

## Improved Vigenere approach incorporating pseudorandom affine functions for encrypting color images

Hamid El Bourakkadi<sup>1</sup>, Abdelhakim Chemlal<sup>1</sup>, Hassan Tabti<sup>2</sup>, Mourad Kattass<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, Sidi Mohamed Ben Abdellah University, Fez, Morocco

### Article Info

#### Article history:

Received Jan 3, 2024

Revised Feb 19, 2024

Accepted Mar 5, 2024

#### Keywords:

Chaotic map

Global permutation

Hybrid chaining

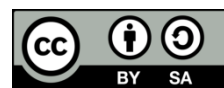
Reversible affine function

S-Box

### ABSTRACT

This article presents an improvement to the traditional Vigenere encryption method, specifically adapted for the encryption of color images. This enhancement relies on the use of two chaotic maps widely employed in the field of cryptography. After vectorizing the original image and calculating the initialization value, which alters the seeding pixel to trigger the encryption process, our approach integrates two new large substitution tables. These tables are linked to confusion and diffusion functions, incorporating multiple reversible pseudo-random affine functions at the pixel level. Finally, a global permutation is applied to the entire resulting vector to increase the temporal complexity of potential attacks on our system. Simulations conducted on a diverse set of images of various sizes and formats demonstrate the resilience of our approach against any unexpected attacks.

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



### Corresponding Author:

Hamid El Bourakkadi

MATSI Laboratory, High School of Technology, Mohammed First University

Oujda 60000, Morocco

Email: hamid.elbourakkadi.d23@ump.ac.ma

## 1. INTRODUCTION

Ciphering images is the process of protecting them by altering their pixels in a way that makes them indecipherable. It protects image confidentiality and integrity, particularly when the images are sensitive and confidential, as is the military case [1] and medical images [2], [3]. Perturbations in ciphering steps complicate the statistical relationships between original and ciphered images and make predicting them difficult. Diffusion, on the other hand, distributes data in its initial form efficiently and uniformly throughout the entire ciphered image [4], [5]. Encryption methods achieve diffusion and confusion through dense substitution and permutation of each pixel. Substitution is performed by changing the image pixel values to other values. Permutation randomly arranges image pixels [6] to conceal the statistical relationships between the image pixels. Various methods can be used for replacement, such as the S-box operation [7]–[9]. As the patterns of replacement and permutations become more difficult, the situation becomes more unpredictable and complex. Therefore, a combination of replacement, which integrates dynamic affine functions, and permutations must be applied at the smallest unit of images, which is the pixel. Thus, improving existing classical techniques such as Hill [10], [11], Caesar, Vigenere [12]–[16], and affine [17], [18] adds value to image encryption. The classic Vigenere encryption method depends on a predetermined (26, 26) matrix that is purpose-built for the encryption of text. Additionally, Researchers described in the article [19], [20] a new encryption method that combines the Vigenere cipher with dynamic deoxyribonucleic acid (DNA) and ribonucleic acid (RNA) sequence coding [17], [18]. First, the image to be encrypted is converted to a vector

format and then mutated genetically. Next, an initialization value is determined to create a diffusion process [15]. This value is used as the initial condition for the Vigenere cipher. The key is a crucial element in the encryption process, playing a vital role in safeguarding the confidentiality of ciphered data. Therefore, the key needs to meet various criteria, including considerations such as length, space, and complexity [21], [22]. This implies that preprocessing of the user-inputted key for image encryption is necessary to generate a more intricate form, such as a pseudorandom sequence [23]. However, these keys can be altered by simple operations like rearranging the image stream [24], [25].

The challenge lies in the fact that most classical algorithms rely on independent block encryption, making them vulnerable to statistical attacks. Additionally, the small size of private keys exposes them to brute force attacks. However, in the absence of diffusion and chaining functions between encrypted and plaintext blocks, these methods remain susceptible to differential attacks.

Our contribution is evident through the development of a novel image encryption system. This new system utilizes a large private key to safeguard against exhaustive attacks. Furthermore, the use of new confusion and diffusion functions employing reversible dynamic affine functions provides high resilience to our system against unforeseen attacks.

The remainder of this article is organized into several sections, including one detailing our method elucidating the intricacies of the encryption and decryption process. Another section is dedicated to results and discussion, presenting research conclusions along with comparisons to other similar techniques. Finally, a section summarizes the findings and suggests directions for future research.

## 2. METHOD

Based on chaos, our method uses a profound improvement of the classic Vigenere technique by employing large substitution tables [26], [27] attached to new pseudorandom substitution functions. We integrated reversible affine functions into the encryption process. This technique is based on the axes below.

### 2.1. Axis 1: choice of chaotic sequences

#### 2.1.1. Skew tent map

The skew tent map [28] is a mathematical function commonly employed in chaos theory and cryptography. It is a one-dimensional map, renowned for its chaotic behavior and ease of integration into any crypto-system. The general expression of this map, denoted by the numerical sequence  $h$ , is defined by (1). The parameters ( $h_0$ ) and ( $k$ ) represent the initial value and control parameter, respectively.

$$\begin{cases} h_0 \in [0.5; 1], k \in [0.5; 1] \\ h_{n+1} = \begin{cases} \frac{h_n}{k} & \text{if } 0 < h_n < k \\ \frac{1-h_n}{1-k} & \text{elsewhere} \end{cases} \end{cases} \quad (1)$$

#### 2.1.1. Logistic map

The logistic map [29] is a mathematical function commonly employed to simulate the growth of a population over time within a constrained space. It is frequently utilized in chaos theory and cryptography. This logistic map is expressed by (2).

$$\begin{cases} l_0 \in [0.5; 1] \text{ and } \delta \in [3.75; 4] \\ l_{n+1} = \delta \cdot l_n(1 - l_n) \end{cases} \quad (2)$$

### 2.2. Axis 2: construction of pseudorandom vectors

#### 2.2.1. Used chaotic sequences

The two selected chaotic maps for our new system are highly sensitive to initial conditions and easy to implement in a cryptographic system. They are used for pseudo-random vector generation. The resulting vectors form the encryption subkeys.

#### 2.2.2. Sub keys construction

Our system needs the construction of keys and sub keys. So, we need seven pseudorandom vectors ( $Vc1$ ), ( $Vc2$ ), ( $Vc3$ ), ( $Vr$ ), ( $Ve$ ), ( $Va$ ), and ( $Vb$ ). These vectors are described by coefficients in the ring ( $Z/256Z$ ) and generated by Algorithm 1.

**Algorithm 1. Pseudorandom vectors generation**

```

for i = 1 to 3nm
  Vc1(i) = [E(sup(h(i);l(i)). ⌊10⌋ ^11 ) mod 253] + 2
  Vc2(i) = [E(((h(i) + 2 * l(i))/3). ⌊10⌋ ^11 ) mod 254] + 1
  Vc3(i) = [E(|h(i) - l(i)|. ⌊10⌋ ^10 ) mod 254] + 1
  Vr(i) = [E((h(i) + l(i)). 1012) mod 253] + 2
  Ve(i) = [E(( $\frac{2 * h(i) + 3 * l(i)}{5}$ ). 1012) mod 253] + 2
  Va(i) = [2 * E((h(i) + l(i)). 1012) + 1] mod 256
  Vb(i) = [2 * E((h(i) * l(i)). 1012) + 1] mod 256
end for

```

The two vectors ( $Va$ ) and ( $Vb$ ) contain only the invertible elements in the ring ( $Z/256Z$ ). In addition, our system requires the generation of three binary vectors, ( $Ba1$ ), ( $Ba2$ ), and ( $Ba3$ ), to control the encryption process. These two vectors are generated by Algorithm 2.

**Algorithm 2. ( $Ba_i$ ) Binary random vectors generation,  $i \in \{1, 2, 3\}$** 

```

//Binary vectors construction
for j ← 1 to 3nm
  if h(j) > l(j) then :Ba1(j) ← 0
  else : Ba1(j) ← 1
  end if
  if h(j) > 0.5 then :Ba2(j) ← 0
  else : Ba2(j) ← 1
  end if
  if h(j) ≤ l(j) then: Ba3(j) ← 0
  else : Ba3(j) ← 1
  end if
end for

```

**2.3. Axis 3: substitution table construction**

Our algorithm requires the development of two new replacement tables ( $Tv1$ ) and ( $Tv2$ ). Each table is a matrix of size (256;256) with coefficients in the ring ( $Z/256Z$ ). The construction of these substitution tables is described as follows.

**2.3.1. ( $Tv1$ ) S-Box construction**

The main mission of this section is to construct the new Vigenere substitution matrix, called ( $Tv1$ ), with a size of (256;256), following the instructions provided below.

- The first row of the table ( $Tv1$ ) is the permutation ( $Pt1$ ) of the first 256 values of the vector ( $Vc1$ ), obtained by sorting them in decreasing order.
- For ranks higher than 1, the rank line is a rank shift  $Vc2(j)$  or  $Vc3(j)$ , depending on the value of the control vector  $Ba1(j)$ . This table was generated by Algorithm 3.

**Algorithm 3. ( $Tv1$ ) Substitution box construction**

```

// First line
for j ← 1 to 256
  Tv1(1,j) ← Pt1(i)
end for
// Next lines
for j ← 2 to 256
  for k ← 1 to 256
    if Ba1(j) = 0 then
      Tv1(j,k) ← Tv1(j - 1, mod(k + Vc2(j),256))
    else : Tv1(j,k) ← Tv1(j - 1, mod(k + Vc3(j),256)) : end if
  end for
end for

```

**2.3.2. ( $Tv2$ ) S-Box construction**

The construction of the new substitution matrix ( $Tv2$ ) of size (256;256) is described by these steps:

- The 1<sup>st</sup> line is the rearrangement ( $Pr1$ ) obtained by a broad ascending sort on the first 256 values of the vector ( $Vc3$ );
- The 2<sup>nd</sup> line is the rearrangement ( $Pr2$ ) obtained by a broad ascending sort on the first 256 values of the vector ( $Vc2$ );
- The 3<sup>rd</sup> line is the rearrangement ( $Pr3$ ) obtained by a broad ascending sort on the first 256 values of the vector ( $Vc1$ );
- The  $i^{\text{th}}$  line ( $i > 3$ ) is the composition of the functions of the line ( $i - 2$ ) and ( $i - 3$ ) or ( $i - 3$ ) and ( $i - 1$ ), depending on the value of the control vector  $Ba2(i)$ .

These steps are illustrated in Algorithm 4.

**Algorithm 4. ( $Tv2$ ) substitution box construction**

```

//3 first lines
for j ← 1 to 256
  Tv2(1,j) ← Pr1(j)
  Tv2(2,j) ← Pr2(j)
  Tv2(3,j) ← Pr3(j)
end for
//Next lines
for j ← 4 to 256
  for k ← 1 to 256
    if Ba2(j) = 0 then : Tv2(j,k) ← Tv2(j - 2, Tv2(j - 3, k))
    else: Tv2(j,k) ← Tv2(j - 3, Tv2(j - 1, k)) : end if
  end for
end for

```

#### 2.4. Axis 4: global permutation construction

To increase the attack complexity on the proposed system, the vector ( $Z$ ) undergoes a global rearrangement process ( $Pg$ ). This permutation is achieved through a descending sort on the first ( $3\text{ nm}$ ) values of the chaotic sequence (1). The permutation process is determined by Algorithm 5.

Algorithm 5. Global permutation

```
for  $i = 1$  to  $3nm$ 
 $Zs(i) = Z(Pg(i))$ : end for
```

#### 2.5. Axis 5: affine and hybrid chaining functions construction

##### 2.5.1. Affine functions in $Z/nZ$

Let ( $f$ ) be an affine function defined in the ring ( $Z/nZ$ ) by (3). The function ( $f$ ) is bijective in ( $Z/nZ$ ) if and only if ( $a$ ) is invertible and ( $b$ ) is any.

$$\begin{cases} f: Z/nZ \rightarrow Z/nZ \\ x \mapsto \text{mod}(ax + b; n) \end{cases} \quad a, b \in Z/nZ \quad (3)$$

Indeed, we have  $y = \text{mod}(ax + b; n)$ , then,  $ax = \text{mod}(y - b; n)$  and  $x = \text{mod}(a^{-1} \cdot (y - b); n)$  where ( $a^{-1}$ ) is the inverse of ( $a$ ) in ring ( $Z/nZ$ ). Or, we know that ( $a$ ) is invertible in ( $Z/nZ$ ) if and only if  $a \wedge n = 1$ . Particular case:  $n = 2^k$ ,  $k \in N$ , ( $a$ ) is invertible in ring ( $Z/2^kZ$ ) if and only if ( $a$ ) is odd.

##### 2.5.2. Expression of the improved affine function

Let ( $f_i$ ) be the family of affine functions acting on pixels. These functions are defined by (4). Here,  $Va(i)$  and  $Vb(i)$  are invertible elements in the ring ( $Z/256Z$ ). The affine functions ( $f_i$ ) generated are reversible for all  $i \in [1; 3\text{ nm}]$ .

$$\begin{cases} f_i: Z/256Z \rightarrow Z/256Z \\ x \mapsto \begin{cases} \text{mod}(Va(i) * X(i) + Ve(i); 256) & \text{if } Ba2(i) = 0 \\ \text{mod}(Vb(i) * X(i) + Vr(i); 256) & \text{if } Ba2(i) = 1 \end{cases} \end{cases} \quad (2)$$

##### 2.5.3. Expression of the improved affine function

The new substitution function involving tables ( $Tv1$ ) and ( $Tv2$ ) is given by Algorithm 6. This replacement function promotes the process of diffusion. It can enhance the security of our system.

Algorithm 6. ( $Fv$ ) Hybrid chaining function expression

```
 $Z(i) = Fv(X(i))$ 
if  $Ba2(i) = 0$  then:  $Z(i) \leftarrow Tv1(Vc1(i), Tv2(Vc2(i); \text{mod}(Va(i) * X(i) + Ve(i); 256)))$ 
else:  $Z(i) \leftarrow Tv2(Vc3(i), Tv1(Vc1(i); \text{mod}(Vb(i) * X(i) + Vr(i); 256)))$ : end if
```

#### 2.6. Axis 6: phase of encryption

The encryption phase unfolds through two stages. The first stage involves vectorizing the image. Then, the second stage introduces the new Vigenere function to enhance the confusion/diffusion operation.

##### 2.6.1. Original image vectorization

After extracting the three color channels (RGB) and converting them into vectors ( $Cr$ ), ( $Cg$ ), and ( $Cb$ ) respectively, a pseudo-random concatenation is applied under the control of the binary decision vector ( $Ba1$ ). This operation gives rise to the vector ( $X$ ) of dimension ( $1, 3\text{ nm}$ ). The concatenation process is determined by Algorithm 7.

Algorithm 1. Original image vectorization algorithm

```
for  $j \leftarrow 1$  to  $nm$ 
if  $Ba1(j) = 0$  then
 $X(3j - 2) \leftarrow Cr(j) \oplus Vc1(j)$ 
 $X(3j - 1) \leftarrow Cg(j) \oplus Vc2(j)$ 
 $X(3j) \leftarrow Cb(j) \oplus Vc3(j)$ 
else
 $X(3j - 2) \leftarrow Cr(j) \oplus Vc3(j)$ 
 $X(3j - 1) \leftarrow Cg(j) \oplus Vc1(j)$ 
 $X(3j) \leftarrow Cb(j) \oplus Vc2(j)$ 
end if: end for
```

##### 2.6.2. Improved Vigenere by the affine method

The new encryption process requires the computation of an initialization value ( $In$ ) linked to the original image. This value is determined solely to alter the starting pixel value and thereby initiate the encryption process. The calculation of this value is provided by Algorithm 8.

**Algorithm 2. Initialization value calculation**

```

In = 0
for i = 2 to 3 nm
    if Ba3(i) = 0 then
        In = In ⊕ X(i) ⊕ Vc2(i)
    else
        In = In ⊕ X(i) ⊕ Vc3(i)
    end if
end for
    
```

To overcome any differential attack, we employ diffusion functions utilizing pseudo-random vectors and dynamic affine functions. This chaining process enhances the impact of the avalanche effect. The application of diffusion functions is depicted by Algorithm 9 and interpreted in Figure 1.

**Algorithm 3. Hybrid chaining function expression**

```

//First-pixel encryption
Z(1) = Fv(X(1) ⊕ In ⊕ Vc1(1))
//Next pixels encryption
for i = 2 to 3 nm
    α = fi(X(i)) ⊕ Z(i - 1)
    if Ba3(i) = 0 then
        Z(i) = Fv(α ⊕ Vc2(i))
    else
        Z(i) = Fv(α ⊕ Vc3(i))
    end if
end for
    
```

The obtained vector ( $Z$ ) undergoes global permutation ( $P_g$ ) to generate the vector ( $Z_s$ ). This vector represents the encrypted image.

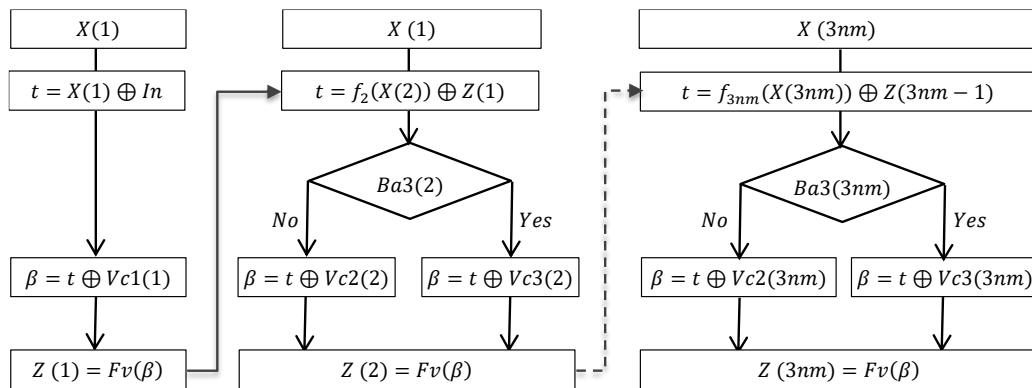


Figure 1. New circuit using dynamic pseudorandom affine functions

**2.7. Axis 7: phase of decryption**

Our new algorithm is a symmetric encryption system employing diffusion functions. Consequently, the decryption process must commence with the final step utilizing the inverse encryption functions. The encrypted image is transformed into a vector ( $Z_s$ ) of dimension  $(1; 3 \text{ nm})$ , upon which the steps below.

**2.7.1. Application of the inverse of the global permutation**

This operation involves permuting the pixels of the vector ( $Z_s$ ) by the inverse function ( $G_p$ ) of the function ( $P_g$ ). This enables the restoration of the vector ( $Z$ ) upon which the inverse Vigenere function is applied. This decryption process is determined by Algorithm 10.

**Algorithm 4. Inverse permutation**

```

for i = 1 to 3 nm
    Gp(Pg(i)) = i:
end for
    
```

**2.7.2. Application of the substitution function**

The inverse of the replacement function requires the construction of two inverse substitution tables ( $Vt1$  and  $Vt2$ ), as provided by Algorithm 11. The inverse function of substitution is given by Algorithm 12.

**Algorithm 5. ( $Vt1$ ) and ( $Vt2$ ) inverse substitution box construction**

```

for j ← 1 to 256
    Vt1(1, Tv1(i; j)) ← j
end for
for j ← 1 to 256
    Vt2(1, Tv2(i; j)) ← j
end for
    
```

Algorithm 6. (*Vf*) The inverse of hybrid chaining function expression

$$X(i) = Vf(Z(i))$$

if  $Ba2(i) = 0$  then:

$$X(i) \leftarrow \text{mod} \left( (Va(i))^{-1} * (Vt2(Vc2(i); Vt1(Vc1(i); Z(i)))) - (Va(i))^{-1} * Ve(i); 256 \right)$$

else:

$$X(i) \leftarrow \text{mod} \left( (Vb(i))^{-1} * (Vt1(Vc1(i); Vt2(Vc3(i); Z(i)))) - (Vb(i))^{-1} * Vr(i); 256 \right); \text{ end if}$$

### 3. RESULTS AND DISCUSSION

All the simulations were implemented in Python on the Windows 10 operating system with a hardware environment consisting of an i7 processor laptop, a 1 TB hard drive, and 32 GB of RAM. The tested images samples were taken from [30]. The keys and other experimental parameters are generated from the chaotic maps described above. Before initiating the decryption process, the secret key needs to be securely transmitted to the recipient through a protected channel.

#### 3.1. Statistical attacks

A statistical attack utilizes statistical analyses of encrypted data to reveal details about the encryption key or the plaintext or plain images. Several reference images chosen at random were tested by our new algorithm at this stage. The recorded simulations are described in the following subsections:

##### 3.1.1. Analysis of possible key space

Our algorithm uses two chaotic maps generated by four real parameters represented by 32 bits each. The all-key space encompasses a key of 120 bits. This ensures that our system is resistant to any brute-force attack.

##### 3.1.2. Key sensitivity analysis

The two chaotic maps used are characterized by their extreme sensitivity to initial conditions. This means that any perturbation of the private key will result in completely different encrypted images. For instance, as illustrated in Figure 2, a small modification in the key value of 0.000001 leads, during decryption, to another decrypted image distinct from the original image.

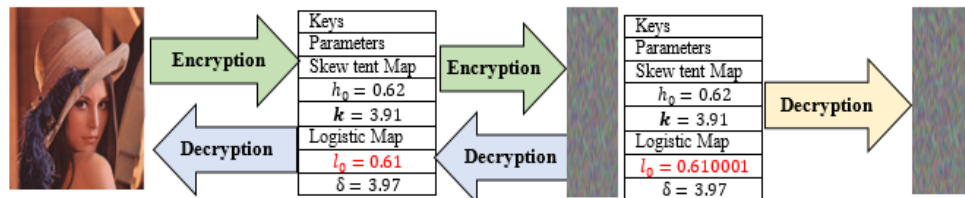


Figure 2. Key sensitivity analysis

##### 3.1.3. Analysis of histograms

Table 1 shows RGB histograms of original and encrypted Lena and Baboon images using our method. The RGB histogram results of encrypted images by our algorithm shows a uniform distribution. These results ensure that our system can withstand histogram-based attacks.

##### 3.1.4. Analysis of entropy

Entropy is a metric that assesses the level of uncertainty in a sequence. Higher entropy indicates greater randomness and increased difficulty in predicting the key or data. Crypto-systems often strive to maximize entropy to enhance security, thereby making the system more resistant to attacks. The entropy of an image is given by (5),

$$S(MC) = \frac{-1}{3nm} \sum_{i=1}^{3nm} p(i) \cdot \log_2(p(i)) \tag{5}$$

where,  $p(i)$  represents the probability of occurrence of level (i) in the plain image.

Table 2 shows a comparison of the entropy value level of our system with other similar algorithms. These results ensure that our technique is more efficient than the other algorithms compared in references [31], [32]. This confirm that our system is robust against statistical attacks.

Table 1. Histograms of cipher and plain images

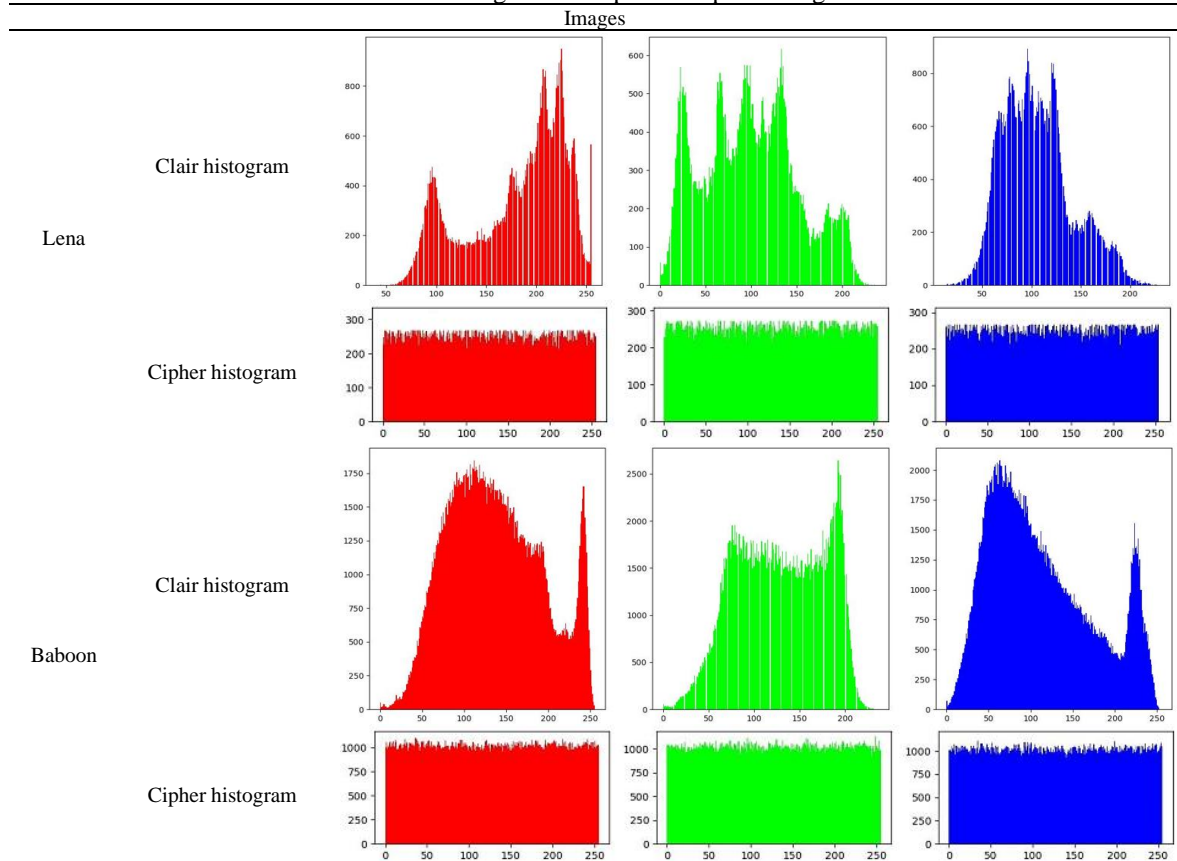


Table 2. Comparison of encrypted image entropy with other methods: (L) Lena, (Pe) Pepper, (H) House

Algorithm	Images	Encrypted		
		Red	Green	Blue
Proposed	(L)	7,9973	7,9974	7,9971
	(Pe)	7,9994	7,9994	7,9995
	(H)	7,9983	7,9982	7,9983
[31]	(L)	7,9974	7,9974	7,9971
	(Pe)	7,9993	7,9994	7,9992
	(H)	7,9993	7,9992	7,9993
[32]	(L)	7,9972	7,9973	7,9970
	(Pe)	7,9993	7,9994	7,9994

**3.1.5. Correlation analysis**

Equation (6) provides the correlation of an image with dimensions (n, m). Table 3 provides the calculated values of the correlation for some reference images taken from the SIPI database [30] tested by our algorithm. The values of correlation calculated by our algorithm adhere to international standards. The obtained results ensure that our crypto-system is immune to correlation attacks.

Table 3. Correlations between pixels in images taken from the SIP database

Images		Original image			Encrypted image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena	Red	0.9558	0.9648	0.9325	-0.003771	0.008149	-0.00132
	Green	0.93556	0.95756	0.91902	-0.002981	0.009127	-0.006732
	Blue	0.90773	0.9393	0.8913	-0.001449	-0.006716	0.000643
Apricot	Red	0.98385	0.96944	0.98629	-0.00136621	-0.0018756	-0.0054082
	Green	0.97883	0.98511	0.96537	-0.00106622	0.0015054	0.0023428
	Blue	0.99153	0.98348	0.98724	0.0048931	-0.0057105	-0.0011768
Panda	Red	0.95175	0.96552	0.93161	0.0051302	-0.0007677	-0.0049534
	Green	0.95215	0.96436	0.93066	0.0078786	-0.0007949	0.0002736
	Blue	0.95542	0.97086	0.94265	0.000070	0.0109689	-0.0010586

$$corr = \frac{cov(x,y)}{\sqrt{var(x)}\sqrt{var(y)}} \quad (6)$$

where  $Cov(x, y)$  is the covariance between the two variables  $x$  and  $y$ ; and  $Var(x)$  and  $Var(y)$  are the variances of the variables  $x$  and  $y$ , respectively.

Table 4 details a comparison at the level of the correlation of our approach with other similar approaches. This shows that our technique is more efficient compared to other algorithms in references [31]–[33]. The obtained results ensure that our crypto-system is immune to statistical attacks.

Table 4. Correlation between ciphered “Lena” pixels

Method	Horizontal	Vertical	Diagonal
Proposed	-0.002733667	0.00352	-0.002469667
[31]	-0.0042707	-0.0032498	-0.0020192
[32]	-0.0029883	0.0091357	-0.0067375
[33]	-0.0098	-0.0050	-0.0013

### 3.2. Differential attacks

Differential attacks leverage variations in input processing within a crypto-system. They focus on the system's responses to subtle changes in plaintext or the key to deduce sensitive information, such as the encryption key. To assess the algorithm's efficacy against differential attacks, metrics such as the number of pixel change rates (NPCR), the unified average change intensity (UACI), and the avalanche effect are employed.

#### 3.2.1. NPCR and UACI metrics analysis

NPCR and UACI are metrics used to evaluate the performance of image encryption algorithms. These metrics are commonly employed in the field of image encryption and provide quantitative measures of the quality and security of the encryption process. They can be given by (7) and (8) respectively.

$$NPCR = \left( \frac{1}{3nm} \sum_{i,j=1}^{3nm} Df(i,j) \right) \cdot 100 \quad (7)$$

$$UACI = \left( \frac{1}{3nm} \sum_{i,j=1}^{3nm} \frac{|Im_1(i,j) - Im_2(i,j)|}{255} \right) \cdot 100 \quad (8)$$

where,

- $Df(i,j) = \begin{cases} 1 & \text{if } Im_1(i,j) \neq Im_2(i,j) \\ 0 & \text{if } Im_1(i,j) = Im_2(i,j) \end{cases}$
- $Im_1(i,j)$  is the first encrypted image pixel of rank  $(i,j)$
- $Im_2(i,j)$  is the second encrypted modified image pixel of rank  $(i,j)$ .

Table 5 presents the UACI and NPCR values calculated on the reference images tested by our algorithm. The values of differential constants calculated by our algorithm adhere to international standards. This confirms that our technique outperforms other algorithms [31], [32], [34], [35]. This ensures the protection of our system against any differential attack.

Table 5. Comparison of the NPCR and UACI (L) Lena and (Pe) Pepper

Method	Lena		Pepper	
	NPCR	UACI	NPCR	UACI
Proposed	99.68%	33.49	99.67	33.48
[31]	99.68%	33.46	99.67	33.48
[31]	99.60%	33.49	99.61	33.46
[34]	99.66%	33.44	99.63	33.47
[35]	99.64%	33.03	-	-

#### 3.2.2. PSNR metric analysis

The PSNR is evaluated in decibels and is inversely proportional to the mean squared error. It is determined by (9).

$$PSNR = 10 * \log_{10} \left( \frac{(2^L - 1)^2}{MSE} \right) (dB) \quad (9)$$



where,

- $MSE = \frac{1}{(3nm)^2} \sum_{i,j=1}^{3nm} |Im_1(i,j) - Im_2(i,j)|^2$ : mean squared error
- $(Im_1)$  and  $(Im_2)$  represent the original and encrypted images, respectively.
- $L=8$  denotes the bit depth of the particular image,  $(n)$  and  $(m)$  are the dimensions of the given image.

Table 6 presents the PSNR values calculated on the reference images tested by our algorithm. The obtained values by our algorithm are within the standards, ensuring that our encryption algorithm is better than that of references [31], [36]. This confirm that our system is robust against differential attacks.

Table 6. The PSNR (dB) between the original image, the encrypted image, and the decrypted image

Method	Type of PSNR	Lena	Baboon	Panda	Vegetables
Ours	Original to Encrypted	∞	∞	∞	∞
	Original to Decrypted	7,0312	7,1811	7,1748	6,8800
[31]	Original to Decrypted	∞	∞	∞	∞
	Original to Encrypted	8,1102	8,7776	8,1648	6,8760
[36]	Original to Encrypted	8.9605	9.2372	-	-

#### 4. CONCLUSION

The analysis of statistical and differential constants, in accordance with international standards, was carried out using pseudo-random and reversible affine functions in the confusion and diffusion functions. Additionally, two S-Boxes generated from chaotic maps were integrated. This approach led to the creation of a large-scale algorithm that imparts a uniformly distributed histogram to each encrypted image. As a result, our cryptographic system exhibits robustness against all known attacks, as demonstrated by comparisons with several similar algorithms.

#### ACKNOWLEDGEMENTS

There are no noteworthy findings to disclose. We are disseminating this article solely for the benefit of the scientific community.




#### REFERENCES

- [1] A. M. Alnajim, E. Abou-Bakr, S. S. Alruwisan, S. Khan, and R. A. Elmanfaloty, "Hybrid chaotic-based PRNG for secure cryptography applications," *Applied Sciences*, vol. 13, no. 13, Jun. 2023, doi: 10.3390/app13137768.
- [2] A. Odeh and Q. Abu Al-Hajja, "Medical image encryption techniques: a technical survey and potential challenges," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 3, pp. 3170–3177, Jun. 2023, doi: 10.11591/ijece.v13i3.pp3170-3177.
- [3] E. Akhtarkavan, B. Majidi, and A. Mandegari, "Secure medical image communication using fragile data hiding based on discrete wavelet transform and  $A_5$  Lattice vector quantization," *IEEE Access*, vol. 11, pp. 9701–9715, 2023, doi: 10.1109/access.2023.3238575.
- [4] B. Rajarao and M. Sreenivasulu, "Hierarchical attribute based cryptographic model to handle security services in cloud environment: a new model," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, pp. 1102–1111, Feb. 2024, doi: 10.11591/ijece.v14i1.pp1102-1111.
- [5] E. Winarno, K. Nugroho, P. W. Adi, and D. R. I. M. Setiadi, "Combined interleaved pattern to improve confusion-diffusion image encryption based on hyperchaotic system," *IEEE Access*, vol. 11, pp. 69005–69021, 2023, doi: 10.1109/access.2023.3285481.
- [6] M. Kaur, D. Singh, K. Sun, and U. Rawat, "Color image encryption using non-dominated sorting genetic algorithm with local chaotic search based," *Future Generation Computer Systems*, vol. 107, pp. 333–350, Jun. 2020, doi: 10.1016/j.future.2020.02.029.
- [7] W. Alexan, Y.-L. Chen, L. Y. Por, and M. Gabr, "Hyperchaotic maps and the single neuron model: a novel framework for chaos-based image encryption," *Symmetry*, vol. 15, no. 5, May 2023, doi: 10.3390/sym15051081.
- [8] W. J. Jun and T. S. Fun, "A new image encryption algorithm based on single S-box and dynamic encryption step," *IEEE Access*, vol. 9, pp. 120596–120612, 2021, doi: 10.1109/ACCESS.2021.3108789.
- [9] M. Ramzan, T. Shah, M. M. Hazzazi, A. Aljaedi, and A. R. Alharbi, "Construction of S-Boxes using different maps over elliptic curves for image encryption," *IEEE Access*, vol. 9, pp. 157106–157123, 2021, doi: 10.1109/access.2021.3128177.
- [10] M. Jarjar, S. Najah, K. Zenkour, and S. Hraoui, "Further improvement of the HILL method applied in image encryption," in *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Apr. 2020, pp. 1–6, doi: 10.1109/IRASET48871.2020.9092046.
- [11] M. Kumari, S. Gupta, and P. Sardana, "A survey of image encryption algorithms," *3D Research*, vol. 8, no. 4, Dec. 2017, doi: 10.1007/s13319-017-0148-5.
- [12] A. JarJar, "Vigenere and genetic cross-over acting at the restricted ASCII code level for color image encryption," *Medical and Biological Engineering and Computing*, vol. 60, no. 7, pp. 2077–2093, Jul. 2022, doi: 10.1007/s11517-022-02566-4.
- [13] Y. Qobbi, A. Jarjar, M. Essaid, and A. Benazzi, "New image encryption scheme based on dynamic substitution and hill cipher," in *WITS 2020*, Springer Singapore, 2021, pp. 797–808.
- [14] M. Jarjar, S. Hraoui, S. Najah, and K. Zenkour, "New technology of color image encryption based on chaos and two improved Vigenere steps," *Multimedia Tools and Applications*, vol. 81, no. 17, pp. 24665–24689, Mar. 2022, doi: 10.1007/s11042-022-12750-1.




- [15] S. Rajendran, M. Marjuk, R. D. Duraisamy, S. Sridharan, and M. Doraipandian, "An image crypt model based on enhanced two-dimensional sine-cosine chaotic map," *Multidisciplinary Science Journal*, vol. 6, Dec. 2023, doi: 10.31893/multiscience.2024ss0106.
- [16] M. Jarjar, A. Jarjar, A. Abid, S. El Kaddouhi, M. Kattass, and A. Benazzi, "An altered Vigenère circuit used in a genetic cross again for encryption of medical images," Apr. 2023, doi: 10.21203/rs.3.rs-2766159/v1.
- [17] P. Murali, G. Niranjana, A. J. Paul, and J. S. Muthu, "Domain-flexible selective image encryption based on genetic operations and chaotic maps," *The Visual Computer*, vol. 39, no. 3, pp. 1057–1079, Mar. 2023, doi: 10.1007/s00371-021-02384-z.
- [18] H. K. Zghair, H. A. Ismael, and A. A.-H. Al-Shamery, "Image scrambler based on novel 4-D hyperchaotic system and magic square with fast Walsh–Hadamard transform," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 11, no. 6, pp. 3530–3538, Dec. 2022, doi: 10.11591/eei.v11i6.4339.
- [19] A. Abdellah, M. Jarjar, A. Benazzi, A. Jarjar, Y. Qobbi, and S. El Kaddouhi, "Vigenère implemented in two chaotic Feistel laps for medical images encryption followed by genetic mutation," in *Lecture Notes in Networks and Systems*, Springer International Publishing, 2023, pp. 824–830.
- [20] M. Jarjar, A. Abid, Y. Qobbi, S. El Kaddouhi, A. Benazzi, and A. Jarjar, "An image encryption scheme based on DNA sequence operations and chaotic system," in *Lecture Notes in Networks and Systems*, Springer International Publishing, 2023, pp. 191–198.
- [21] J. Chen, Z. Zhu, L. Zhang, Y. Zhang, and B. Yang, "Exploiting self-adaptive permutation–diffusion and DNA random encoding for secure and efficient image encryption," *Signal Processing*, vol. 142, pp. 340–353, Jan. 2018, doi: 10.1016/j.sigpro.2017.07.034.
- [22] G. Ye and X. Huang, "An efficient symmetric image encryption algorithm based on an intertwining logistic map," *Neurocomputing*, vol. 251, pp. 45–53, Aug. 2017, doi: 10.1016/j.neucom.2017.04.016.
- [23] Z. B. Madouri, N. Hadj Said, and A. Ali Pacha, "A new pseudorandom number generator based on chaos in digital filters for image encryption," *Journal of Optics*, Jan. 2024, doi: 10.1007/s12596-023-01606-y.
- [24] Y. Wang, X. Leng, C. Zhang, and B. Du, "Adaptive fast image encryption algorithm based on three-dimensional chaotic system," *Entropy*, vol. 25, no. 10, Sep. 2023, doi: 10.3390/e25101399.
- [25] 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 (IJECE)*, vol. 13, no. 6, pp. 6952–6963, Dec. 2023, doi: 10.11591/ijece.v13i6.pp6952-6963.
- [26] S. M. Kareem, A. Al-Adhami, and A. M. S. Rahma, "An improvement for CAST-128 encryption based on magic square and matrix inversion," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 13, no. 1, pp. 377–387, Feb. 2024, doi: 10.11591/eei.v13i1.5340.
- [27] M. Vijayakumar and A. Ahilan, "An optimized chaotic S-box for real-time image encryption scheme based on 4-dimensional memristive hyperchaotic map," *Ain Shams Engineering Journal*, Jan. 2024, doi: 10.1016/j.asej.2023.102620.
- [28] 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 (IJECE)*, vol. 13, no. 2, pp. 1903–1913, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1903-1913.
- [29] S. R. Victor Juvvanapudi, P. Rajesh Kumar, and K. V. V. Satyanarayana Reddy, "Hybrid chaotic map with L-shaped fractal Tromino for image encryption and decryption," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, pp. 389–397, Feb. 2024, doi: 10.11591/ijece.v14i1.pp389-397.
- [30] A. Weber, "The USC-SIPI image database," USC-SIPI, <https://sipi.usc.edu/database/database.php?volume=misc> (accessed Feb. 21, 2024).
- [31] E. Moya-Albor, A. Romero-Arellano, J. Brieva, 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.
- [32] Y. Chen, S. Xie, and J. Zhang, "A hybrid domain image encryption algorithm based on improved henon map," *Entropy*, vol. 24, no. 2, Feb. 2022, doi: 10.3390/e24020287.
- [33] R. Qumsieh, M. Farajallah, and R. Hamamreh, "Joint block and stream cipher based on a modified skew tent map," *Multimedia Tools and Applications*, vol. 78, no. 23, pp. 33527–33547, Aug. 2019, doi: 10.1007/s11042-019-08112-z.
- [34] X. Zhang and J. Tian, "Multiple-image encryption algorithm based on genetic central dogma," *Physica Scripta*, vol. 97, no. 5, Apr. 2022, doi: 10.1088/1402-4896/ac66a1.
- [35] 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 (IJECE)*, vol. 13, no. 2, pp. 1744–1756, Apr. 2023, doi: 10.11591/ijece.v13i2.pp1744-1756.
- [36] A. H. Khaleel and I. Q. Abduljaleel, "Secure image hiding in speech signal by steganography-mining and encryption," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 21, no. 3, pp. 1692–1703, Mar. 2021, doi: 10.11591/ijeecs.v21i3.pp1692-1703.

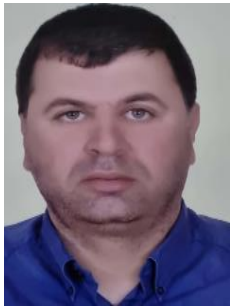
## BIOGRAPHIES OF AUTHORS






**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).






**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](mailto:hassan.tabti1@usmba.ac.ma).






**Mourad Kattass**    received a master's degree in robotics and embedded systems from the Faculty of Science and Technique, Abdelmalek Essaadi University, Morocco, in 2020 and a Bachelor's degree in computer science, electronic, electrotechnical, and automatic from Sidi Mohammed Ben Abdellah University, Morocco, in 2006, respectively. Currently, Ph.D. degrees in mathematics and computer science from Mohammed First University, Oujda, Morocco. His research interests include computer science. He can be contacted at email: [mourad.kattass@ump.ac.ma](mailto:mourad.kattass@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: [abdoujjar@gmail.com](mailto:abdoujjar@gmail.com).



**Abdelhamid Benazzi**    received a master's degree in fundamental mathematics from Franche Compté Besonçon University, France, in 1987. Professor of mathematics and searcher in computer science from Mohammed First University, Oujda, Morocco. His research interests include computer science. He can be contacted at email: [a.benazzi@ump.ac.ma](mailto:a.benazzi@ump.ac.ma).