



Hybrid Vigenère-Hill Approach for Color Image Encryption

Mourad Kattass^{*}, Hicham Rrghout^{*}, Hamid El Bourakkadi^{*}, Abdellah Abid^{*}, Abdellatif Jarjar^{*},
Abdelhamid Benazzi^{*}

MATSI Laboratory, Mohammed First University, Oujda 60000, Morocco

Corresponding Author Email: mourad.kattass@ump.ac.ma

Copyright: ©2025 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijss.150605>

ABSTRACT

Received: 10 May 2025

Revised: 12 June 2025

Accepted: 23 June 2025

Available online: 30 June 2025

Keywords:

chaotic maps, confusion and diffusion, dynamic S-box, Hill cipher, image encryption, encryption process, S-box, Vigenère cipher

This paper introduces an enhanced encryption scheme for color images, combining improved Vigenère and Hill cipher techniques. Our approach leverages two carefully selected chaotic maps, exploiting their extreme sensitivity to initial conditions for cryptographic security. The encryption process begins with RGB channel separation and vector conversion, followed by initial confusion operations generating a partially encrypted image vector. This vector is then divided into 3-pixel subblocks for subsequent processing. Each block undergoes multi-stage encryption controlled by a binary vector, employing three expanded substitution tables with optimized confusion-diffusion functions. These functions operate sequentially across pixels with chaining mechanisms between adjacent pixels. The modified Hill cipher then processes each block using an invertible matrix combined with dynamic translation vectors, effectively addressing the linearity limitations of traditional Hill cipher implementations. To enhance security, we implement an inter-block diffusion mechanism that dynamically links each block's final encrypted pixel with the next block's initial pixel through a specialized diffusion function. This design significantly strengthens avalanche effects while providing robust resistance against differential attacks. Tests on a diverse set of randomly chosen color images yielded statistical (histogram, correlation, entropy) and differential (UACI, NPCR) metrics meeting international standards, confirming our cryptosystem's robustness against known attacks.

1. INTRODUCTION

With the rapid advancement of hardware and software technologies, ensuring the secure transmission of confidential data has become a critical challenge. One of the most effective methods is cryptography, which transforms clear data into unreadable data. Encryption is a method that makes text or images unintelligible except to authorized persons. Satellite or medical image data containing sensitive, secret, and confidential information must be encrypted before being securely transferred, even through public channels [1-3]. With advances in mathematics, cryptography quickly became an independent science in the field of security. The emergence of several techniques such as Hill [4, 5], Vigenère [6-8] and others has been widely observed.

The authors [1, 8] improved the conventional Vigenère method by constructing new large S-boxes with the implementation of new substitution and diffusion functions using several pseudo-random vectors. El Bourakkadi et al. [9] proposed an enhancement of the classical Vigenère method by incorporating dynamic affine functions for the construction of the substitution function. Furthermore, the authors of article [10] presented the use of the improved Vigenere technique combined with genetic crossover adapted for image encryption. The study [4] proposed a method for color image encryption based on the use of chaos and an improvement of

the classical Hill method. The principle of this method involves replacing the linear transformation with an affine transformation ensured by an invertible matrix of order (3×3) with a particular form and a fixed translation vector, which allows overcoming the problem of linearity in the classical system. However, by changing the variable, this transformation becomes linear. Moreover, to overcome the problem related to the encryption key size, many image encryption approaches have improved classical systems for better adaptation to encrypting large volumes of data with high redundancy and high correlation by relying on chaos theory [3, 5, 11-16].

The rapid development of chaos theory and the straightforward use of chaotic maps within a cryptosystem encourage researchers to explore the integration of multiple chaotic maps into an encryption algorithm. This aims to maximize the size of the secret key, thereby protecting the system against brute-force attacks. Additionally, adhering to Shannon's principles [16], a good cryptosystem must incorporate confusion, diffusion, and, in the worst case, permutation.

The conventional Vigenère technique uses a static and public substitution table of size 26×26 , which exposes it to statistical and frequency attacks. Moreover, in the absence of any encryption mode of operation, this method remains vulnerable to differential attacks. Furthermore, the classical

Hill technique employs a static encryption matrix that is easy to invert and of small size. This linear technique exposes weaknesses against statistical and brute-force attacks. Additionally, the lack of chaining mechanisms leaves it open to differential analysis.

Most existing works implement Hill cipher and Vigenère cipher independently with a single chaining mechanism. Our approach couples both enhanced techniques (Hill and Vigenère) via dual-stage chaining to amplify the avalanche effect and strengthen resistance against differential cryptanalysis.

The key contributions of our approach:

Improvement of the classical Vigenère technique and first diffusion:

- Generation of large pseudo-random substitution tables.
- Application of the most frequently employed chaotic maps in the field of cryptography.
- New substitution functions executed under the control of several binary decision vectors.
- Implementation of an initial chaining (first diffusion) within a block.

Improved affine Hill transformation:

- Application of an improved affine transformation on the obtained block.
- Use of a full 3×3 encryption matrix.
- Construction of the matrix from the product of:
-A matrix with a particular form.
-An invertible lower triangular matrix.
- A dynamic translation vector is employed to address the issue of linearity.

Second diffusion:

Application of diffusion between the last pixel of the encrypted block and the first pixel of the following block.

The rest of the document is structured into the following sections: The first section describes the proposed method, explaining the development of pseudo-random vectors and the construction of substitution tables necessary for the implementation of the improved Vigenère technique accompanied by that of Hill. The second section analyzes the results obtained from testing multiple images using our new approach, followed by a comparison with other similar works and discussions. Finally, the third section summarizes the results and proposes research perspectives.

2. PROPOSED METHOD

Our method, grounded in chaos theory, begins with the selection of two chaotic maps. Second, a generation of several pseudo-random vectors for the implementation of confusion and diffusion functions [9, 10, 15, 17-19]. Finally, an encryption and decryption process will be implemented, followed by a simulation and comparison study.

2.1 Theoretical foundations

2.1.1 Choice of chaotic maps

All encryption parameters in our method are derived from two widely used chaotic maps in cryptography. They are chosen by their extreme sensitivity to initial conditions and their ease of configuration. These chaotic maps used are:

The logistics map (LM_n). The logistic map defines a sequence through the iterative application of a nonlinear second-order polynomial function [20]. This sequence

presents a chaotic aspect under the conditions of Eq. (1).

$$\begin{cases} LM_0 \in [0.5, 1] \mu \in [3.57, 4] \\ LM_{n+1} = \mu LM_n (1 - LM_n) \end{cases} \quad (1)$$

The PWLCM map (PM_n). The PWLCM is defined by a first-degree stepwise polynomial [21]. This sequence exhibits chaotic behavior under the conditions of Eq. (2).

$$PM_n = \begin{cases} PM_n \in [0,1] \text{ et } p \in [0,0.5] \\ \frac{PM_{n-1}}{p} & \text{if } 0 \leq PM_{n-1} < p \\ \frac{PM_{n-1}-p}{0.5-p} & \text{if } p \leq PM_{n-1} < 0.5 \\ f(1 - PM_{n-1}, p) & \text{elsewhere} \end{cases} \quad (2)$$

2.1.2 Generation of pseudo-random vectors (sub-keys)

The two chaotic maps are used to generate three pseudo-random vectors (C1), (C2), and (C3) with coefficients in $(\mathbb{Z}/256\mathbb{Z})$ for the confusion process, and two vectors (B1) and (B2) in $(\mathbb{Z}/2\mathbb{Z})$ for event control. These vectors constitute the sub-keys of our algorithm.

Construction of sub-keys. The pseudo-random vectors (C1), (C2), and (C3) with coefficients in the ring $(\mathbb{Z}/256\mathbb{Z})$ are considered as sub-keys by our system. These vectors are generated by Algorithm 1. Our encryption system relies on the dynamic generation of two binary control vectors (denoted as (B1) and (B2)). These vectors are synthesized via Algorithm 2.

Algorithm 1. Generation of pseudo-random vector

Input: Chaotic sequences lm and pm

Output: Chaotic vectors C1, C2 and C3

Begin

 // Confusion vectors

 for $k \leftarrow 1$ to $3nm$

 // $E(x)$ means the integer part of x

$C1(i) \leftarrow [E(\sup(lm(k), pm(k)).10^{11}) \bmod 251] + 4$

$C2(i) \leftarrow [E(((lm(k) + 2 * pm(k))/3).10^{11}) \bmod 252] +$

 3

$C3(k) \leftarrow [E(|lm(k) - pm(k)|.10^{10}) \bmod 253] + 2$

 endFor

End

Algorithm 2. Generation of two pseudo-random binary vectors, denoted as B1 and B2

 //Construction of binary vectors

Input: Chaotic sequences lm and pm

Output: Control vectors: B1 and B2

Begin

 for $i \leftarrow k$ to $3nm$

 if $pl(k) > pm(k)$ then

$B1(k) \leftarrow 0$

 else $B1(k) \leftarrow 1$

 endif

 if $lm(k) > 0.5$ then

$B2(k) \leftarrow 0$

 else $B2(k) \leftarrow 1$

 endif

 endFor

End

2.1.3 Generation of S-boxes

Our algorithm uses three new large substitution tables (SB1), (SB2), and (SB3). These tables, each of dimension (256×256) , with coefficients in the ring $(\mathbb{Z}/256\mathbb{Z})$, are constructed by three different procedures.

1st S-BOX (SB1). The substitution matrix SB1 is

constructed through the following steps:

Step 1: Its first row, denoted P1, is derived by sorting the initial 256 values of vector C1 in ascending order.

Step 2: Each row k (where $k \geq 1$) is constructed by cyclically shifting the previous row (k-1) by $C_1(k)$ or $C_2(k)$ positions, determined by the binary selector $B_2(k)$. This dynamic S-box structure is generated via Algorithm 3.

Algorithm 3. Construction of the first S-box (SB1)

Input: Chaotic vectors: C1 and C2

Control vector: B2

Permutation Vector: P1

Output: Substitution Matrix: SB1 of size (256,256)

Begin

for k ← 1 to 256 // First line

SB1(1,k) ← P1(k)

endFor

for k ← 2 to 256 // Next lines

for l ← 1 to 256

if $B_2(k) = 0$ then

SB1(k,l) ← SB1(k-1, mod(l + C1(k),256))

else

SB1(k,l) ← SB1(k-1, mod(l + C2(k),256))

endif

endIf

endFor

endFor

End

An example of the construction of the S-box (SB1) is illustrated in Tables 1 and 2.

Table 1. Example of permutation construction

| Rank | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------------------------------|---|---|---|---|---|---|---|---|
| The first eight values of (C1) | 5 | 8 | 7 | 5 | 6 | 4 | 2 | 4 |
| Ascending sort | 5 | 8 | 7 | 4 | 6 | 3 | 1 | 2 |
| Permutation | $P1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 8 & 7 & 4 & 6 & 3 & 1 & 2 \end{pmatrix}$ | | | | | | | |

Table 2. Example of SB1 construction

| SB1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | C1 | C2 | B2 |
|-----|---|---|---|---|---|---|---|---|----|----|----|
| 1 | 5 | 8 | 7 | 4 | 6 | 3 | 1 | 2 | 4 | 5 | 0 |
| 2 | 7 | 4 | 6 | 3 | 1 | 2 | 5 | 8 | 5 | 2 | 1 |
| 3 | 2 | 5 | 8 | 7 | 4 | 6 | 3 | 1 | 3 | 5 | 1 |
| 4 | 6 | 3 | 1 | 2 | 5 | 8 | 7 | 4 | 5 | 2 | 0 |
| 5 | 7 | 4 | 6 | 3 | 1 | 2 | 5 | 8 | 6 | 2 | 0 |
| 6 | 1 | 2 | 5 | 8 | 7 | 4 | 6 | 3 | 2 | 4 | 1 |
| 7 | 8 | 7 | 4 | 6 | 3 | 1 | 2 | 5 | 3 | 1 | 0 |
| 8 | 3 | 1 | 2 | 5 | 8 | 7 | 4 | 6 | 1 | 4 | 1 |

2nd S-BOX (SB2). The construction of the substitution matrix (SB2) is determined by the steps below:

Step 1: The generation of two permutations P1 and P2 of size 1×256 using Algorithms 4 and 5, denoted P1[C2, 256, 256] and P2[C3, 256, 256] [8].

Step 2: The first two rows are initialized by P1 and P2, respectively.

Step 3: Each row i (for $k > 2$) of the substitution box SB2 is computed as the composition, in the functional sense, of row (k-1) with row (k-2), or of row (k-2) with row (k-1), based on the corresponding bit in the control vector B1. The complete procedure is described in Algorithm 6.

Table 3 shows an example of SB2 generation over the ring $\mathbb{Z}/8\mathbb{Z}$ controlled by B1.

Algorithm 4. P1 [C2, 256, 256]

Input: Chaotic vector: C2

Output: Permutation Vector P1

Begin

c ← 0

for k ← 1 to 256

for l ← 1 to 256

if $C_2(l) = k$ then

P1(k) ← c

c + +

endif

endFor

endFor

End

Algorithm 5. P2 [C3, 256, 256]

Input: Chaotic vector: C3

Output: Permutation Vector P2

Begin

c ← 0

for k ← 1 to 256

for l ← 1 to 256

if $C_3(l) = k$ then

P2(l) ← c

c + +

endif

endFor

endFor

End

Algorithm 6. Construction of the substitution table (SB2)

Input: Control vector: B1

Permutation Vectors P1 and P2

Output: Substitution Matrix SB2 of size (256,256)

Begin

for k ← 1 to 256

//The first 2 lines

SB2(1,k) ← P1(k)

SB2(2,i) ← P2(i)

endFor

for ik ← 3 to 256

//the other lines

for l ← 1 to 256

if $B_1(k) = 0$ then

SB2(k,l) ← SB2(k-1, SB2(k-2, l))

else

SB2(k,l) ← SB2(k-2, SB2(k-1, l))

endif

endFor

endFor

End

Table 3. Example of SB2 construction

| SB2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | B1 |
|------------|---|---|---|---|---|---|---|---|----|
| P1 | 1 | 5 | 8 | 7 | 4 | 6 | 3 | 1 | 2 |
| P2 | 2 | 3 | 2 | 5 | 8 | 1 | 7 | 4 | 6 |
| P3 = P1oP2 | 3 | 7 | 8 | 6 | 2 | 5 | 1 | 4 | 3 |
| P4 = P2oP3 | 4 | 4 | 6 | 7 | 2 | 1 | 3 | 8 | 5 |
| P5 = P4oP3 | 5 | 5 | 1 | 7 | 3 | 4 | 2 | 6 | 8 |
| P6 = P4oP5 | 6 | 8 | 4 | 7 | 2 | 5 | 3 | 1 | 6 |
| P7 = P6oP5 | 7 | 8 | 4 | 7 | 2 | 5 | 3 | 1 | 6 |
| P8 = P7oP6 | 8 | 5 | 8 | 1 | 7 | 2 | 4 | 3 | 6 |

3rd S-BOX. The construction of the new substitution matrix (SB3) is carried out according to the following procedure:

- The first row, referred to as permutation P1, is derived by performing a broad ascending sort of the first 256 values from the chaotic sequence lm ;

- The second row, permutation P2, results from a strict

ascending sort of the first 256 values extracted from the chaotic sequence pl;

The third row, permutation P3, is obtained through a broad ascending sort of the initial 256 values of the pseudo-random vector C1;

•Each subsequent row k (for k > 3) is computed as the functional composition of rows (k-2) and (k-1), or rows (k-1) and (k-3), depending on the corresponding value in the control vector B2.

This construction is given by Algorithm 7.

Algorithm 7. Construction of the substitution table (SB3)

Input: Control vector B2

Permutation Vectors P1, P2 and P3

Output: Substitution Matrix SB3 of size (256,256)

Begin

//3 first lines

for k ← 1 to 256

SB3(1,k) ← P1(k)

SB3(2,k) ← P2(k)

SB3(3,k) ← P3(k)

endFor

//Next lines

for k ← 4 to 256

for l ← 1 to 256

if B2(k) = 0 then

SB3(k,l) ← SB3(k-2, SB3(k-1,l))

else

SB3(k,l) ← SB3(k-1, SB3(k-3,l))

endif

endFor

endFor

End

Table 4 shows an example of SB3 generation over the ring $\mathbb{Z}/8\mathbb{Z}$ controlled by B2.

Table 4. Example of SB3 construction

| SB3 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | B2 |
|------------|---|---|---|---|---|---|---|---|----|
| P1 | 1 | 5 | 8 | 7 | 4 | 6 | 3 | 1 | 1 |
| P2 | 2 | 3 | 2 | 5 | 8 | 1 | 7 | 4 | 0 |
| P3 | 3 | 6 | 4 | 2 | 5 | 3 | 1 | 8 | 1 |
| P4 = P2oP3 | 4 | 7 | 8 | 2 | 1 | 5 | 3 | 6 | 0 |
| P5 = P3oP4 | 5 | 8 | 7 | 4 | 6 | 3 | 2 | 1 | 5 |
| P6 = P5oP3 | 6 | 2 | 6 | 7 | 3 | 4 | 8 | 5 | 1 |
| P7 = P6oP4 | 7 | 5 | 1 | 6 | 2 | 4 | 7 | 8 | 3 |
| P8 = P6oP7 | 8 | 4 | 2 | 8 | 6 | 3 | 5 | 1 | 7 |
| | | | | | | | | | 0 |

2.1.4 Substitution function (Fv)

This substitution function is a significant improvement over the classical Vigenere function and is analytically given by the expression in Eq. (3): Let Y(k) be the transformed pixel X(k) by the function Fv using the three tables (SB1), (SB2) and (SB3).

2.1.5 Diffusion function (Dv)

To deal with any differential attack, we apply a diffusion function defined by Eq. (4):

$$Y(k) = Fv(X(k)) = \begin{cases} SB1 \left(\begin{matrix} C1(k), \\ C2(k); \\ SB3(C3(k); X(k)) \end{matrix} \right) & \text{if } B2(k) \\ SB2 \left(\begin{matrix} C3(k), \\ C2(k); \\ SB3(C1(k); X(k)) \end{matrix} \right) & \text{elsewhere} \end{cases} \quad (3)$$

$$Dv(Y(k)) = SB3(C1(k); Y(k)) \oplus X(k+1) \quad (4)$$

2.1.6 Construction of the Hill matrix (HM)

A matrix M is invertible in $(\mathbb{Z}/256\mathbb{Z})$ if and only if $\det(M)$ is odd. The secret Hill encryption matrix (HM) is constructed as the product of two matrices, A and B, of the same dimension, each defined by the following specific structures:

$$A = \begin{pmatrix} 1 & a1 & a3 \\ a2 & 1 + a1a2 & a4 \\ 0 & 0 & 2a5 + 1 \end{pmatrix} \quad B = \begin{pmatrix} b1 & 0 & 0 \\ b2 & b3 & 0 \\ b4 & b5 & b6 \end{pmatrix} \quad HM = A * B$$

$\det(HM) = \det(A) * \det(B) = (2a5 + 1) * b1 * b3 * b6$, Since $(2a5 + 1)$, $b1$, $b3$, and $b6$ are all odd, HM is an invertible matrix in $(\mathbb{Z}/256\mathbb{Z})$.

The inverse matrix $(HM)^{-1}$ is given by the following expression: $(HM)^{-1} = B^{-1} * A^{-1}$. To overcome the problem of linearity in the classical transformation, a translation has been implemented using the XOR operation between vector C3 and matrix HM. For brute force attacks, the number of possibilities is:

•Number of possible choices for matrix HM: $(2^8)^5 * (2^8)^6 = 2^{88}$

•Number of possible choices for vector C3: $(2^8)^{3nm}$

•The total number of possibilities for $HM \oplus C3$ is: $2^{88} * (2^8)^{3nm} = 2^{88+24nm} \gg 2^{100}$

2.2 Hybrid image encryption algorithm design

2.2.1 Encryption process

The encryption process is based on the following steps:

Step 1: Pseudo-random vectorization

After loading the original image of size (N×M) and extracting the three-color channels R, G, and B (1×NM) into three vectors (Vr), (Vg), and (Vb) respectively, we proceed with controlled concatenation of these three vectors by the binary vector B1, while adding confusion with pseudo-random vectors (C1), (C2), and (C3) as shown in Figure 1.

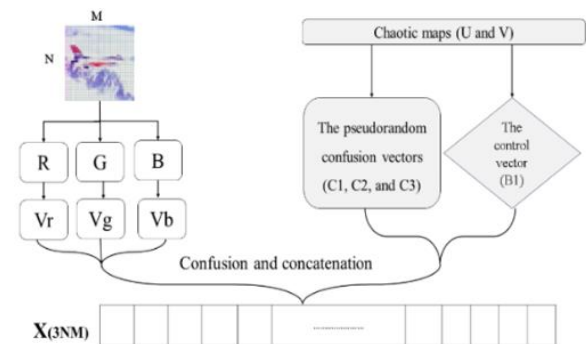


Figure 1. Diagram of original image preparation

Mathematically, this phase is described in Algorithm 8.

The first step is light encryption, where the encrypted image is protected from static and frequency attacks. To overcome the problem of differential attacks, a second round of encryption is considered.

Step 2: Initialization Vector (Iv)

The initialization value serves exclusively to modify the initial pixel and to trigger the encryption process. This value, which is appended to the lightly encrypted image, is computed according to Algorithm 9.

Algorithm 8. Algorithm for original image vectorization

Input: Original image channels Vr, Vg and Vb of size (1, nm)
Control vector: B1
Chaotic Vectors: C1, C2 and C3
Output: Vector image X of size (1,3nm)
Begin
for $l \leftarrow 1$ to nm
if $B1(l) = 0$ then
 $X(3l-2) \leftarrow Vr(l) \oplus C1(l)$
 $X(3l-1) \leftarrow Vg(l) \oplus C2(l)$
 $X(3l) \leftarrow Vb(l) \oplus C3(l)$
else
 $X(3l-2) \leftarrow Vr(l) \oplus C2(l)$
 $X(3l-1) \leftarrow Vg(l) \oplus C3(l)$
 $X(3l) \leftarrow Vb(l) \oplus C1(l)$
endif
endfor
End

Algorithm 9. Calculation of the initialization value

Input: Vector image X of size (1, 3nm)
Chaotic Vectors C2 and C3, Control vector B2
Output: integer Iv

```

Begin
Iv  $\leftarrow$  0
for  $k \leftarrow 2$  to  $3nm$ 
  if  $B2(k) = 0$  then
    Iv  $\leftarrow$  Iv  $\oplus$  X(k)  $\oplus$  C2(k)
  else
    Iv  $\leftarrow$  Iv  $\oplus$  X(k)  $\oplus$  C3(k)
  endif
endfor
end

```

Step 3: Encryption diagram

After generating a lightly encrypted image vector, it is subdivided into 3-pixel sub-blocks. Each sub-block undergoes enhanced confusion and diffusion operations, guided by a binary control vector and three large substitution tables. These operations are applied sequentially to each pixel, with each step dynamically linked to the next pixel to ensure nonlinear propagation of changes; the first component of this process is illustrated in Figure 2 (First block) and Algorithm 10. After encrypting each sub-block, the Hill method is applied using an invertible matrix combined with dynamic translation vectors. This hybrid approach overcomes the linearity limitations inherent in classical Hill transformations, thereby introducing adaptability and increased resistance to linear cryptanalysis; the first component of this process is illustrated in Figure 3 (First block). A diffusion function dynamically links the last pixel of an encrypted sub-block to the first pixel of the next sub-block. This inter-block dependency ensures that even minor changes in the input propagate nonlinearly throughout the entire encrypted image. The overall encryption process is illustrated in Figure 4.

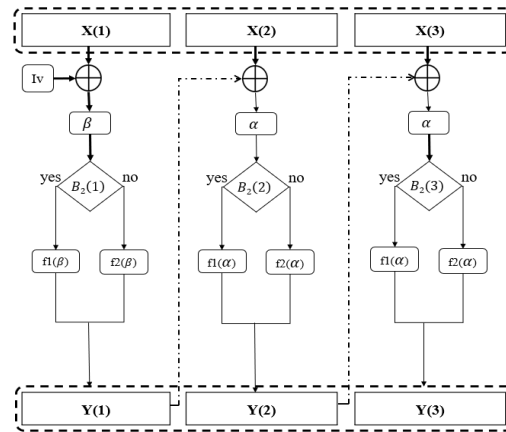


Figure 2. The first component of the enhanced Vigenère encryption mechanism

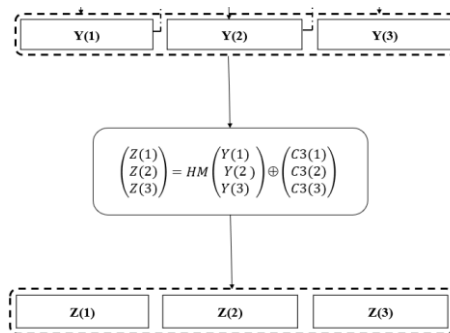


Figure 3. The first component of the enhanced Hill encryption mechanism

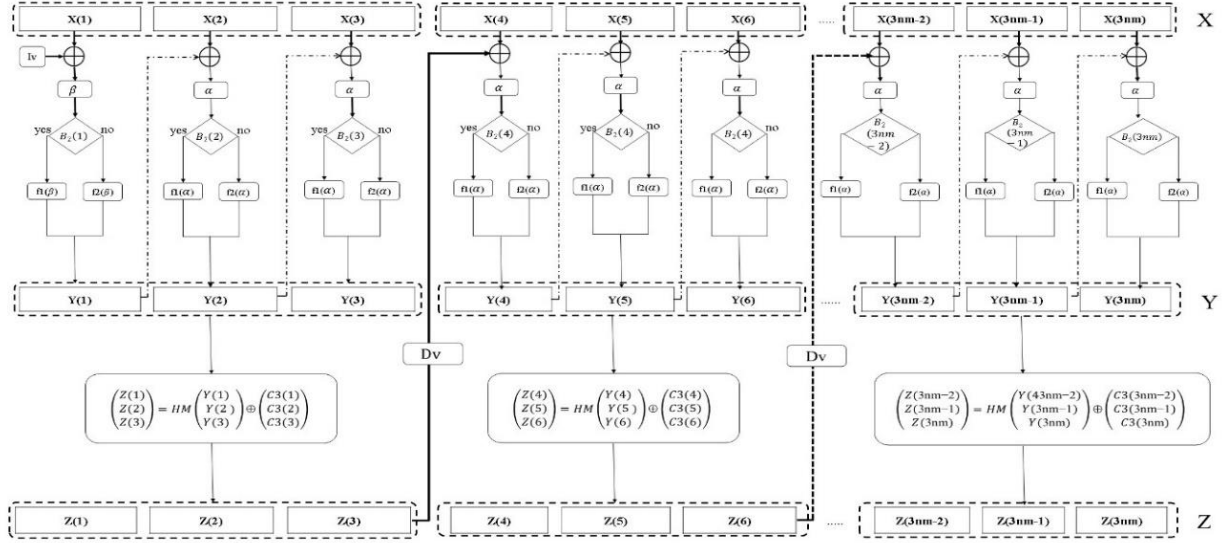


Figure 4. The complete encryption process

Algorithm 10. The enhanced Vigenère phase of the encryption process

//First pixel encryption

Input: Substitution matrices of size (256, 256) SB1, SB2, and SB3

Chaotic vectors: C1, C2 and C3

Vector image X of size (1, 3nm), Control vector B2

Initialization value Iv

Output: Encrypted matrix image: Y of size (1, 3nm)

Begin

$\beta \leftarrow X(1) \oplus Iv$

if $B2(1) = 0$ then

$Y(0) \leftarrow SB1(C1(1), SB2(C2(1); SB3(C3(1); \beta)))$

// $f1(\beta)$

else

$Y(1) \leftarrow SB2(C3(1), SB1(C2(1); SB3(C1(1); \beta)))$ //

$f2(\beta)$ endIf

//Encryption of the next pixels

for $k \leftarrow 2$ to $3nm$

$\alpha \leftarrow X(k) \oplus Y(k-1)$

if $B2(k) = 0$ then

$Y(k) \leftarrow SB1(C1(k), SB2(C2(k); SB3(C3(k); \alpha)))$

else $Y(k) \leftarrow SB2(C3(k), SB1(C2(k); SB3(C1(k); \alpha)))$

endIf

endFor

End

$$\begin{pmatrix} Z(3h) \\ Z(3h+1) \\ Z(3h+2) \end{pmatrix} = HM \begin{pmatrix} Y(3h) \\ Y(3h+1) \\ Y(3h+2) \end{pmatrix} \oplus \begin{pmatrix} C3(3h) \\ C3(3h+1) \\ C3(3h+2) \end{pmatrix} \quad (5)$$

Then:

$$\begin{pmatrix} Y(3h) \\ Y(3h+1) \\ Y(3h+2) \end{pmatrix} = (HM)^{-1} \left(\begin{pmatrix} Z(3h) \\ Z(3h+1) \\ Z(3h+2) \end{pmatrix} \oplus \begin{pmatrix} C3(3h) \\ C3(3h+1) \\ C3(3h+2) \end{pmatrix} \right) \quad (6)$$

It is necessary to calculate $(HM)^{-1}$ in the ring $\mathbb{Z}/256\mathbb{Z}$. The inverse of matrix A is determined by solving the following system of equations:

$$\begin{pmatrix} 1 & a_1 & a_3 \\ a_2 & 1+a_1a_2 & a_4 \\ 0 & 0 & 2a_5+1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix}$$

$$A^{-1} = \begin{pmatrix} (1+a_1a_2) & -a_1 & -[a_3\alpha(1+a_1a_2)-a_1-a_4\alpha]z' \\ a_2 & 1 & -(a_4\alpha-a_2a_3\alpha)z' \\ 0 & 0 & \alpha \end{pmatrix} \pmod{256}$$

where, α being the inverse of $(2a_5+1)$ in $\mathbb{Z}/256\mathbb{Z}$.

We follow the same process to determine the inverse of matrix B. We solve the following system of equations:

$$\begin{cases} b_1x = x' \\ b_2x + b_3y = y' \\ b_4x + b_5y + b_6z = z' \end{cases} \text{ then}$$

$$B^{-1} = \begin{pmatrix} \beta & 0 & 0 \\ -\beta b_2\gamma & \gamma & 0 \\ \beta b_2\gamma b_5\delta - \beta b_4\delta & -\gamma b_5\delta & \delta \end{pmatrix} \pmod{256}$$

where, β , γ and δ being the inverses of b_1 , b_3 , and b_6 in $\mathbb{Z}/256\mathbb{Z}$.

The algorithm describes a cryptographic scheme for encrypting an image using a combination of substitution matrices, chaotic vectors, and control vectors.

2.2.2 Decryption process

Our cryptosystem is a symmetric encryption scheme, which necessitates initiating the decryption process from the final block and applying, at each step, the inverse functions of those used during encryption. The encrypted image is first converted into a one-dimensional vector and segmented into blocks of three pixels. The decryption procedure then proceeds as follows:

Inverse of the Hill transformation. For each $h \in [1, nm]$, we have:

The inverse of the Vigenere transformation. The substitution matrix for this transformation is constructed using Algorithm 11.

Algorithm 11. Calculation of the inverse of the Vigenère transformation

Input: 256×256 S-boxes: SB1, SB2, and SB3
Output: 256×256 inverse S-boxes: D1, D2 and D3
Begin
for $k \leftarrow 1$ to 256
for $l \leftarrow 1$ to 256
 $D1(k, S1(k, l)) \leftarrow l$
 $D2(k, S2(k, l)) \leftarrow l$
 $D3(k, S3(k, l)) \leftarrow l$
Next j, i
End

Construction of the inverse confusion function. According to the classic Vigenere technique we have:

$$\text{If } z = V(cl(k), X(k)) \text{ then } X(k) = W(cl(k), z)$$

where, W is the matrix of the inverse transformation of Vigenere.
we have:

$$\begin{aligned}
 Y(k) &= Fv(X(k)) = \\
 &S1(C1(k), S2(C2(k); S3(C3(k); X(k)))) \\
 &\quad \text{if } B2(k) = 0 \\
 &S2(C3(k), S1(C2(k); S3(C1(k); X(k)))) \\
 &\quad \text{elsewhere} \\
 &\text{if } B2(k) = 0 \text{ then :} \\
 &(Fv(Y(k)))^{-1} = X(k) = \\
 &D3(C3(k); D2(C2(k); D1(C1(k); Y(k)))) \\
 &\text{else} \\
 &(Fv(Y(k)))^{-1} = X(k) = \\
 &D3(C1(k); D1(C2(k); D2(C3(k); Y(k))))
 \end{aligned}
 \tag{7}$$

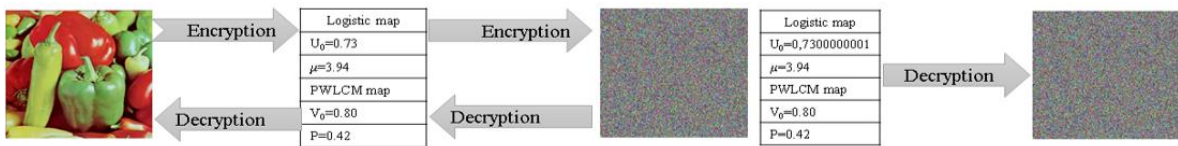
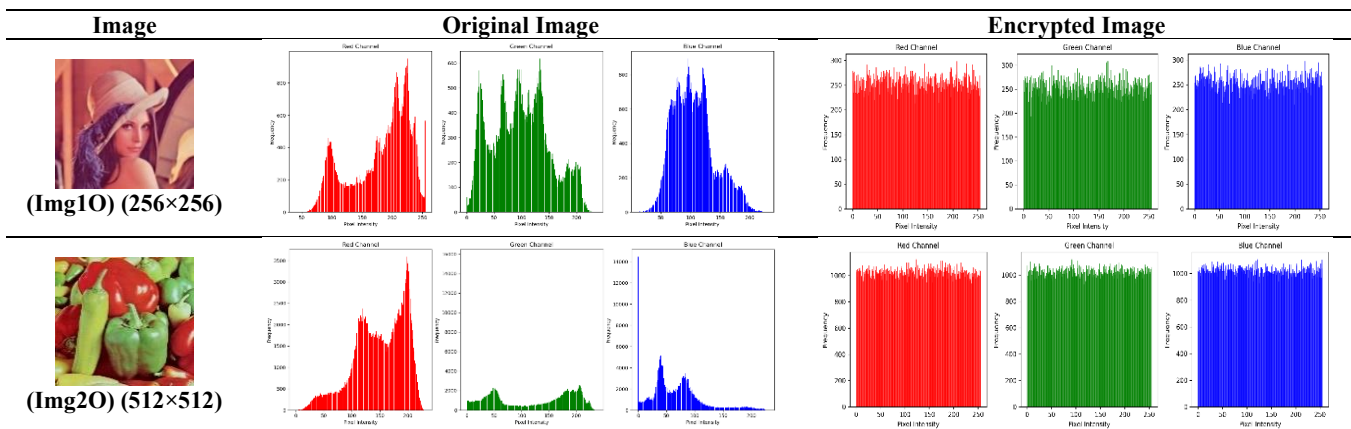


Figure 5. Key sensitivity analysis

Table 6. Visual analysis of histograms for original and encrypted images



endlf

3. RESULTS AND DISCUSSION

This section presents a robustness evaluation of our cryptographic scheme based on multiple metrics such as key space, histogram analysis, entropy, correlation coefficients, NPCR, UACI, AE, MSE, and NPSR. Before the decryption process can commence, the secret key must be securely delivered to the intended recipient via a secure channel.

3.1 Working environment

All the simulations mentioned in this study were carried out on a personal computer. Table 5 summarizes the hardware environment, software, and the source of the color images used in our experiments.

3.2 Key sensitivity analysis

The proposed system incorporates two chaotic maps frequently employed in cryptographic applications. Their inherent sensitivity to initial conditions and control parameters ensures that any minor change in the secret key produces a decrypted image distinct from the original, as shown in Figure 5.

Table 5. Simulation specifications

| Specifications | |
|----------------------|---|
| Processor | Intel® Core™ i7-6600U CPU @ 2.60 GHz (up to 2.80 GHz) |
| RAM | 16GB |
| Operating system | Windows 10 professional – 64 bits |
| Programming language | Python 3.12 |
| Image source | The USC-SIPI Image Database and NIH Clinical Center Releases Dataset of 32,000 CT Images [22, 23] |

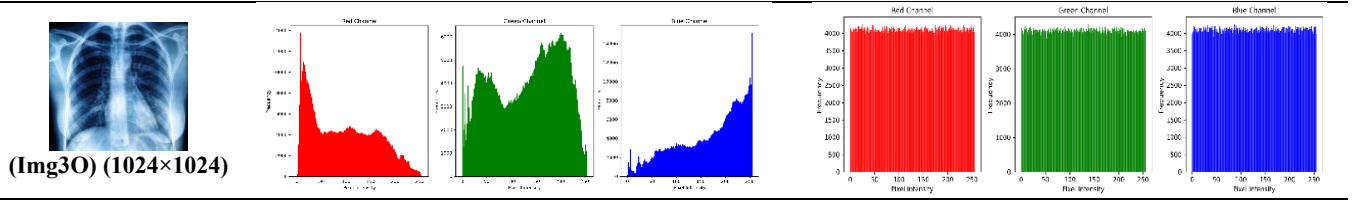








Table 7. Correlation coefficients of original and encrypted images

| Image | CH | CV | CD | Image | CH | CV | CD | Image | CH | CV | CD |
|--|---------|--------|--------|--|---------|---------|--------|--|--------|---------|---------|
|  Img1O | 0.9420 | 0.9471 | 0.8980 |  Img2O | 0.9617 | 0.977 | 0.9628 |  Img3O | 0.9979 | 0.9985 | 0.9985 |
|  Img1C | -0.0018 | 0.0007 | 0.0024 |  Img2C | -0.0015 | -0.0006 | 0.0005 |  Img3C | 0.000 | -0.0009 | -0.0006 |

3.3 Statistical attack analysis

The study of simulations of a cryptosystem is a crucial step in proving its robustness. Therefore, several analyses will be performed on encrypted and clear images randomly selected from a database. To address this issue, we focus on the study of histogram attacks, correlation attacks, and entropy attacks [24-26].

3.3.1 Histogram analysis

To be protected from any statistical attack, each image encrypted by our cryptosystem must exhibit a uniformly distributed histogram. Table 6 illustrates the histograms of the encrypted images and the reference original images.

It is observed that the histograms of the encrypted images are uniformly distributed. Consequently, our algorithm does not reveal any information about the pixel distribution in the original images. This ensures protection against any statistical attack.

3.3.2 Correlation analysis

A good cryptosystem must eliminate any correlation between neighboring pixels and reduce the high redundancy of the original image. Therefore, any correlation close to zero eliminates linear correlation between adjacent pixels. Let a and b be two image vectors of the same size N . The correlation between a and b is given by Eq. (8).

$$\text{Corr}_{ab} = \frac{\text{Cov}(a, b)}{\sqrt{V(a) \times V(b)}}$$

$$\text{where, } \text{Cov}(a, b) = \frac{1}{N} \sum_{i=1}^{i=N} (a_i - E(a))(b_i - E(b)) \quad (8)$$

$$E(a) = \frac{1}{N} \sum_{i=1}^{i=N} a_i \quad V(a) = \frac{1}{N} \sum_{i=1}^{i=N} (a_i - E(a))^2$$

where, $E(a)$, $V(a)$, $E(b)$, and $V(b)$ respectively represent the expected value and variance of vectors a and b .

Table 7 shows the correlation of two adjacent pixels vertically (CV), horizontally (CH), and diagonally (CD) in the original and encrypted images. It is observed that the

correlation of pixels in the original images is very close to 1 in all directions, whereas these coefficients are closer to zero in the case of encrypted images. This means that attackers cannot obtain any information from the encrypted image using this method.

3.3.3 Entropy analysis

According to Shannon's information theory, the entropy of each image channel (R, G, or B) is calculated using Eq. (9). In our information source, the number of possible states is $256=2^8$. Therefore, the theoretical entropy is close to the maximum value of 8. Table 8 provides the entropy values of images encrypted by our new technique [24, 27].

$$E = - \sum_{i=0}^{255} \text{pr}((x_i)) \log_2(\text{pr}(x_i)) \quad (9)$$

with $\text{pr}(x_i)$: Probability of occurrence of pixel value x_i (where $x_i \in [0, 255]$).

Table 8. Entropy values of the encrypted images

| Image | Img1C | Img2C | Img3C |
|---------|--------|--------|--------|
| Entropy | 7.9973 | 7.9992 | 7.9998 |

The entropy of any image encrypted by our approach is close to 8. Consequently, the proposed approach is immune to any entropy attack.

3.4 Differential attack analysis

Table 9. The values of NPCR, UACI, and AE for the encrypted images

| Image | Img1C | Img2C | Img3C |
|-------|-------|-------|-------|
| NPCR | 100.0 | 100.0 | 100.0 |
| UACI | 33.61 | 33.60 | 33.59 |
| AE | 50.26 | 50.22 | 50.20 |

An adversary may introduce a slight modification to the original image in order to evaluate the impact of such a perturbation on the corresponding encrypted image. A robust encryption system must exhibit high sensitivity to these changes. In modern cryptography, the Number of Pixels

Change Rate (NPCR), the Unified Average Changing Intensity (UACI), and the Avalanche Effect (AE) are key metrics used to assess resistance against differential attacks [18, 27, 28]. The NPCR is determined by Eq. (10). The related results are depicted in Table 9.

$$NPCR = \left(\frac{1}{3nm} \sum_{i,j=1}^{3nm} \delta(i,j) \right) * 100\% \quad (10)$$

where,

$$\delta(i,j) = \begin{cases} 1 & \text{if } CO_1(i,j) \neq CM_2(i,j) \\ 0 & \text{if } CO_1(i,j) = CM_2(i,j) \end{cases}$$

CO₁: Is the encrypted image of the original image by our method

CM₂: Is the encrypted image of the slightly modified original image by our method

The UACI is determined by Eq. (11):

$$UACI = \left(\frac{1}{3nm} \sum_{i,j=1}^{3nm} \frac{|CO_1(i,j) - CM_2(i,j)|}{255} \right) * 100\% \quad (11)$$

The AE is determined by Eq. (12):

$$AE = \left(\frac{\text{Number of modified bits}}{\text{Total number of bits}} \right) * 100\% \quad (12)$$

The computed values of AE, UACI, and NPCR surpass the commonly accepted thresholds—50% for AE, 33.40% for UACI, and 99.60% for NPCR. These results highlight the strong sensitivity of our method to even minimal changes in the input image: A difference of just one bit between two images leads to entirely distinct decrypted outputs. This confirms that the proposed technique is effective in defending against differential attacks.

3.5 Analysis of encryption time and algorithmic complexity

The encryption time of a cryptosystem is a critical parameter that can impact its operational efficiency. The encryption time measured for images of varying sizes, obtained through the proposed approach, is presented in Table 10. These values were calculated using the hardware and software configuration outlined in Table 1. We note that the encryption time depends on the size of the original image. The complexity or cost of an algorithm corresponds to the number of fundamental (basic) operations it performs on an image of size (n×m). Therefore, our new technique has a complexity equivalent to O(nm).

Table 10. Encryption time in seconds

| Image | From Img1O to img1C (256×256) | From Img2O to img2C (512×512) | From Img3O to img3C (1024×1024) |
|-----------------|-------------------------------|-------------------------------|---------------------------------|
| Encryption time | 0.104 | 0.141 | 0.186 |

Table 11. PSNR and MSE values

| | Image | Img1 (256×256) | Img2(512×512) | Img3(1024×1024) |
|------|-----------------------|----------------|---------------|-----------------|
| PSNR | Original to Encrypted | 3.8274 dB | 3.3082 dB | 2.9462 dB |
| | Decrypted to original | ∞ | ∞ | ∞ |
| MSE | Original to Encrypted | 26936 | 30357 | 32995 |
| | Decrypted to original | 0 | 0 | 0 |

Table 12. Comparison of our method with others

| Parameters | Correlation H | | Entropy | | NPCR | | UACI | | PSNR | |
|------------|---------------|---------|---------|--------|----------|----------|----------|---------|--------|--------|
| | Img1C | Img2C | Img1C | Img2C | Img1C | Img2C | Img1C | Img2C | Img1C | Img2C |
| proposed | -0.0018 | -0.0015 | 7.9973 | 7.9992 | 100% | 100.0% | 33.61% | 33.60% | 3.8274 | 3.3082 |
| Ref. [9] | -0.00273 | - | - | - | 99.68% | 99.67% | 33.49% | 33.48% | 7.0312 | - |
| Ref. [13] | 0.0004 | -0.0048 | 7.9993 | 7.9973 | 99.6098% | 99.6108% | 33.4536% | 33.5173 | - | - |
| Ref. [30] | 0.00311 | - | 7.7043 | - | 100% | - | 50.2011% | - | 7.7268 | - |

3.6 Cryptographic robustness via MSE and PSNR

To evaluate cryptographic robustness, a rigorous comparison is conducted between the decrypted image and the original, including:

-Qualitative analysis (visual inspection): The two images exhibit a perfect visual identity, with no observable differences to the naked eye.

-Quantitative similarity metrics (MSE, PSNR) validating global fidelity.

The Mean Square Error (MSE) measures the difference between two images of size 1*(3*N*M) [29]. It is defined by Eq. (13). The Peak Signal-to-Noise Ratio (PSNR) quantifies the ratio of the maximum signal power to the noise power [29].

It is defined by Eq. (14).

$$MSE = \frac{1}{3nm} \sum_{i=1}^{3nm} (IO(i) - IED(i))^2 \quad (13)$$

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right) \quad (14)$$

With:

IO(i): Original vector (unencrypted).

IED(i): Encrypted/decrypted vector.

Table 11 presents the PSNR (dB) and MSE values calculated for the reference images tested by our algorithm.

The values obtained by our algorithm meet the standards.

3.7 Comparison

In this section, we compare the horizontal correlation, entropy, NPCR, UACI, and PSNR values calculated for the Img1C and Img2C images using our algorithm, as illustrated in Table 12. The obtained results exceed established standards, confirming that our encryption method outperforms those proposed in references [9, 13, 30]. This evidence further confirms that our system is resistant to all known attacks, particularly differential ones.

4. CONCLUSION

The three substitution tables constructed from the most widely used chaotic maps in cryptography, along with the implementation of new enhanced substitution functions, have yielded highly satisfactory results. Similarly, applying the new encryption mode to introduce diffusion processes has provided strong protection against differential attacks in the new system. Comparisons made between our system and other similar algorithms may encourage further improvements. Concurrently, the encryption time encourages researchers to implement our system in new algorithms for encrypting large data and video sequences.

Instead of working with the conventional ring $\mathbb{Z}/256\mathbb{Z}$ imposed by classical encryption systems, we choose to construct a field with 256 elements, on which a new system for encrypting color images will be developed, leveraging the specific properties of this structure.

REFERENCES

- [1] Jamal, S.S., Hazzazi, M.M., Khan, M.F., Bassfar, Z., Aljaedi, A., ul Islam, Z. (2024). Region of interest-based medical image encryption technique based on chaotic S-boxes. *Expert Systems with Applications*, 238: 122030. <https://doi.org/10.1016/j.eswa.2023.122030>
- [2] Odeh, A., Al-Haija, Q.A. (2023). Medical image encryption techniques: A technical survey and potential challenges. *International Journal of Electrical and Computer Engineering (IJECE)*, 13(3): 3170-3177. <https://doi.org/10.11591/ijece.v13i3.pp3170-3177>
- [3] Mahajan, V.T., Sridaran, R. (2023). Taxonomy of image encryption techniques-A survey. In *International Conference on Advancements in Smart Computing and Information Security*, Rajkot, India, pp. 274-290. https://doi.org/10.1007/978-3-031-59100-6_20
- [4] Hraoui, S., Gmira, F., Abbou, M.F., Oulidi, A.J., Jarjar, A. (2019). A new cryptosystem of color image using a dynamic-chaos hill cipher algorithm. *Procedia Computer Science*, 148: 399-408. <https://doi.org/10.1016/j.procs.2019.01.048>
- [5] Wen, H., Lin, Y., Yang, L., Chen, R. (2024). Cryptanalysis of an image encryption scheme using variant Hill cipher and chaos. *Expert Systems with Applications*, 250: 123748. <https://doi.org/10.1016/J.ESWA.2024.123748>
- [6] Kumari, M., Gupta, S., Sardana, P. (2017). A survey of image encryption algorithms. *3D Research*, 8: 1-35. <https://doi.org/10.1007/SB13319-017-0148-5>
- [7] Wang, X., Yang, J. (2020). A novel image encryption scheme of dynamic S-boxes and random blocks based on spatiotemporal chaotic system. *Optik*, 217: 164884. <https://doi.org/10.1016/j.ijleo.2020.164884>
- [8] Qobbi, Y., Abid, A., Jarjar, M., El Kaddouhi, S., Jarjar, A., Benazzi, A. (2023). Adaptation of a genetic operator and a dynamic S-box for chaotic encryption of medical and color images. *Scientific African*, 19: e01551. <https://doi.org/10.1016/j.sciaf.2023.e01551>
- [9] El Bourakkadi, H., Chemlal, A., Tabti, H., Kattass, M., Jarjar, A., Benazzi, A. (2024). Improved Vigenere approach incorporating pseudorandom affine functions for encrypting color images. *International Journal of Electrical and Computer Engineering (IJECE)*, 14(3): 2684. <https://doi.org/10.11591/ijece.v14i3.pp2684-2694>
- [10] Kattass, M., Rrghout, H., Jarjar, M., Jarjar, A., Gmira, F., Benazzi, A. (2023). Chaotic image encryption using an improved vigenère cipher and a crossover operator. In *International Conference on Computing, Intelligence and Data Analytics*, pp. 181-191. https://doi.org/10.1007/978-3-031-53717-2_17
- [11] Ali, T.S., Ali, R. (2020). A new chaos based color image encryption algorithm using permutation substitution and Boolean operation. *Multimedia Tools and Applications*, 79(27): 19853-19873. <https://doi.org/10.1007/SB11042-020-08850-5>
- [12] Nkandeu, Y.P.K., Tiedeu, A. (2019). An image encryption algorithm based on substitution technique and chaos mixing. *Multimedia Tools and Applications*, 78(8): 10013-10034. <https://doi.org/10.1007/SB11042-018-6612-2>
- [13] Wang, X., Zhang, M. (2021). An image encryption algorithm based on new chaos and diffusion values of a truth table. *Information Sciences*, 579: 128-149. <https://doi.org/10.1016/j.ins.2021.07.096>
- [14] Hosny, K.M., Elnabawy, Y.M., Salama, R.A., Elshewey, A.M. (2024). Multiple image encryption algorithm using channel randomization and multiple chaotic maps. *Scientific Reports*, 14(1): 30597. <https://doi.org/10.1038/s41598-024-79282-6>
- [15] Al-Saadi, H.M., Alshawhi, I.S. (2023). Efficient and secure hybrid chaotic key generation for light encryption device block cipher. *Indonesian Journal of Electrical Engineering and Computer Science*, 31(2): 1032-1040. <https://doi.org/10.11591/ijeecs.v31.i2.pp1032-1040>
- [16] Andono, P.N. (2022). Improved pixel and bit confusion-diffusion based on mixed chaos and hash operation for image encryption. *IEEE Access*, 10: 115143-115156. <https://doi.org/10.1109/ACCESS.2022.3218886>
- [17] Singh, A.K., Chatterjee, K., Singh, A. (2022). An image security model based on chaos and DNA cryptography for IIoT images. *IEEE Transactions on Industrial Informatics*, 19(2): 1957-1964. <https://doi.org/10.1109/TII.2022.3176054>
- [18] Durdu, A. (2024). Image transfer with secure communications application using a new reversible chaotic image encryption. *Multimedia Tools and Applications*, 83(2): 3397-3424. <https://doi.org/10.1007/s11042-023-15707-0>
- [19] Wang, X., Guan, N., Liu, P. (2022). A selective image encryption algorithm based on a chaotic model using modular sine arithmetic. *Optik*, 258: 168955. <https://doi.org/10.1016/j.ijleo.2022.168955>
- [20] Al-Ofeishat, H.A., Alkasassbeh, J.S., Alzyoud, K.Y., Al-

- Taweel, F.M., Alrawashdeh, H., Al-Rawashdeh, A.Y. (2024). A novel approach to simplified and secure message cryptography using chaotic logistic maps and index keys. *International Journal of Electrical & Computer Engineering*, 14(5): 5139-5152. <https://doi.org/10.11591/ijece.v14i5.pp5139-5152>
- [21] Chen, Y., Tang, C., Yi, Z. (2020). A novel image encryption scheme based on PWLCM and standard map. *Complexity*, 2020(1): 3026972. <https://doi.org/10.1155/2020/3026972>
- [22] The USC-SIPI Image Database. <https://sipi.usc.edu/database/>.
- [23] NIH Clinical Center Releases Dataset of 32,000 CT Images. <https://www.nih.gov/news-events/news-releases/nih-clinical-center-releases-dataset-32000-ct-images>.
- [24] Silva-García, V.M., Flores-Carapia, R., Cardona-López, M.A. (2024). A hybrid cryptosystem incorporating a new algorithm for improved entropy. *Entropy*, 26(2): 154. <https://doi.org/10.3390/e26020154>
- [25] Kumar, M., Aggarwal, J., Rani, A., Stephan, T., Shankar, A., Mirjalili, S. (2022). Secure video communication using firefly optimization and visual cryptography. *Artificial Intelligence Review*, 55: 2997-3017. <https://doi.org/10.1007/s10462-021-10070-8>
- [26] Zhang, Y. (2021). Statistical test criteria for sensitivity indexes of image cryptosystems. *Information Sciences*, 550: 313-328. <https://doi.org/10.1016/j.ins.2020.10.026>
- [27] Daoui, A., Yamni, M., Chelloug, S.A., Wani, M.A., El-Latif, A.A.A. (2023). Efficient image encryption scheme using novel 1D multiparametric dynamical tent map and parallel computing. *Mathematics*, 11(7): 1589. <https://doi.org/10.3390/math11071589>.
- [28] Kumar, Y., Guleria, V. (2024). Mixed-multiple image encryption algorithm using RSA cryptosystem with fractional discrete cosine transform and 2D-Arnold Transform. *Multimedia Tools and Applications*, 83(13): 38055-38081. <https://doi.org/10.1007/s11042-023-16953-y>
- [29] Daoui, A., Karmouni, H., Sayyouri, M., Qjidaa, H., Maaroufi, M., Alami, B. (2021). New robust method for image copyright protection using histogram features and sine cosine algorithm. *Expert Systems with Applications*, 177: 114978. <https://doi.org/10.1016/j.eswa.2021.114978>
- [30] Rashid, A.A., Hussein, K.A. (2023). Image encryption algorithm based on the density and 6D logistic map. *International Journal of Electrical & Computer Engineering*, 13(2): 1903-1913. <https://doi.org/10.11591/ijece.v13i2.pp1903-1913>