5968-6\_25.
- [20] Y. Zhang, “A unified image cryptography based on a perceptron-like network,” *Visual Computer*, vol. 39, no. 10, pp. 4985–5000, Aug. 2023, doi: 10.1007/s00371-022-02641-9.
- [21] Y. Qobbi, A. Abid, M. Jarjar, S. El Kaddouhi, A. Jarjar, and A. Benazzi, “An image encryption algorithm based on substitution and diffusion chaotic boxes,” in *Lecture Notes in Networks and Systems*, vol. 635, Springer International Publishing, 2023, pp. 184–190, doi: 10.1007/978-3-031-26254-8\_26.
- [22] M. Akraam, T. Rashid, and S. Zafar, “A novel and secure image encryption scheme based on two-dimensional logistic and Arnold Cat map,” *Cluster Computing*, vol. 27, no. 2, pp. 2029–2048, Jun. 2024, doi: 10.1007/s10586-023-04084-w.
- [23] X. Wang and P. Liu, “A new full chaos coupled mapping lattice and its application in privacy image encryption,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 69, no. 3, pp. 1291–1301, Mar. 2022, doi: 10.1109/TCSI.2021.3133318.
- [24] X. Kang, A. Ming, and R. Tao, “Reality-preserving multiple parameter discrete fractional angular transform and its application to color image encryption,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 6, pp. 1595–1607, Jun. 2019, doi: 10.1109/TCSVT.2018.2851983.
- [25] Q. Zhang and X. Wei, “RGB Color image encryption method based on lorenz chaotic system and DNA computation,” *IETE Technical Review (Institution of Electronics and Telecommunication Engineers, India)*, vol. 30, no. 5, pp. 404–409, 2013, doi: 10.4103/0256-4602.123123.
- [26] Y. Q. Zhang and X. Y. Wang, “A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice,” *Information Sciences*, vol. 273, pp. 329–351, Jul. 2014, doi: 10.1016/j.ins.2014.02.156.
- [27] S. Suri and R. Vijay, “A Pareto-optimal evolutionary approach of image encryption using coupled map lattice and DNA,” *Neural Computing and Applications*, vol. 32, no. 15, pp. 11859–11873, Dec. 2020, doi: 10.1007/s00521-019-04668-x.
- [28] M. Z. Talhaoui, X. Wang, and M. A. Midoun, “A new one-dimensional cosine polynomial chaotic map and its use in image encryption,” *Visual Computer*, vol. 37, no. 3, pp. 541–551, Mar. 2021, doi: 10.1007/s00371-020-01822-8.
- [29] A. Abid, Y. Qobbi, A. Benazzi, M. Jarjar, and A. Jarjar, “Two enhanced feistel steps for medical image encryption,” *2022 IEEE 3rd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, Fez, Morocco, 2022, pp. 1–4, doi: 10.1109/ICECOCS55148.2022.9982938.
- [30] S. Wang, L. Hong, and J. Jiang, “An image encryption scheme using a chaotic neural network and a network with multistable hyperchaos,” *Optik*, vol. 268, Oct. 2022, doi: 10.1016/j.ijleo.2022.169758.
- [31] A. N. K. Telem, H. B. Fotsin, and J. Kengne, “Image encryption algorithm based on dynamic DNA coding operations and 3D chaotic systems,” *Multimedia Tools and Applications*, vol. 80, no. 12, pp. 19011–19041, Feb. 2021, doi: 10.1007/s11042-021-10549-0.

## BIOGRAPHIES OF AUTHORS






**Abdelhakim Chemlal**    received the master’s degree in computer engineering with a software engineering specialization from National School of Applied Science in AL Houceima, Morocco, in currently, Ph.D. degrees in mathematics and computer science in Mohammed First University, Oujda, Morocco. His research interests include computer science. He can be contacted at email: Abdelhakim.chemlal.d23@ump.ac.ma.






**Hassan Tabti**    received the master’s degree in computer science infography and imaging from Sidi Mohammed Ben Abdellah University, Morocco, in 2014. Currently, Ph.D. degrees in mathematics and computer science in Mohammed First University, Fez, Morocco. His research interests include computer science. He can be contacted at email: hassan.tabti1@usmba.ac.ma.






**Hamid El Bourakkadi**    received the master's degree in physics of materials and nanostructures from Sidi Mohammed Ben Abdellah University, Morocco, in 2012 and master degree in intelligent and mobile systems from Sidi Mohammed Ben Abdellah University, Morocco, in 2021, respectively. Currently, Ph.D. degrees in mathematics and computer science in Mohammed First University, Oujda, Morocco. His research interests include computer science. He can be contacted at email: hamid.elbourakkadi.d23@ump.ac.ma.






**Hicham Rrghout**    received a master's degree in electronics and telecommunications from Abdelmalek Essaadi University, in 2012. Currently, a Ph.D. student registered in the MATSI Laboratory at ESTO of Mohamed University, Oujda, Morocco, and a professor of computer science in the qualifying secondary cycle. Interested in cryptography and image processing. He can be contacted at email: h.rrghout@ump.ac.ma.



**Abdellatif Jarjar**    received the master's degree in fundamental mathematics from Franche Compté Besonçon University, French, in 1987 and Laureate in mathematics from High Normal School, Morocco, in 1988, respectively. Currently, Searcher in mathematics and computer science in Mohammed First University, Oujda, Morocco. His research interests include computer science. He can be contacted at email: abdoujjar@gmail.com.



**Abdelhamid Benazzi**    professor and searcher in computer science in Mohammed First University, Oujda, Morocco. His research interests include computer science. He can be contacted at email: a.benazzi@ump.ac.ma.