

# Novel cryptosystem integrating the Vigenere cipher and one Feistel round for color image encryption

Hassan Tabti<sup>1</sup>, Hamid El Bourakkadi<sup>2</sup>, Abdelhakim Chemlal<sup>2</sup>, Abdellatif Jarjar<sup>2</sup>, Said Najah<sup>1</sup>,  
Khalid Zenkouar<sup>1</sup>

<sup>1</sup>LSIA Laboratory, Faculty of Science and Technology, Sidi Mohamed Ben Abdellah University, Fez, Morocco

<sup>2</sup>MATSI Laboratory, Mohammed First University, Oujda, Morocco

## Article Info

### Article history:

Received Apr 16, 2024

Revised Jul 8, 2024

Accepted Jul 17, 2024

### Keywords:

Chaotic map

Confusion function

Diffusion Function

Feistel round

S-Box

## ABSTRACT

The present research will propose an innovative technique for pixel-level encryption of color images. After isolating the R, G, and B channels and converting them into vector mode, an enhanced Feistel network will be applied at the hexadecimal level, facilitated by integrating a substitution table generated from the employed chaotic maps. This is followed by a binary conversion and a shift ensured by pseudo-random vectors. A diffusion function is applied, incorporating another replacement matrix constructed from commonly used chaotic maps in cryptography. This operation links the cipher pixel to the next pixel, thereby reinforcing the avalanche effect and safeguarding the system against any differential attacks. Simulations conducted using our new system on various color images, arbitrarily selected from multiple databases, have yielded satisfactory and highly promising results.

*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

In the current context of the digital world, it is essential to secure sensitive data, especially color images. The proliferation of digital data exchanges and storage requires a robust approach to ensure the confidentiality and integrity of images [1], [2]. Given that these images may contain confidential, strategic, or even private information, their unauthorized disclosure can have major repercussions for individuals and entities. Despite the progress made in image encryption, some methods remain vulnerable to sophisticated attacks [3].

The integration of mathematical concepts into cryptography has given rise to encryption standards such as Hill [4], [5], Vigenere [6], [7], and Feistel [8], [9], which were developed in response to growing security needs. Advances in chaos theory, combined with the use of hybrid chaotic maps exploiting new mathematical functions to enhance their chaotic behavior, enable the creation of key stream sequences with remarkable pseudo-random characteristics. This approach significantly enhances the resilience of these cryptographic systems.

Encryption techniques involve making pixels more complex by altering them, thus making them blurry. The first method used is substitution [10], which modifies the pixel values [11]. Then permutation comes into play to randomly change the pixel positions, thereby reducing statistical correlation [12], [13]. Various approaches have been developed to implement substitution, such as using the S-box and integrating a Feistel scheme, aiming to introduce confusion and diffusion [14]–[17].

Meanwhile, other techniques have been employed to perform permutations to scramble the image, increasing unpredictability and the complexity of potential attacks against the system. Almost all traditional encryption methods rely on distinct blocks of operations. This means they remain susceptible to statistical and frequency attacks. Furthermore, without internal chaining in the encryption process, these techniques are exposed to differential attacks.

In our method, we intend to incorporate S-boxes to implement confusion and diffusion operations. This will occur after applying a round of Feistel at the half-pixel level. After converting a pixel into its binary representation and performing appropriate bit shifts, we aim to establish a correlation between the cipher pixel and the subsequent plaintext pixel. This connection will serve to propagate diffusion, thereby reinforcing the system's protection against differential attacks by increasing the impact of the avalanche effect.

The structure of this document is outlined as follows: section 2 examines prior research, while section 3 presents a comprehensive description of the proposed cipher method setup. The experimental security and results analysis of this recommended structure are presented in section 4. The conclusions are addressed in the final section.

## 2. METHOD

Our approach centers on four essential components. First, we meticulously select appropriate chaotic maps [18], [19]. Next, we generate pseudorandom tables integrating confusion and diffusion functions. Thirdly, these elements form the foundation for encrypting the initial image. Lastly, the encrypted image undergoes decryption to recover the original data.

### 2.1. Chaotic sequences design

This initial phase highlights the selection of the two most commonly utilized chaotic maps in cryptography for our cryptographic system [19]. Specifically, the logistic map and the piecewise linear chaotic map (PWLCM) map. Pseudo-random vectors will be generated from these two maps for the construction of all the parameters necessary for the encryption process.

#### 2.1.1. Logistic map

Equation (1) defines the logistic map [20], a cyclic pattern generated by a second-degree polynomial function. This map recurs as a sequence, exhibiting complex behavior often seen in dynamic systems. Its formula encapsulates the iterative process driving its evolution.

$$\begin{cases} \omega_0 \in ]0,5 ; 1[ , \alpha \in ]3,75 ; 4[ \\ \omega_{n+1} = \alpha \omega_n (1 - \omega_n) \end{cases} \quad (1)$$

$\alpha$  is the col,  $\omega_0$  is the initial condition parameter.

#### 2.1.2. PWLCM map

The PWLCM sequence [21], as denoted by (2), is characterized by its piecewise linear nature. It follows a distinct pattern delineated by the specified equation. This sequence exhibits a series of linear segments, each defined by the given equation.

$$\mu_n = f(\mu_{n-1}) = \begin{cases} \mu_0 \in ]0 ; 1[ \quad \theta \in ]0,5 ; 1[ \\ \frac{\mu_{n-1}}{h} \quad \text{if } 0 \leq \mu_{n-1} \leq \theta \\ \frac{\mu_{n-1} - h}{0.5 - h} \quad \text{if } \theta \leq \mu_{n-1} \leq 0.5 \\ f(1 - \mu_{n-1}) \quad \text{else} \end{cases} \quad (2)$$

$(\mu_0)$  and  $(\theta)$  represent, respectively, the initial state and its control parameter.

## 2.2. Pseudo-random tables and substitution functions design

The development process begins with constructing pseudorandom tables and substitution and diffusion functions, which are pivotal components. Following this, pixel-level operations are implemented utilizing an improved Vigenere technique. This phase mandates the creation of specific parameters detailed in the outlined requirements.

- (TS): Tables for confusion and diffusion process in  $(\mathbb{Z}/256\mathbb{Z})$ .
- (TB): Binary tables for control and decision process in  $(\mathbb{Z}/2\mathbb{Z})$ .
- (FS): Substitution function in  $(\mathbb{Z}/256\mathbb{Z})$
- (FH): Substitution function in  $(\mathbb{Z}/16\mathbb{Z})$
- (FV): Diffusion function in  $(\mathbb{Z}/256\mathbb{Z})$ .

### 2.2.1. (TS) table generation

The table (TS) is a pseudorandom matrix of size  $(3 \text{ nm}; 4)$  with coefficients in  $(\mathbb{Z}/256\mathbb{Z})$ . Every individual column of the table (TS) signifies an independent pseudo-random vector distinct from the remaining vectors. The purpose of this table is to induce aliasing and diffusion effects on the original image pixel values. The construction of this table is detailed in Algorithm 1.

#### Algorithm 1. (TS) Pseudorandom table design

```

1. For i = 1 to 3 nm
2. TS(i;1) = mod(E(|ω(i)-μ(i)|*1012); 252)+3
3. TS(i;2) = mod(E((ω(i)+μ(i))*1010); 252)+2
4. TS(i;3) = mod(E(max(ω(i); μ(i))*1011); 254)+1
5. TS(i;4) = mod(E( $\frac{\omega(i)+2*\mu(i)}{4} * 10^{11}$ ); 253)+2
6. Next i

```

### 2.2.2. (TB) binary table's construction

The table (TB) is a pseudorandom matrix of size  $(3 \text{ nm}; 2)$  with coefficients in  $(\mathbb{Z}/2\mathbb{Z})$ . Every column within this table depicts a binary vector that will function as a control for one or multiple encryption procedures. The purpose of this table is to control any cryptographic operations used in our new system. The construction of this table is detailed in Algorithm 2.

#### Algorithm 2. (TB) design

```

1. For i=1 to 3 nm
2. //First column
3. if ω(i)>μ(i) Then
4. TB(i;1)=0 Else TB(i;1)=1
5. End if
6. //Second column
7. if TS(i;1)≥TS(i;2) Then
8. TB(i;2)=0 Else TB(i;2)=1
9. end if
10. Next i

```

### 2.2.3. (S) SBox construction

The primary objective of this section is to create the new Vigenere substitution S-box denoted as (S). It is a pseudorandom table with dimensions  $(256; 256)$  with coefficients in  $(\mathbb{Z}/256\mathbb{Z})$ . The process for constructing this S-box adheres to the outlined instructions.

- The rearrangement ("P1") is generated by conducting a wide ascending sort on the initial 256 values of the vector "TS(i; 3)".
- The second line, designated as "P2" is derived from conducting an extensive ascending sort on the initial 256 values of the vector "TS(i; 4)".
- The intermediate line, denoted as "P3" is generated through a comprehensive ascending sorting process applied to the initial 256 values of the vector "TS(i; 2)".
- The line labeled as " $i > 3$ " is the result of combining the functions from the line " $i - 2$ "/" $i - 3$ " or " $i - 3$ "/" $i - 1$ " contingent upon the decision vector "TB(i; 2)" value.

The construction of this table is detailed in Algorithm 3. Table 1 illustrate an example for (S) creation of size  $(8 ;8)$  controlled by TB(i;2).

#### Algorithm 3. (S) SBox construction

```

// First three lines
for j=1 to 256 //3 first lines
    S(1;j)=Pr1(j)
    S(2;j)=Pr2(j)
    S(3;j)=Pr3(j) : Next j

```

```

// Other lines
for i = 4 to 256 //other lines
for j = 1 to 256
  if TB(i;2) = 0 then
    S(i;j) = S(i-2;S(i-3,j))
  else
    S(i,j) = S(i-3;S(i-1,j)) :
  end if
end if
Next j, i

```

Table 1. (S) Creation example

S	1	2	3	4	5	6	7	8	TB(i;2)	
P1	1	8	4	7	3	1	2	6	5	<del>1</del>
P2	2	4	5	6	3	8	2	1	7	<del>0</del>
P3	3	7	5	8	6	4	1	2	3	<del>0</del>
P4 = P1oP3	4	6	1	5	2	3	8	4	7	1
P5 = P2oP4	5	2	4	8	5	6	7	3	1	1
P6 = P4oP5	6	1	2	7	3	8	4	5	6	0
P7 = P4oP6	7	6	1	4	5	7	2	3	8	1
P8 = P6oP5	8	2	3	6	8	4	5	7	1	0

### 2.2.4. Hybrid confusion function expression

This section is dedicated to developing a new confusion function that will be applied to each pixel. This function relies on a substitution box (S) as a crucial element. Algorithm 4 describes the steps for constructing such a function.

Algorithm 4. (FS) Hybrid chaining function expression

$$Z(i) = FS(Y(i)) = \begin{cases} S(TS(i;1);S(TS(i;2);Y(i))) & \text{if } TB(i;1) = 0 \\ S(TS(i;3);S(TS(i;4);Y(i))) & \text{if } TB(i;1) = 1 \end{cases}$$

### 2.2.5. Diffusion function

The diffusion process is a crucial operation in any encryption system. This section aims to design a new diffusion function operating at the pixel level, connecting the encrypted pixels to the subsequent plaintext pixels. The expression of such a function is given by (3).

$$FD(Z(i)) = \begin{cases} FS(Z(i) \oplus TS(i;2)) \oplus V_x(i+1) & \text{if } TB(i;2) \\ FS(Z(i) \oplus TS(i;1)) \oplus V_x(i+1) & \text{if } TB(i;2) \end{cases} \quad (3)$$

### 2.2.6. (H) SBox construction

The primary objective of this part is to create the enhanced Vigenere substitution SBox denoted as (H). It is a pseudorandom table with dimensions (256; 16) with coefficients in  $(Z/16Z)$ . The process for constructing this SBox is generated by Algorithm 5 and adheres to the outlined instructions:

- The initial row of the table, designated as (H), represents the permutation Q1 of the initial 16 values extracted from the vector  $TH(i;1)$ . This permutation is achieved by arranging these values in descending order;
- For orders greater than 1, the rank line represents an order shift either from  $TH(i;2)$  or  $TH(i;1)$ , contingent upon the  $TB(i;1)$  decision vector value. Tables 2 and 3 illustrate an example explaining the construction of this SBox.

Algorithm 5. (H) Substitution box construction

```

for j ← 1 to 16 // First line
  H(1;j) ← Q1(j)
for i ← 2 to 256 // Next lines
  if TB(i;1)=0 then
    for j ← 1 to 16
      H(i;j) ← H(i-1;mod(j+TS((i;2);16)))
    else
      for j ← 1 to 16
        H(i,j) ← H(i-1;mod(j+TS((i;1);16)))
      end for
    end if
  end for
Next i

```

Table 2. First-line design procedure

Order	1	2	3	4	5	6	7	8
TH(i;1) Values	3	5	3	6	7	2	4	2
Sort	3	6	4	7	8	1	5	2
Permutation	Q1 = $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 6 & 4 & 7 & 8 & 1 & 5 & 2 \end{pmatrix}$							

Table 3. Creation of (H) of size (8;8) controlled by TB(i;1) example

H	1	2	3	4	5	6	7	8	TH(i;1)	TH(i;2)	TB(i;1)
1	3	6	4	7	8	1	5	2	2	5	1
2	1	5	2	3	6	4	7	8	7	5	1
3	5	2	3	6	4	7	8	1	1	3	0
4	4	7	8	1	5	2	3	6	4	3	0
5	1	5	2	3	6	4	7	8	3	3	1
6	4	7	8	1	5	2	3	6	2	5	1
7	8	1	5	2	3	6	4	7	2	4	0
8	2	3	6	4	7	8	1	5	6	3	1

2.2.7. (FH) Hybrid confusion function

This section is dedicated to developing a new confusion function that will be applied to each pixel. This function relies on a substitution box (S) as a crucial element. Algorithm 6 describes the steps for constructing such a function.

Algorithm 6. (FH) Expression of the chaining function

$$YR(i) = FH(XR(i)) = \begin{cases} H(TS(i;1); H(TS(i;3); XR(i))) & \text{if } TB(i;1) = 0 \\ H(TS(i;2); H(TS(i;4); XR(i))) & \text{if } TB(i;1) = 1 \end{cases}$$

2.3. Encryption phase

The encryption process consists of four essential steps. Initially, the original image undergoes vectorization, consolidating the three RGB channels into a one-dimensional vector. Next, an initialization value is calculated to start the encryption, followed by an enhanced Feistel round incorporating specified substitution and diffusion functions. Finally, a global replacement operation will be implemented to enhance the complexity of attacks on our system.

2.3.1. Plain image to vector transition

Following the extraction of the three-color channels from the original image and their conversion into vectors (VR), (VG), and (VB), a pseudo-random concatenation is formed, guided by the decision vector TB(i;1). This process is detailed in Algorithm 7 and visually represented in Figure 1. The preliminary step involves preparing the plain image before advancing to the encryption process resulting in a new encrypted image with the ability to resist statistical and frequency attacks.

Algorithm 7. Plain image to vector transition algorithm

```

for j ← 1 to nm
    if TB(i;1) = 0 then
        VX(3i-2) = Vr(i) ⊕ TS(i;3)
        VX(3i-1) = Vg(i) ⊕ TS(i;4)
        VX(3i) = Vb(i) ⊕ TS(i;1)
    else
        VX(3i-2) = Vr(i) ⊕ TS(i;3)
        VX(3i-1) = Vg(i) ⊕ TS(i;4)
        VX(3i) = Vb(i) ⊕ TS(i;1)
    end if : Next i
    
```

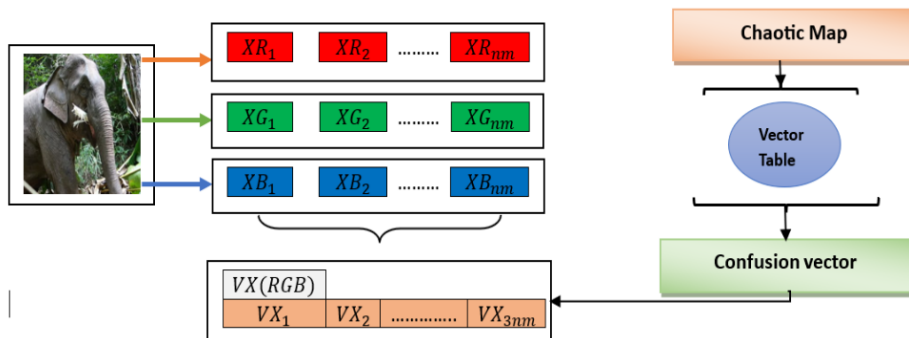


Figure 1. Plain image to vector transition flowchart

**2.3.2. Initialization value calculation**

An initialization value (VI), closely associated with both the plain image and the pseudorandom table (TS) and under the control of the decision table (TB), is calculated. This operation aims solely the alteration of the seed pixel value to initiate the encryption process. It is outlined in Algorithm 8. At the output of this first round, we will implement a Feistel circuit applied at the half-pixel level.

Algorithm 8. The constant of the Initialization computation algorithm

```

VI=0
for i = 2 to 3nm
    if TB(i;2)=0 then
        VI= VI⊕VX(i)⊕TS(i;1)
    else
        VI= VI⊕VX(i)⊕TS(i;3)
    end if
Next i
    
```

**2.3.3. Improved Feistel round**

In this step, we apply a single round of improved Feistel cipher. This operation is applied to each pixel of the original image divided into two sub-blocks with hexadecimal values. The entire round involves two substitution functions and two diffusion functions. The entire process is given by Algorithm 9. Figures 2 and 3 illustrate the flowchart and an example of this algorithm respectively. The encryption phase is depicted in the diagram of Figure 4. This process involves chaining encrypted pixels with the following plaintext pixels. The diffusion step is a crucial stage in enhancing the security of the system.

Algorithm 9. Reverse operation of substitution tables

```

For i = 1 to 3nm
    For j = 1 to 16
        IH(i,H(i,j))= j
    Next j, i
    
```

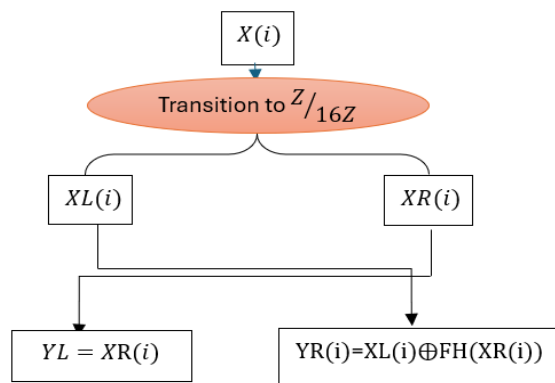


Figure 2. Original images transform into two sub-blocks with hexadecimal values

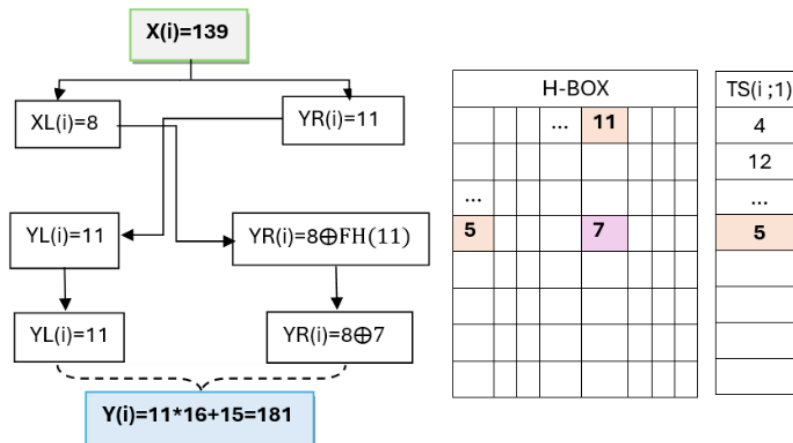


Figure 3. Example of original images transform into sub-blocks with hexadecimal values

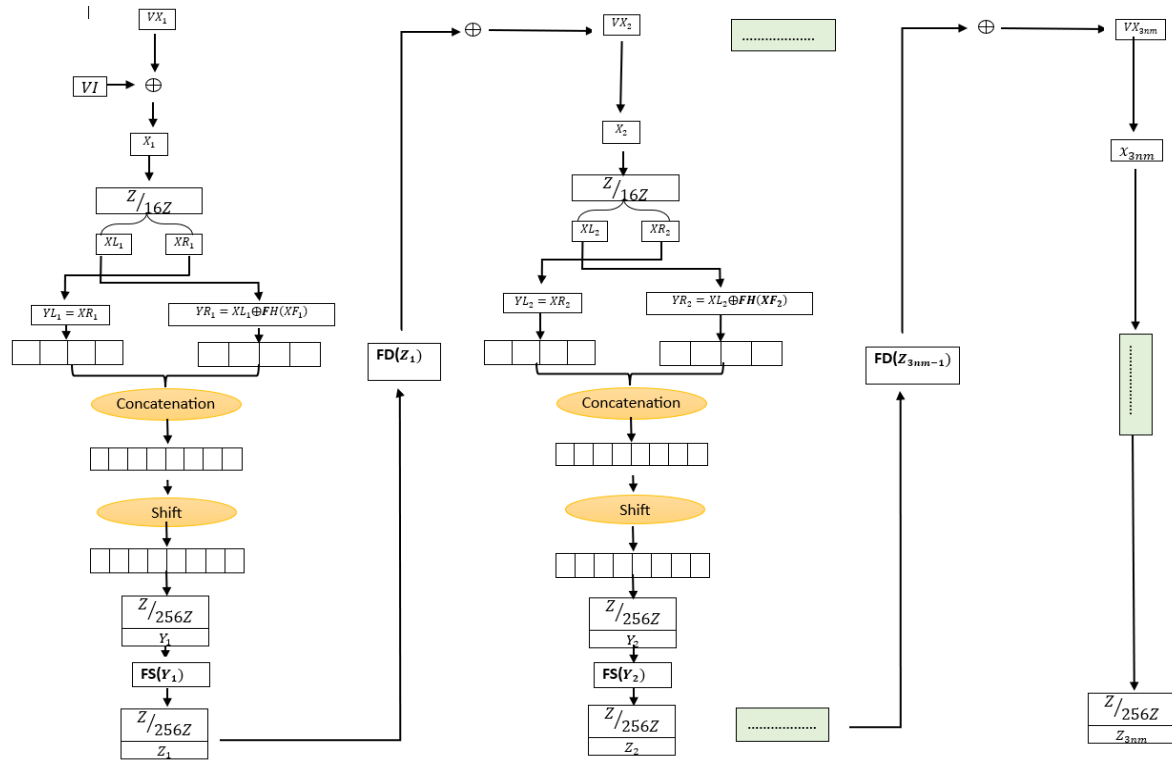


Figure 4. Encryption process diagram

**2.4. Decryption operation description**

The present cipher system is symmetric and employs confusion/diffusion functions. Consequently, the reverse procedure commences with the final step, involving the application of reverse functions. This procedure is described by the following steps:

**2.4.1. Inverse operation of Vigenere**

Our system is a symmetric architecture which needs that all parts of its must be invertible. In this case, the functions (S) must be invertible. The inverse operation of Vigenere (SF) entails constructing substitution tables (IS) as outlined in Algorithm 10. Figure 5 illustrates an application example of such an algorithm.

Algorithm 1. (SF) Hybrid chaining inverse function expression

$$Y(i) = SF(Z(i)) = \begin{cases} IS(TS(i; 2); IS(TS(i; 1); Z(i))) & \text{if } TB(i; 1) = 0 \\ IS(TS(i; 4); IS(TS(i; 3); Z(i))) & \text{if } TB(i; 1) = 1 \end{cases}$$

H	1	2	3	4	5	6	7	8
1	6	8	5	1	7	3	2	4
2	3	5	1	6	8	7	4	2
3	5	7	4	2	3	8	1	6
4	7	3	6	8	4	5	2	1
5	4	3	7	5	6	1	8	2
6	6	4	8	7	2	5	1	3
7	2	6	7	8	1	3	5	4
8	5	2	6	3	8	1	4	7

IH	1	2	3	4	5	6	7	8
1	4	7	6	8	3	1	5	2
2	3	8	1	7	2	4	6	5
3	7	4	5	3	1	8	2	6
4	8	7	2	5	6	3	1	4
5	6	8	2	1	4	5	3	7
6	7	5	8	2	6	1	4	3
7	5	1	6	8	7	2	3	4
8	6	2	4	7	1	3	8	5

Figure 5. The inverse operation of the Vigenere substitution

**2.4.2. (HF) reverse functions**

Our system features a symmetric architecture, necessitating that all its components be reversible. In this case, the functions (FH) must be invertible. So, the reverse of the first function is dedicated to confusion operation is given by Algorithm 11.

Algorithm 11. (HF) Hybrid chaining inverse function expression  

$$XR(i) = HF(YR(i)) = \begin{cases} IH(TS(i; 3); IH(TS(i; 1); YR(i))) & \text{if } TB(i; 1) = 0 \\ IH(TS(i; 4); IH(TS(i; 2); YR(i))) & \text{if } TB(i; 1) = 1 \end{cases}$$

**2.4.3. Feistel reverse round**

Our system employs a symmetric architecture, requiring all components to be reversible. Therefore, the Feistel functions must be invertible. Accordingly, Algorithm 12 specifies the reverse operation for the Feistel round.

Algorithm 12. The reverse Feistel round  

$$XR(i) = YL(i)$$

$$XL(i) = YR(i) \oplus FH(YL(i))$$

**3. RESULTS AND DISCUSSION**

All trials were carried out utilizing the Python programming language, using a laptop equipped with an i5 processor, 8 GB of RAM, and a 256 GB capacity hard drive. Figure 6 illustrates the primary test image, named "Peppers" alongside its encrypted and subsequently decrypted renditions, in addition to every simple image employed. The images provided in this collection were obtained from the SIPI database, accessible at [20]. The keys and assorted experimental parameters are created utilizing the chaotic maps as outlined in prior descriptions. Before starting the procedure of decryption, it is crucial to securely transfer the private key to the intended receiver via secure communication.

**3.1. Study of statistical assaults**

The recently developed algorithm was implemented on a randomly chosen set of images for testing and evaluation. Subsequent security studies were conducted, leveraging simulation findings to explore various aspects of its effectiveness and resilience. The discussions below delve into these studies and their implications for enhancing overall system security.

**3.1.1. Study of the key space**

The system utilizes a pair of chaotic maps created through the manipulation of four real parameters. Each of the four parameters is symbolized by a set of 32 bits, resulting in a key comprising a total of 128 bits. This particular design ensures protection from brute-force assaults.

**3.1.2. Study of the key sensibility**

The system employs two chaotic maps that are frequently utilized in the cryptography domain. Their increased sensibility to initial states a notable degree of responsiveness to the encryption key, as depicted in Figure 6. We note that changing the cipher key leads to the creation of two separate cipher images in the procedure of encryption. Furthermore, in the decryption phase, two distinct decrypted images are produced, each exhibiting unique shapes. This underscores the increased sensitivity of the suggested technique to even minor variations in the cipher key.

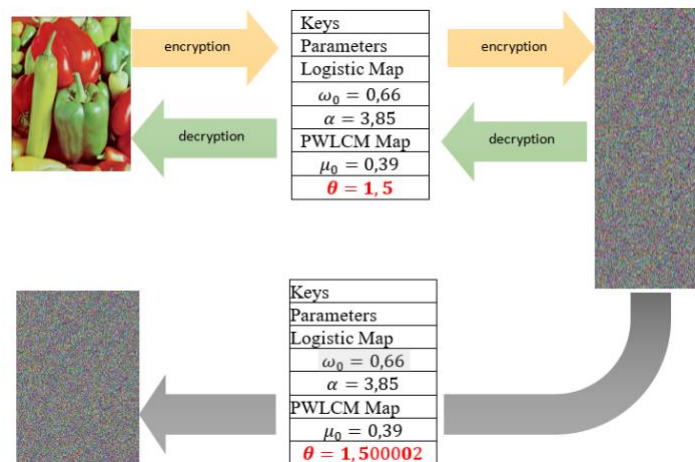


Figure 6. Key sensibility study



**3.1.3. Histogram analysis**

Histograms illustrate the dispersion of pixel values within images. Individuals who lack proper authorization might obtain significant information from an encrypted image through the examination of its non-uniform histograms. Hence, it is imperative to verify that the cipher image histogram shows no numerical resemblance to the original image and ensure and uphold a steady distribution of pixels. The histograms depicted in Figures 7(a) and 7(b) illustrate the distribution of RGB channels in unencrypted images, whereas the histograms in Figures 8(a) and 8(b) display the RGB channel distribution in encrypted images. Employing the suggested approach, it is evident that the histograms of images generated during the encryption phase display a close-to-uniform and evenly distributed pattern.

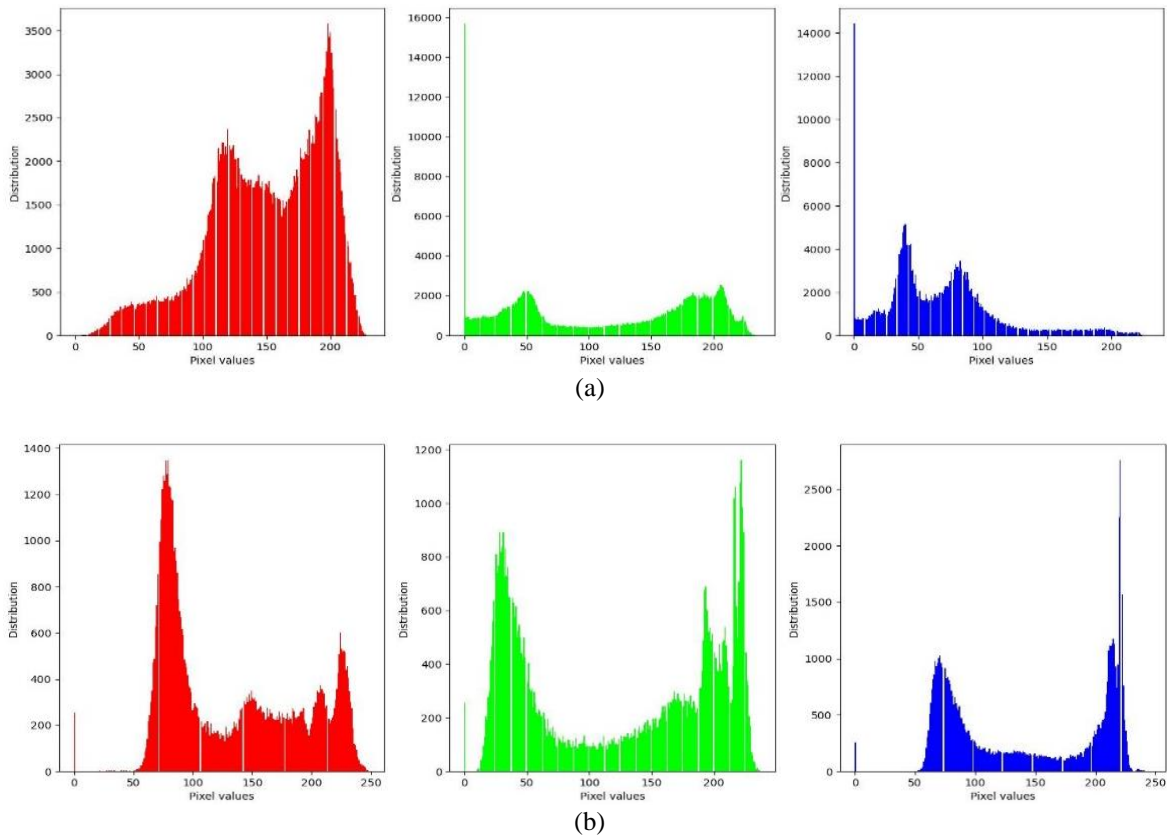


Figure 7. Plain (RGB) image histograms (a) histogram of peppers and (b) histogram of vegetables

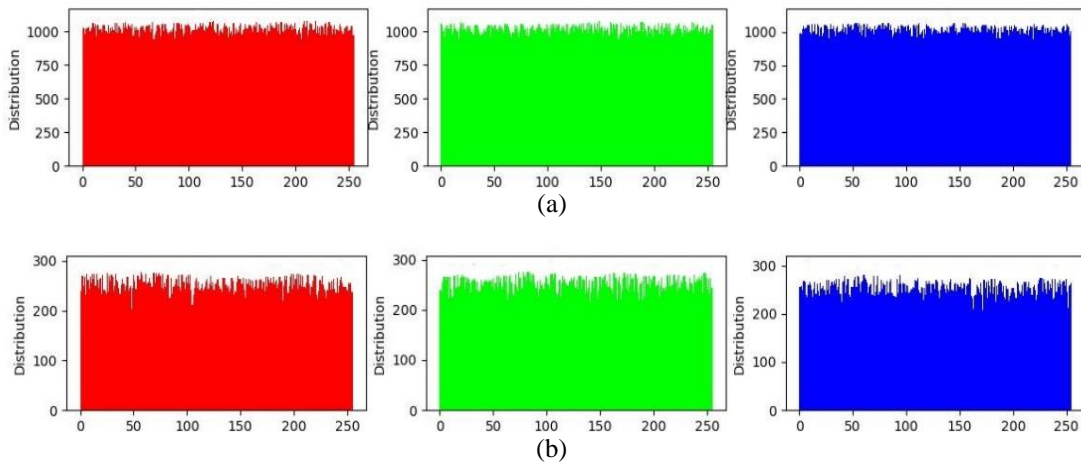


Figure 8. Cipher (RGB) images histograms (a) histogram of peppers and (b) histogram of vegetables

### 3.1.4. Entropy analysis

The entropy of an image sized (n, m) serves as a metric for assessing the effectiveness of the image encryption technique in ensuring security. It represents the level of uncertainty and variability naturally present within the system, as denoted by (5).

$$\left\{ \begin{aligned} S(MC) &= \frac{1}{3nm} \sum_{j=1}^{3nm} p(j) \cdot \log_2 \left( \frac{1}{p(j)} \right) \\ p(j) &\text{ denotes the likelihood of level } j \text{ appearing within the plain image} \end{aligned} \right. \quad (5)$$

As the entropy nears the value 8, the haphazard image pixel's distribution becomes increasingly organized. Augmented entropy aids in reducing the likelihood of information leakage from the cipher image. Table 4 presents the entropy values of the analyzed images juxtaposed with those from various cipher methods in the literature. The findings reveal that the values of this metric consistently remain exceed or equal to 7,997, In line with the research conducted by study [22], our findings surpass the results reported by numerous other researchers [23].

Table 4. Analyzing the entropy of cipher images in contrast to alternative methodologies

Method	Images	Cipher		
		Red	Green	Blue
Ours	Lena	7.9973	7.9974	7.9971
	Pepper	7.9994	7.9992	7.9994
	Vegetables	7.9994	7.9995	7.9994
Ref [22]	Lena	7.9972	7.9973	7.9970
	Pepper	7.9993	7.9994	7.9994
	Vegetables	7.9993	7.9994	7.9993
Ref [23]	Lena	7.9974	7.9974	7.9971
	Pepper	7.9993	7.9994	7.9992
	Vegetables	7.9993	7.9994	7.9993

### 3.1.5. Study of the correlation

Equation (6) illustrates the correlation between an image with dimensions (n, m), while Table 3 presents the calculated correlation coefficients for specific reference images sourced from the SIPI dataset [24]. Assessed through our proprietary algorithm, the correlation values generated adhere to recognized global standards, this confirms the strength of our cryptographic system against correlation-based attacks, affirming its resilience.

$$corr = \frac{cov(A,B)}{\sqrt{var(A)} \cdot \sqrt{var(B)}} \quad (6)$$

Table 5 shows the pixel correlation for images sourced from the SIPI dataset, along with additional test images. Upon reviewing the table data, it is evident that there exists a significantly strong correlation in the original image, with values nearing 1 in each channel. On the contrary, pictures encrypted using the suggested algorithm display notably reduced correlation. This observation underscores the algorithm's effective security measures. Additionally, these findings underscore a notable decrease in correlation within the cipher image, meaning that potential attackers are incapable of extracting information.

Table 5. Pixel correlations within images sourced from the SIP dataset

Images		Plain image			Cipher image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Peppers	Red	0.97582	0.97547	0.94929	-0.001265	-0.001875	-0.005407
	Green	0.9777	0.98171	0.95789	-0.001056	0.001505	0.002341
	Blue	0.98938	0.98803	0.97825	0.004789	-0.005700	-0.001175
Vegetables	Red	0.97886	0.98012	0.96108	0.005135	-0.000765	-0.004952
	Green	0.97749	0.97952	0.95947	0.007877	-0.000793	0.000273
	Blue	0.97224	0.97091	0.94729	0.000065	0.010967	-0.001057
Lena	Red	0.9558	0.9648	0.9325	-0.003671	0.008139	-0.001310
	Green	0.93556	0.95756	0.91902	-0.002885	0.009126	-0.006730
	Blue	0.90773	0.9393	0.8913	-0.001348	-0.006715	0.000642

Table 6 illustrates the pixel correlations identified in the 'Lena' image. In contrast to prior methods, it is evident that the adjacent pixel's cipher image mutual correlation is lower than that observed in numerous studies [22], [23], [25]. Although greater than that of study [26], the proposed encryption system exhibits nearly no correlation measure for all tested image correlations, ensuring protection against statistical assaults.

Table 6. The correlation comparison among the cipher pixels of the "Lena" image

Method	Horizontal	Vertical	Diagonal
Ours	-0,002634667	0,003516667	-0,002466
Ref [22]	-0.0029883	0.0091357	-0.0067375
Ref [23]	-0.0042707	-0.0032498	-0.0032498
Ref [25]	0.002733667	0.00352	0.002469667
Ref [26]	0.000059	0.0041	0.00013

### 3.2. Study of differential attacks

The number of changing pixel rate (NPCR) and unified averaged changed intensity (UACI) metrics are utilized to quantify the disparity between two ciphertext images by modifying the value of a pixel in the plaintext image and employing the same encryption algorithm to encrypt this variation compared to the plain image. The definition formulas of UACI and NPCR are:

$$\begin{cases} \text{UACI} = \left( \frac{1}{3nm} \sum_{i,j=1}^{3nm} \frac{|I_1(i,j) - I_2(i,j)|}{255} \right) * 100\% \\ \text{NPCR} = \left( \frac{1}{3nm} \sum_{i,j=1}^{3nm} \text{Df}(i,j) \right) * 100\% \end{cases} \quad (7)$$

where,  $\text{Df}(i,j) = \begin{cases} 1 & \text{if } I_1(i,j) \neq I_2(i,j) \\ 0 & \text{if } I_1(i,j) = I_2(i,j) \end{cases}$ ,  $I_1(i,j)$  is the cipher modified image pixel of rank  $(i,j)$  and  $I_2(i,j)$  is the cipher image pixel of rank  $(i,j)$ .

The proposed method undergoes a rigorous comparison with other algorithms [27]–[30] to assess its compliance with general requirements and its behavior. Lena and Peppers are chosen as test images, with the use of NPCR and UACI for in-depth analysis. Upon careful examination of the various algorithms, it becomes evident that the suggested method provides enhanced security against differential attacks, surpassing the performance of current methods. The detailed comparison findings are documented in Table 7. Additionally, it may be relevant to include additional metrics such as compression efficiency, robustness to steganography, or fault attack resistance.

Table 7. Our UACI and NPCR comparison with those of other techniques

Method	Peppers		Lena	
	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)
Ours	33.53	99.64	33.56	99.67
Ref [27]	33.46391676	99.61054611	33.46184906	99.60851796
Ref [28]	33.4097	99.5897	33.5098	99.6016
Ref [29]	-	-	33.5501	99.5987
Ref [30]	33.4097	99.5897	33.5098	99.6016

### 3.3. Peak signal-to-noise ratio metric study

The peak signal number rate (PSNR) and mean squared error (MSE) serve as prevalent benchmarks for assessing image encryption methods. PSNR evaluates an encryption technique by quantifying the discrepancy in pixel values between the plain and cipher images. A lower PSNR value denotes superior encoding quality. The formulas for PSNR and MSE [31] are:

$$\begin{cases} \text{MSE} = \frac{1}{3nm} \sum_{i,j=1}^{3nm} (p_1(i,j) - p_0(i,j))^2 \\ \text{PSNR} = 20 \log_{10} \left( \frac{255}{\sqrt{\text{MSE}}} \right) \text{dB} \end{cases} \quad (8)$$

$$\text{PSNR} = 20 \log_{10} \left( \frac{255}{\sqrt{\text{MSE}}} \right) \text{dB} \quad (9)$$

where  $M$  and  $N$  denote the height and width of images, respectively.

For the original and ciphered images, the intensity values of the pixels are represented by  $P_0(i, j)$  and  $P_1(i, j)$ . As per the findings, a low PSNR value and a high MSE value between the plain image and their cipher one signifies desirable cipher quality. The decreased PSNR values noted in Table 8 for the plain-to-cipher images imply that the suggested methods offer enhanced ciphering in comparison to other techniques outlined in various studies [31], [32], meaning that the suggested technique can restore images with minimal loss of information.

Table 8. The calculated PSNR (dB) between the plain/cipher, and the plain/decrypted images

Method	Type of PSNR	Lena	Peppers	Cameraman
Ours	Original/decrypted	$\infty$	$\infty$	$\infty$
	Original/Encrypted	7.0450	7.0640	7.0750
Ref [33]	Original/Encrypted	8.3655	8.8532	-
Ref [31]	Original/decrypted	$\infty$	-	-
	Original/Encrypted	7.7268	-	-
Ref [34]	Original/Encrypted	9.2500	8.9100	8.4000
Ref [35]	Original/Encrypted	9.2267	8.8792	8.4045

#### 4. CONCLUSION

A groundbreaking encryption system, skillfully blending the hybrid techniques of Feistel and Vigenère, has been meticulously designed and implemented. This revolutionary device not only incorporates two new substitution tables but also robust diffusion and confusion functions, thus ensuring enhanced security. Subsequently, a specially adapted permutation for image encryption is globally applied, adding an extra layer of protection. Innovatively, the integration of chaotic maps of high sensibility to initial conditions enhances the reliability of our new method, especially in color image encryption. Simulation results outcomes derived from a randomly selected array of images, encompassing diverse formats and sizes, have fully validated the effectiveness of our algorithm against all known types of attacks. In terms of future perspectives, our approach will evolve to encompass a range of sophisticated algorithms such as wavelet transformations, reinforcement learning, supervised learning, and fuzzy methods. This ensures ongoing adaptation and advanced security against future cryptographic challenges.




#### REFERENCES

- [1] A. Gonsai, R. Tanma, and R. Somaiya, "Implementation and evaluation of EMAES – a hybrid encryption algorithm for sharing multimedia files with more security and speed," *International journal of electrical and computer engineering systems*, vol. 14, no. 4, pp. 401–409, Apr. 2023, doi: 10.32985/ijeces.14.4.4.
- [2] 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/ijss.140120.
- [3] A. Mehmood, A. Shafique, M. Alawida, and A. N. Khan, "Advances and vulnerabilities in modern cryptographic techniques: a comprehensive survey on cybersecurity in the domain of machine/deep learning and quantum techniques," *IEEE Access*, vol. 12, pp. 27530–27555, 2024, doi: 10.1109/ACCESS.2024.3367232.
- [4] S. Hraoui, F. Gmira, M. F. Abbou, A. J. Oulidi, and A. Jarjar, "A new cryptosystem of color image using a dynamic-chaos Hill cipher algorithm," *Procedia Computer Science*, vol. 148, pp. 399–408, 2019, doi: 10.1016/j.procs.2019.01.048.
- [5] L. S. Mezher and A. M. Abbass, "Mixed Hill cipher methods with triple pass protocol methods," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 5, pp. 4449–4457, Oct. 2021, doi: 10.11591/ijece.v11i5.pp4449-4457.
- [6] M. Jarjar, S. Hraoui, S. Najah, and K. Zenkouar, "New cryptosystem using two improved Vigenere laps separated by a genetic operator," *Research Square*, 2022.
- [7] 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.
- [8] S. M. Kareem and A. M. S. Rahma, "New modification on Feistel DES algorithm based on multi-level keys," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 3, pp. 3125–3135, Jun. 2020, doi: 10.11591/ijece.v10i3.pp3125-3135.
- [9] Y. Wang, L. Teng, and X. Wang, "An image encryption algorithm based on circular rotation and generalized Feistel structure," *Soft Computing*, Jun. 2023, doi: 10.1007/s00500-023-08747-z.
- [10] M. Kattass, H. Rrghout, A. Jarjar, A. Abid, M. Jarjar, and A. Benazzi, "An efficient image encryption algorithm using chaotic S-boxes of pseudo-random size," in *Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security*, May 2023, pp. 1–6, doi: 10.1145/3607720.3607754.
- [11] Z. Bashir, N. Iqbal, and M. Hanif, "A novel gray scale image encryption scheme based on pixels' swapping operations," *Multimedia Tools and Applications*, vol. 80, no. 1, pp. 1029–1054, Jan. 2021, doi: 10.1007/s11042-020-09695-8.
- [12] D. Wei, M. Jiang, and Y. Deng, "A secure image encryption algorithm based on hyper-chaotic and bit-level permutation," *Expert Systems with Applications*, vol. 213, Mar. 2023, doi: 10.1016/j.eswa.2022.119074.
- [13] L. Teng, X. Wang, and J. Meng, "A chaotic color image encryption using integrated bit-level permutation," *Multimedia Tools and Applications*, vol. 77, no. 6, pp. 6883–6896, Mar. 2018, doi: 10.1007/s11042-017-4605-1.
- [14] A. H. Zahid, E. Al-Solami, and M. Ahmad, "A novel modular approach based substitution-box design for image encryption," *IEEE Access*, vol. 8, pp. 150326–150340, 2020, doi: 10.1109/ACCESS.2020.3016401.




- [15] Y. Aydın and F. Özkaynak, "Automated chaos-driven s-box generation and analysis tool for enhanced cryptographic resilience," *IEEE Access*, vol. 12, pp. 312–328, 2024, doi: 10.1109/ACCESS.2023.3346319.
- [16] F. Artuğer and F. Özkaynak, "An effective method to improve nonlinearity value of substitution boxes based on random selection," *Information Sciences*, vol. 576, pp. 577–588, Oct. 2021, doi: 10.1016/j.ins.2021.07.036.
- [17] Q. Sheng, C. Fu, Z. Lin, J. Chen, L. Cao, and C.-W. Sham, "An efficient chaotic image encryption scheme using simultaneous permutation–diffusion operation," *The Visual Computer*, vol. 40, no. 3, pp. 1643–1658, 2024, doi: 10.1007/s00371-023-02876-0.
- [18] A. JarJar, "Improvement of Feistel method and the new encryption scheme," *Optik*, vol. 157, pp. 1319–1324, Mar. 2018, doi: 10.1016/j.ijleo.2017.12.065.
- [19] R. B. Naik and U. Singh, "A review on applications of chaotic maps in pseudo-random number generators and encryption," *Annals of Data Science*, Jan. 2022, doi: 10.1007/s40745-021-00364-7.
- [20] D. R. I. M. Setiadi and N. Rijati, "An image encryption scheme combining 2D cascaded logistic map and permutation-substitution operations," *Computation*, vol. 11, no. 9, Sep. 2023, doi: 10.3390/computation11090178.
- [21] W. Alexan, M. Elkandoz, M. Mashaly, E. Azab, and A. Aboshousha, "Color image encryption through chaos and KAA map," *IEEE Access*, vol. 11, pp. 11541–11554, 2023, doi: 10.1109/ACCESS.2023.3242311.
- [22] S. Khan, H. Lansheng, Y. Qian, H. Lu, and S. Meng Jiao, "Security of multimedia communication with game trick based fast, efficient, and robust color-/gray-scale image encryption algorithm," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 2, Feb. 2021, doi: 10.1002/ett.4034.
- [23] K. K. Butt, G. Li, S. Khan, and S. Manzoor, "Fast and efficient image encryption algorithm based on modular addition and SPD," *Entropy*, vol. 22, no. 1, Jan. 2020, doi: 10.3390/e22010112.
- [24] USC Viterbi, "SIPI database," *Signal and Image Processing Institute*, <https://sipi.usc.edu/database/database.php?volume=misc> (accessed Feb. 28, 2024).
- [25] 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.
- [26] J. G. Sekar, E. Periyathambi, and A. Chokkalingam, "Hybrid chaos-based image encryption algorithm using Chebyshev chaotic map with deoxyribonucleic acid sequence and its performance evaluation," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 6, pp. 6952–6963, Dec. 2023, doi: 10.11591/ijece.v13i6.pp6952-6963.
- [27] E. Moya-Albor, A. Romero-Arellano, J. Brievea, and S. L. Gomez-Coronel, "Color image encryption algorithm based on a chaotic model using the modular discrete derivative and Langton's ant," *Mathematics*, vol. 11, no. 10, May 2023, doi: 10.3390/math11102396.
- [28] H. R. Shakir, S. A. Mehdi, and A. A. Hattab, "A new four-dimensional hyper-chaotic system for image encryption," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 2, pp. 1744–1756, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1744-1756.
- [29] X. Zhang, L. Wang, G. Cui, and Y. Niu, "Entropy-based block scrambling image encryption using des structure and chaotic systems," *International Journal of Optics*, vol. 2019, pp. 1–13, Aug. 2019, doi: 10.1155/2019/3594534.
- [30] F. Budiman, P. N. Andono, and M. Setiadi, "Image encryption using double layer chaos with dynamic iteration and rotation pattern," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 2, pp. 57–67, Apr. 2022, doi: 10.22266/ijies2022.0430.06.
- [31] A. A. Rashid and K. A. Hussein, "Image encryption algorithm based on the density and 6D logistic map," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 2, pp. 1903–1913, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1903-1913.
- [32] A. Ullah *et al.*, "An efficient lightweight image encryption scheme using multichaos," *Security and Communication Networks*, vol. 2022, pp. 1–16, Oct. 2022, doi: 10.1155/2022/5680357.
- [33] A. A. Rashid and K. A. Hussein, "A lightweight image encryption algorithm based on elliptic curves and a 5D logistic map," *Iraqi Journal of Science*, pp. 5985–6000, Nov. 2023, doi: 10.24996/ijis.2023.64.11.41.
- [34] A. ur Rehman, D. Xiao, A. Kulsoom, M. A. Hashmi, and S. A. Abbas, "Block mode image encryption technique using two-fold operations based on chaos, MD5 and DNA rules," *Multimedia Tools and Applications*, vol. 78, no. 7, pp. 9355–9382, Apr. 2019, doi: 10.1007/s11042-018-6516-1.
- [35] S. Benaissi, N. Chikouche, and R. Hamza, "A novel image encryption algorithm based on hybrid chaotic maps using a key image," *Optik*, vol. 272, Feb. 2023, doi: 10.1016/j.ijleo.2022.170316.

## BIOGRAPHIES OF AUTHORS






**Hassan Tabti**    received a 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 from 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 a master's degree in physics of materials and nanostructures from Sidi Mohammed Ben Abdellah University, Morocco, in 2012 and a Master's degree in intelligent and mobile systems from Sidi Mohammed Ben Abdellah University, Morocco, in 2021, respectively. Currently, Ph.D. degrees in 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.






**Abdelhakim Chemlal**    received a master's degree in computer engineering with a software engineering specialization from the 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](mailto:abdelhakim.chemlal.d23@ump.ac.ma).






**Abdellatif Jarjar**    received a master's degree in fundamental mathematics from Franche Compté Besonçon University, France, in 1987 and a Laureate in mathematics from High Normal School, Morocco, in 1988, respectively. Currently, searcher in mathematics and computer science from Mohammed First University, Oujda, Morocco. His research interests include computer science. He can be contacted at email: [abdoujar@gmail.com](mailto:abdoujar@gmail.com).



**Said Najah**    He received a Ph.D. degree in computer science from the Faculty of Science, University Sidi Mohamed Ben Abdellah, Fez, Morocco in 2006. He is currently a professor of the Department of Computer Science, Faculty of Science and Technology Fez Morocco. He is a member in the LSIA Laboratory (Laboratory of intelligent systems and application). His current research interests include parallel computing, big data analytics and artificial intelligence. He can be contacted at email: [Said.najah@usmba.ac.ma](mailto:Said.najah@usmba.ac.ma).



**Khalid Zenkouar**    Received a Ph.D. degree in image analysis from Faculty of Science, University Sidi Mohamed Ben Abdellah, Fez, Morocco in 2006. Now he is a professor of the Department of Computer Engineering, Faculty of Science and Technology Fez Morocco. He is a member in the LSIA Laboratory (Laboratory of intelligent systems and application). His current research interests include image analysis, machine intelligence and pattern recognition. He can be contacted at email: [khalid.zenkouar@usmba.ac.ma](mailto:khalid.zenkouar@usmba.ac.ma).