ISSN: 2690- 912X

Research Article

Journal of Genetic Engineering and Biotechnology Research

Medical Images Encryption Using Two Improved Vigenère Laps Separated by Genetic Crossover

Mariem Jarjar, Abid Abdellah, Hicham Rrghout, Mourad Kattass, Abdellatif JarJar* and Abdellhamid Benazzi

Mohamed First University, MATSI Laboratory, Oujda, Morocco

*Corresponding Author
Abdellatif JarJar, Mohamed First University, MATSI Laboratory, Oujda, Morocco

Submitted: 2023, Oct 10; **Accepted**: 2023, Nov 14; **Published:** 2024, Jan 31

Citation: Jarjar, M., Abdellah, A., Reghout, H., Kattass, M., Jarjar, A., Benazzi, A. (2024). Medical Images Encryption Using Two Improved Vigenère Laps Separated by Genetic Crossover. *J Gene Engg Bio Res, 6*(1), 01-25.

Abstract

In this article, we present a novel method for encrypting medical images, based on a significant improvement of the classical Vigenère technique. This method utilizes dynamically generated large-sized substitution tables derived from multiple pseudo-random vectors. After vectorizing the original image and applying confusion with a pseudo-random vector, the first Vigenère round encryption is performed using two substitution tables, followed by a genetic crossover adapted for image encryption and controlled by a pseudo-random table. Next, a second Vigenère round encryption is applied, implementing new confusion and diffusion functions to maximize the avalanche effect and protect our system against all differential attacks. The output vector is then fully subjected to permutation to increase attack complexity. Simulations are conducted on a large number of images of various sizes and formats to ensure that our approach is not vulnerable to known attacks.

Notations	1.	$G_t = {}^{\mathbb{Z}}/_{t\mathbb{Z}} - \text{ring}$
	2.	$\oplus$: XOR operator.
	3.	$E(x)$: The integer fraction of the real x

Keywords: Substitution Table, Chaotic Map, Genetic Crossover, Image Encryption.

1. Introduction

The need for data transmission in insecure networks through the internet and social networks has led to the invention of various encryption techniques. With the advancement in information transmission technology, different forms of multimedia data, such as images, are transmitted over the internet. However, due to the intrinsic properties of images, such as strong correlation between neighboring pixels and the large size of the data, conventional techniques exhibit weaknesses and disadvantages.

The rapid advancement in chaos theory mathematics, along with the unique properties of chaotic systems such as randomness and sensitivity to initial conditions, provides researchers with the opportunity to further enhance certain classical encryption systems. Given the significant interest in security, numerous image encryption techniques have emerged in the digital world, attempting to improve some classical techniques such as Hill Caesar, Vigenère and Feistel [1-9].

In recent years, substitution tables have been recognized as one of the most widely used tools in symmetric encryption systems. These S-boxes enable the confusion process [10]. Several research works based on the utilization of substitution tables have been proposed. In this regard, the authors of reference introduced an image encryption system based on the utilization of a new chaotic system composed of two maps: Sine and Tent [11]. A wide chaotic range characterizes this new chaotic system (STS). This property allows for the generation of improved chaotic sequences used in generating two substitution tables, S1 and S2. The utilization of two S-boxes enhances the resistance of the encryption algorithm against all four types of attacks.

The authors of have developed an image encryption system for square images of size N×N, based on the utilization of two hyper-chaotic systems, the logistic map and the tent map [12]. Pseudo-random number sequences are used for constructing N S-boxes. Each S-box is employed for encrypting a row of the plaintext image.

In reference the authors proposed a digital image encryption algorithm based on two permutation-substitution phases. In the first phase, a pixel position rearrangement of the plaintext image is performed to break the correlation between adjacent pixels [13]. Then, the logistic-sine map is used to generate a robust S-box for establishing a substitution phase of pixel values in the scrambled image. This confusion technique ensures non-linearity in the components of the encrypted image.

The authors of the article proposed a new encryption algorithm based on the combination of the logistic map and the tent map for the development of a new chaotic system denoted as (STL) [14]. The properties of the STL system and linear algebra are exploited for constructing strong S-boxes against all types of attacks. In most image encryption systems, the encryption process is typically different from the decryption process.

In the article the authors proposed a cryptosystem in which the encryption and decryption algorithms are identical. This type of cryptosystem is referred to as a unified image encryption algorithm. In this article, a three-dimensional S-box is proposed [15].

The authors of the article proposed an image encryption scheme based on the development of three large-sized S-boxes: SR, SG, and SB. These tables are used to establish the two encryption processes, confusion and diffusion [16].

1) Problematic

In the conventional Vigenère system, the identification of the private key size makes the algorithm susceptible to statistical attacks, as revealed and elaborated by Babbage. Moreover, the knowledge of the substitution matrix renders the conventional system susceptible to brute-force attacks. Furthermore, in the absence of diffusion operations and chaining functions, conventional systems re remain vulnerable to differential attacks.

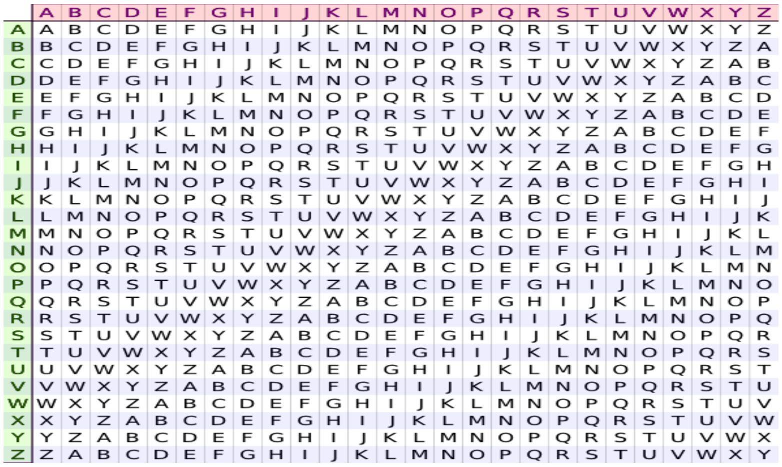
2) Our Contribution

Our contribution aims to enhance the functionality and structure of the conventional Vigenère encryption through the utilization of dynamic matrices as a replacement for the traditional substitution matrix. This involves constructing two improved encryption functions and generating two substitution matrices using the most widely adopted chaotic maps in the field of cryptography. Additionally, we employ the principle of double encryption, applying it to all pixels of the original image. Furthermore, a chaotic diffusion function is employed in the second round to amplify the avalanche effect and fortify the system against differential attacks. We also incorporate a genetic crossover operation between the two rounds of improved Vigenère rounds. Notably, our approach deviates from the classical method by employing different S-boxes for the decryption process.

2. Preliminary

1) The Classic Vigenère Method

The technique was based on a static matrix (V) [26] presented by [figure 1]:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 1: Classical Vigenère table

The encryption and the decryption process are defined in equation (1).

$$\begin{cases} C_i = V(P_i, K_i) = (P_i + K_i) \mod 26 \\ P_i = V(C_i, K_i) = (P_i - K_i) \mod 26 \end{cases} \text{For each i in } [1\dots \text{lm}]$$

With (P):plaintext, (C):ciphertext; (K):encryption key, (V)Vigenère matrix and (lm):clear text size.

2. Genetic Crossover

a. Definition:

In a genetic algorithm, crossover is a genetic operation used to modify the genes of the descendants. This operation is based on a coupling of two individuals of the parent pool, chosen in a pseudo random way in order to generate strong individuals. The method chosen depends on the coding method.

b. Different Types of Genetic Crossover:

The most well-known crossover operators are:

- **Single Crossover:** A specific crossover point is chosen along the parent organism's chain, at which point the data beyond that point in the organism's chain is swapped between the two parent organisms [figure 2].

Example:

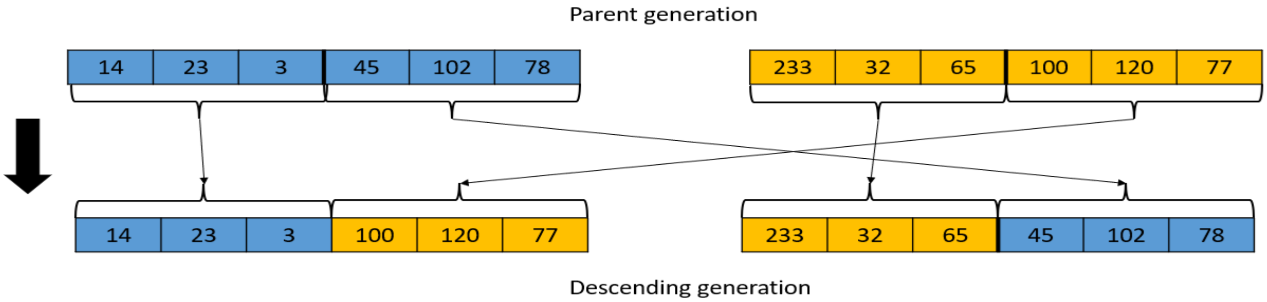


Figure 2: Single crossover

- **Double crossover:** This represents a particular instance of the K-point crossover technique, where two random points are selected on the individual chromosomes (chains), facilitating an exchange of genetic material between these designated points [figure 3].

Example:

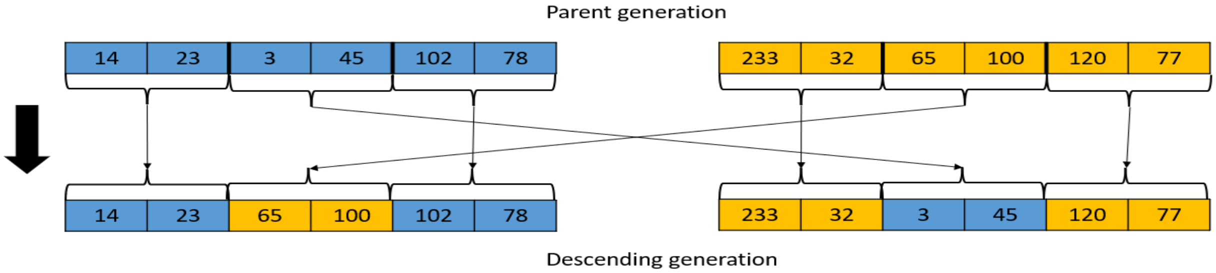


Figure 3: Double crossover

- **Uniform Crossover:** Each gene (bit) is randomly chosen from the corresponding genes of the parent chromosomes [figure 4].

Example:

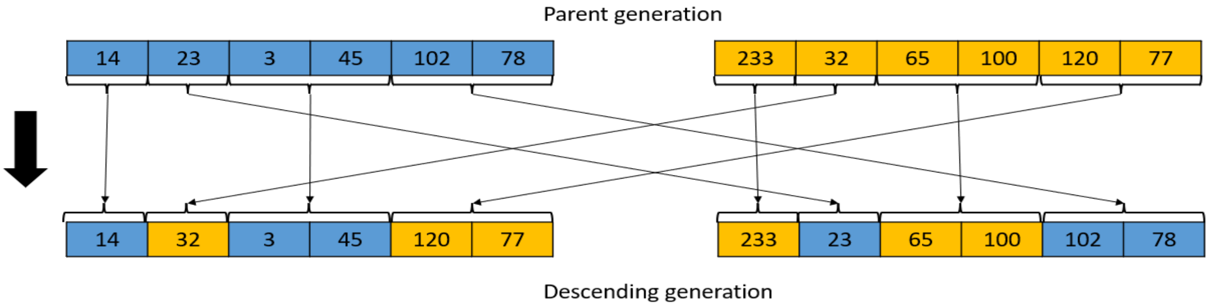


Figure 4: Uniform crossover

3. The Proposed Method:

The proposed method consists of two stages. In each stage, confusion and diffusion operations based on the use of two S-boxes are applied. Additionally, a genetic crossover is performed between the two stages. The following steps describe our algorithm:

- Construction of chaotic sequences.
- Generation of two substitution matrices.
- Defining a new cipher function.
- Transformation of the original image into a single vector.
- Combination of the original image with a chaotic vector.
- Application of the first round of Vigenère.
- Application of genetic crossover.
- Application of the second Vigenère round.
- Applying a global permutation.
- Encryption process simulation result.

a. Generation of Chaotic Sequences

The encryption parameters necessary for the optimal performance of our system are obtained from the two most extensively utilized chaotic maps in cryptography. This selection is motivated by their ease of implementation and high sensitivity to initial parameters.

a) The Logistic Map

The logistic map is a recurrent sequence described by a simple quadratic polynomial defined by equation (2) [17,18].

$$\begin{cases} u_0 \in]0,5 \quad 1[\quad , \quad \mu \in [3,75 \quad 4] \\ u_{n+1} = \mu u_n(1-u_n) \end{cases} \qquad (2)$$

The Lyapunov exponent λ of the logistic map is $\lambda = \log(2) > 0$, which proves that this sequence is highly sensitive to initial conditions.

b) HENON Map

Henon's two-dimensional chaotic map was first discovered in 1978. It is described by equation (3) [19,20].

$$\begin{cases} v_0 \text{ et } w_0 \quad a = 0.3 \text{ et } b \in [1.07\,, 1.4] \\ v_{n+1} = 1 + w_n - a v_n^2 \\ w_{n+1} = b v_n \end{cases} \qquad (3)$$

We can convert the two-dimensional map expression into a one-dimensional map which is easy to implement in the encryption system. This formula is described by equation (4).

$$\begin{cases} v_0 \text{ and } v_1 \text{ in } [0\,,1] \text{ and } a = 0.3 \text{ et } b \in [1.07, 1.4] \\ v_{n+2} = 1 - a v_{n+1}^2 + b v_n \end{cases} \qquad (4)$$

This chaotic map is also sensitive to initial conditions and possesses a very large key size, which makes it highly secure against brute-force attacks.

c). Chaotic Vector Design

Our work requires the construction of three chaotic vectors (VC1),(VC2) and (VC3), with coefficients in (G_{256}), and two binary vectors (VB) and (BV), which will serve as control vectors. This construction is presented in algorithm 1.

Algorithm 1: chaotic vectors

```
Input: two chaotic sequences u, v
Output: (VC1), (VC2), (VC3) chaotic vectors
Begin
    For i =1 to 3nm
        $\mathrm{VC1(i)} = \mathrm{mod}\left(\mathrm{E}\left(\frac{\mathrm{inf}(\mathrm{u(i)},\mathrm{v(i)})+\mathrm{u(i)}}{2} * 10^{11}, 254\right) + 1\right)$
        $\mathrm{VC2(i)} = \mathrm{mod}\left(\mathrm{E}\left(\frac{2*\mathrm{u(i)}+3*\mathrm{v(i)}}{5} * 10^{11}, 253\right) + 2\right)$
        $\mathrm{VC3(i)} = \mathrm{E}\left(\frac{3*\mathrm{VC1(i)}+2*\mathrm{VC2(i)}}{5}\right)$
    End For
end
```

The design of (VB) and (BV) vectors is illustrated by Algorithm 2.

Algorithm 2: design of binary control vectors (VB) and (BV)

```
Input: (VC1), (VC2) chaotic vectors
Output: (VB) binary vector
Begin
    For i =1 to 3nm
        Yew $\mathrm{VC1(i)} \geq \mathrm{VC2(i)}$ then
            $\mathrm{VB(i)} = 0$
        Else
            $\mathrm{VB(i)} = 1$
        End If
    End For
end
```

```
Input: u, v chaotic sequences
Output: (BV) binary vector
Begin
    For i =1 to 3nm
        If $\mathrm{u(i)} \geq \mathrm{v(i)}$ then
            $\mathrm{BV(i)} = 0$
        Else
            $\mathrm{BV(i)} = 1$
        End If
    End For
end
```

d) Global Permutation Design

This permutation is generated by Algorithm 3:

Algorithm 3: global permutation design

```
Input: binary control vector (VB)
Output: Global permutation (GP)
Begin
    k=1
```

```
        For i =1 to 3nm
            If VB(i) = 0 then
            k=k+1
        end If
    end For
    For i =3nm to 1
        If VB(i) = 1 then
            k=k+1
        end If
    end For
end
```

Example:
[Table 1] illustrate an example of permutation generated using algorithm 3

VB	1	0	0	1	1	0	1	0	0	1	0	1	0	0	1	1	0	1	0	1
count	10	9	8	11	12	7	13	6	5	14	4	15	3	2	16	17	1	18	0	19

$$GP = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 \\ 10 & 9 & 8 & 11 & 12 & 7 & 13 & 6 & 5 & 14 & 4 & 15 & 3 & 2 & 16 & 17 & 1 & 18 & 19 & 0 \end{pmatrix}$$

Table 1: Example of permutation in (G_{20}).

b) Encryption Process

a) First Confusion of the Original Image

Initially, the original image of size (n, m, 3) is transformed into a vector (img) of size (3nm). Subsequently, a confusion operation is applied between the vector (img) and the chaotic vector (VC1). This procedure is detailed in [Figure 5] and Algorithm 4.

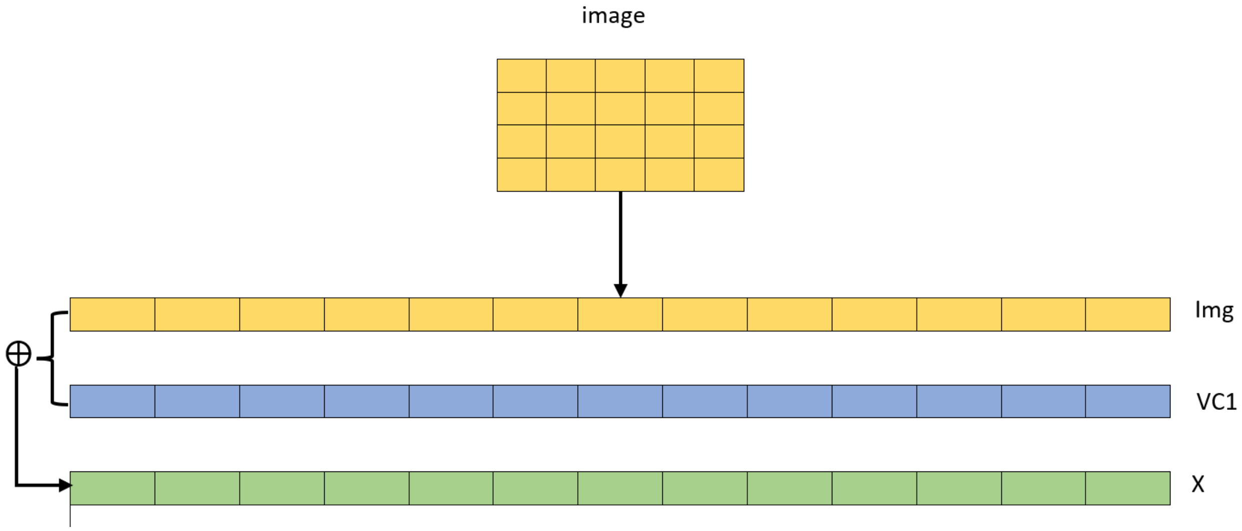


Figure 5: First confusion of the original image

Algorithm 4: First confusion of the original image

```
Input: original image (img) and chaotic vector (VC1)
Output: (X)
Begin
    For i =1 to 3nm
        $X(i) = img(i) \oplus VC1(i)$
    End For
end
```

c) First Round of Improved Vigenère

By taking inspiration from the classical Vigenère technique, we will generate two chaotic substitution matrices, (SM1) and (SM2), each with a size of (256,256), using the following procedure:

- (P1): Permutation obtained by a large ascending sort of the first 256 values of the sequence.(u).
- (P2): Permutation obtained by strict descending sorting of the first 256 values of the sequence.(v).

This new construction is entirely governed by the vector (BV). It is given by algorithm 5.

Algorithm 5: S-box design

```
Input: two permutations (P1) and (P2)
Output: two S-boxes (SM1) and (SM2)
Begin
    For i =1 to 256
        $SM1(1, i) = P1(i)$
        $SM2(1, i) = P2(i)$
    End For
    For i =2 to 256
        For j =1 to 256
            If BV(i)=1 then
                $SM1(i,j) = SM1(i-1, mod(j + VC1(i), 256))$
                $SM2(i,j) = SM2(i-1, mod(j + VC2(i), 256))$
            Else
                $SM1(i,j) = SM1(i-1, mod(j + VC2(i), 256))$
                $SM2(i,j) = SM2(i-1, mod(j + VC1(i), 256))$
            End If
        End For
    End For
end
```

This is a chaotic displacement applied at the level of line i-1 to generate the elements of line i. It is noteworthy that the construction of the two matrices is entirely controlled by the vector (BV). [Figure 6] illustrates an example of a substitution matrix with values in (G_8).

(SM1)	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**		B.V.	VC1	VC2		**(SM2)**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	3	5	0	6	2	7	1	4						**0**	0	4	5	7	1	2	6	3
1	2	7	1	4	3	5	0	6		1	5	4		**1**	6	2	1	7	3	0	4	5
2	4	3	5	0	6	2	7	1		1	3	5		**2**	7	1	2	4	5	6	3	0
3	2	7	1	4	3	5	0	6		0	3	4		**3**	5	2	3	0	6	7	1	4
4	0	6	2	7	1	4	3	5		1	4	2		**4**	0	4	1	3	2	6	7	5
5	7	5	2	4	6	1	0	3		0	4	5		**5**	6	3	4	7	0	5	1	2
6	1	3	2	6	0	4	7	5		0	5	2		**6**	4	7	5	1	6	0	3	2
7	2	0	4	7	6	1	3	5		1	2	3		**7**	5	1	7	2	3	4	6	0

Figure 6: substitution matrix with values in (G_8)

i) Expression of the classical Vigenère function

The matrices (SM1) and (SM2) will be used in the encryption process through a substitution function. Equation 5 illustrates the substitution function of the classical Vigenère method:

$$\mathrm{Y(i)} \;= \mathrm{V}\big(\mathrm{K}, \mathrm{X(i)}\big) \qquad (5)$$

With (K) is the encryption key.

ii) New mathematical expression for the substitution function.

Equation 6 illustrates the new effective expression of the image Y(i) as a function of the pixel X(i) using the advanced Vigenère method.

$$\mathrm{AV_1}\big(\mathrm{X(i)}\big) = \mathrm{Y(i)} = \begin{cases} \text{if VB(i)} = 0 \text{ then} \\ \mathrm{Y(i)} \;= \mathrm{SM1}\big(\mathrm{VC1(i)}, \mathrm{SM2}\big(\mathrm{VC2(i)}, \mathrm{X(i)} \oplus \mathrm{VC3(i)}\big)\big) \\ \text{else} \\ \mathrm{Y(i)} \;= \mathrm{SM2}\,\big(\mathrm{VC2(i)}, \mathrm{SM1}\big(\mathrm{VC3(i)}, \mathrm{X(i)} \oplus \mathrm{VC1(i)}\big)\big) \\ \text{endif} \end{cases} \qquad (6)$$

Note that (AV_1) is the cipher function of the first round.

iii) The First Round of the Encryption Process

The first round of encryption process is defined by Algorithm 6.

```
Algorithm 6: first round of Vigenere
Input: (X) vector of size (3nm),
Output: (Y) vector of size (3nm)
Begin
    For i =1 to 3nm
        $\mathrm{Y(i)} = \mathrm{AV_1}\big(\mathrm{X(i)}\big)$
    End For
end
```

[Figure 7] shows the first round of the encryption process.

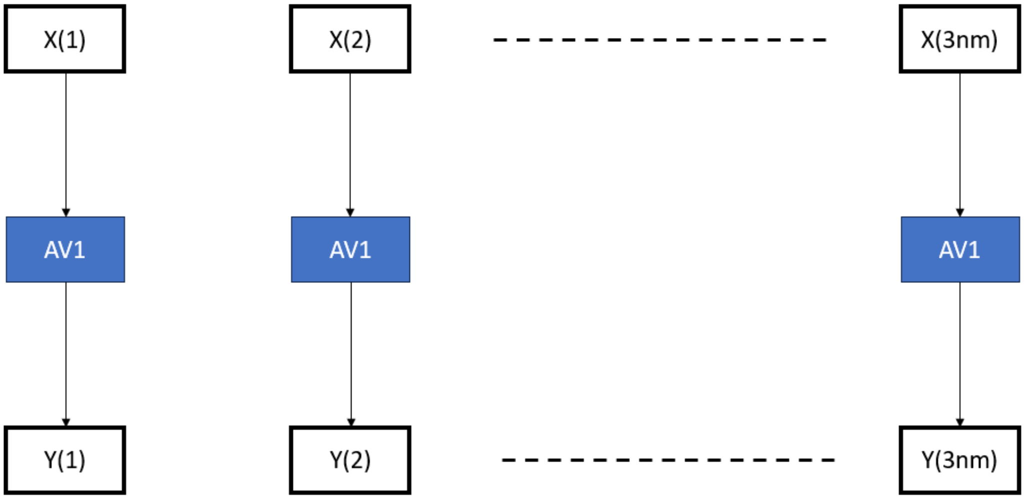


Figure 7: First round of enhanced Vigenère

After completing the first round, the resulting vector (Y) will be regarded as a clear image and subsequently employed for genetic crossover.

It is observed that this first round produces a slightly encrypted image [Table 2], but is safe from any frequency and statistical attacks. In order to increase the complexity of attacks, we add a second encryption round

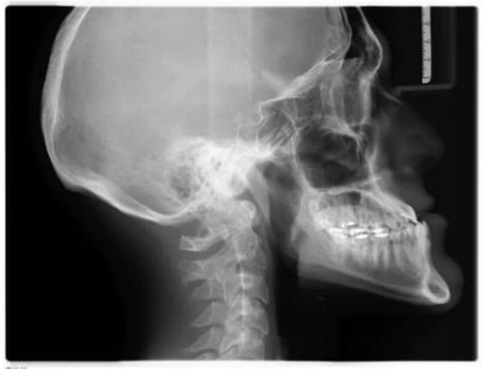
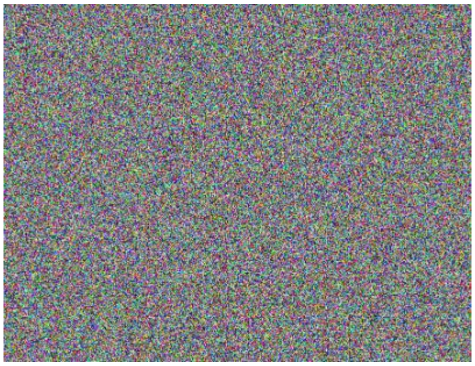
Image	Cipher	Parameters		
		Vertical correlation	R	0.00350
			G	-0.00056
			B	-0.00180
		Horizontal correlation	R	0.00019
			G	0.00194
			B	-0.00247
		Diagonal correlation	R	-0.00057
			G	0.00092
			B	0.00385
		Entropy	R	7.99870
			G	7.99873
			B	7.99876

Table 2: Vigenère first round results

d) Genetic Crossover Suitable For Image Encryption:

The classical genetic algorithm is considered a class of optimization algorithm; therefore, the original data (population) is lost by applying a crossover operation, which is not suitable for an encryption process [21-34].

The main idea of our technique is to apply the k-point crossover method by subdividing the vector (Y) and the chaotic vector (VC1) into n blocks of size 3m, and then creating a genetic crossover table (CT) of size (n, 3), defined as follows:

- The first column is a permutation obtained by sorting the first n values of the sequence (u) in ascending order, indicating the index of the block (Y).
- The second column is a permutation obtained by sorting the first n values of the sequence (v) in ascending order, indicating the index of the block (VC1).
- The third column is a permutation obtained by sorting the first n values of the sequence (VC3) in ascending order, indicating the index of the block (YC) obtained by the crossover of the first two blocks.

The crossover operation is illustrated by Algorithm 7.

```
Algorithm 7: genetic crossover
Input: (Y) vector of size (3nm), VC1 chaotic vector and (CT) crossover table
Output: (YC) vector of size (3nm)
Begin
  For i =1 to n
      For j=2 to 3m
  $YC((CT(i,3)-1)*3m+j) = VC1((CT(i,2)-1)*3m+j) \oplus Y((CT(i,1)-1)*3m+j)$
      End For
  End For
end
```

Example:

Let (Y) and (VC1) be two vectors of size 9, divided into 3 blocks of size 3 each, and let CT be a crossover table of size (3,3), this table is described by [Table 3].

Y block index	VC1 block index	YC block index
3	2	3
2	1	1
1	3	2

Table 3: Cross table of size (3,3)

Each row of (CT) represents the indices of the blocks interacting with each other in the genetic crossover operation. For instance, the first row of (CT) indicates that the crossover operation is performed between the 3rd block of (Y) and the 2nd block of (VC1), and the result is stored in the 3rd block of (YC). The rest of the process is described in [Figure 8].

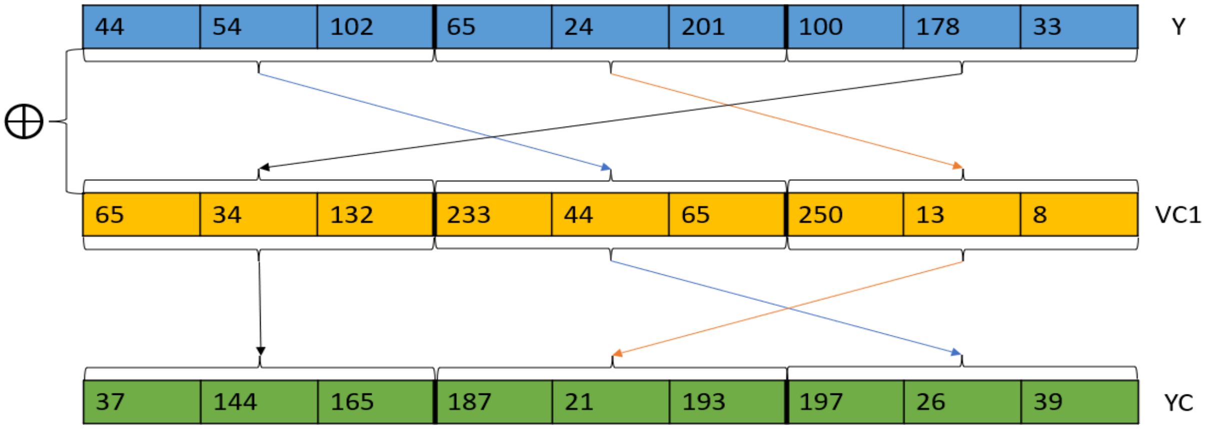


Figure 8: Genetic crossing operation

The resulting vector of the crossover operation will be the object of a second rounds.

e) Second round of the improved Vigenère method

After the genetic crossover, a second round of Vigenère encryption is performed, distinguished from the first round by the implementation of a new confusion function associated with a diffusion phase. This round begins with the calculation of an initialization value.

i) Calculation of the initialization value

At the end of the genetic crossover, an initialization value (IV) will be calculated, according to Algorithm 10, to initiate the encryption process.

Algorithm 8: calculates the second initial value

```
Input: (YC) vector of size 3nm, chaotic vectors (VC3)
Output: (IV) second initial value
Begin
    For i =2 to 3nm
        $IV = IV \oplus YC(i) \oplus VC3(i)$
    End For
end
```

ii) Expression of the confusion function in the second round.

The second round can also be ensured by using another confusion function using the matrices of the first round. This function is given by equation 7.

$$Y(i) = AV_2\big(X(i)\big) = \begin{cases} \text{if } BV(i) = 0 \text{ then} \\ Y(i) \;\; = SM2\big(VC2(i), SM1\big(VC3(i), \mathrm{mod}(a * X(i) + b, 256) \oplus VC1(i)\big)\big) \\ \text{else} \\ Y(i) \;\; = SM1\Big(VC3(i), SM2\big(VC1(i), \mathrm{mod}(c * X(i) + d, 256) \oplus VC2(i)\big)\Big) \\ \text{endIf} \\ \text{with} \begin{cases} a = \mathrm{mod}\left(2 \, * \, \mathrm{mod}\left(\sum_{i=1}^{n} VC1(i)\,, 256\right) + 1{,}256\right) \\ c = \mathrm{mod}\left(2 \, * \, \mathrm{mod}\left(\sum_{i=1}^{m} VC2(i)\,, 256\right) + 1{,}256\right) \\ b = \; \mathrm{mod}\left(\sum_{i=1}^{3n} VC3(i)\,, 256\right) \\ d = \mathrm{mod}\left(\sum_{i=1}^{3m} VC3(i)\,, 256\right) \end{cases} \end{cases} \qquad (7)$$

Note that (AV_2) is the cipher function of the second round. (a,c) are two invertible factors in (G_{256}) and (d,b) two values of (G_{256}). The most judicious measure consists in making the functions defined by equations 8-9 reversible:

$$\begin{cases} h\colon \;\; G_{256} \longrightarrow G_{256} \\ \qquad x \mapsto \mathrm{mod}(ax \oplus b, 256) \end{cases} \qquad (8) \qquad \text{So} \begin{cases} h^{-1}\colon \;\; G_{256} \longrightarrow G_{256} \\ \qquad x \mapsto \mathrm{mod}(a^{-1}(x \oplus b), 256) \end{cases} \qquad (9)$$

The number of transformations (h) greatly exceeds $2^{127+256} \approx 2^{380}$. This guarantees strong protection against brutal attacks.

iii) Diffusion Function Expression in Second Round

The second round will be equipped with the diffusion function (Ω) provided by the substitution matrix (SM2). The expression of this function is defined by equation 10.

$$\forall\, i > 1 \quad \Omega\big(YC(i)\big) = SM2\big(VC1(i), Z(i-1) \oplus YC(i)\big) \qquad (10)$$

iv) The Second-Round Encryption Process:

The second-round encryption process is defined by Algorithm 9:

Algorithm 9: second round of Vigenere

```
Input: (YC) vector of size (3nm),
Output: (Z) vector of size (3nm)
Begin
  $Y'(1) = IV \oplus YC(1)$
  $Z(1) = AV_2(Y'(1))$
   For i =2 to 3nm
     $Y'(i) = \Omega\big(YC(i)\big)$
     $Z(i) = AV_2\big(Y'(i)\big)$
   End For
end
```

[Figure 9] shows the second-round encryption process.

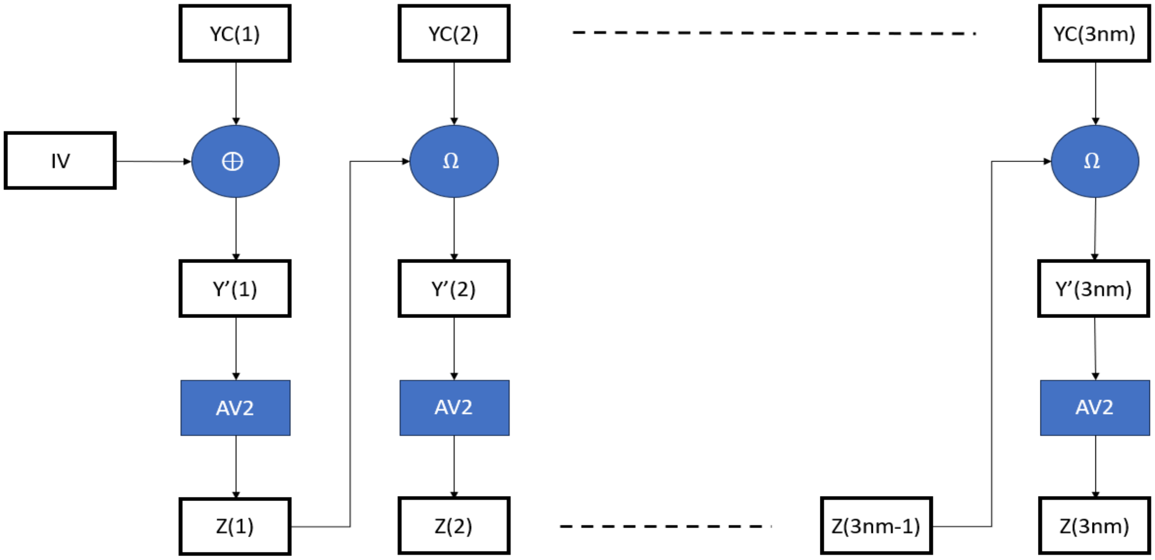
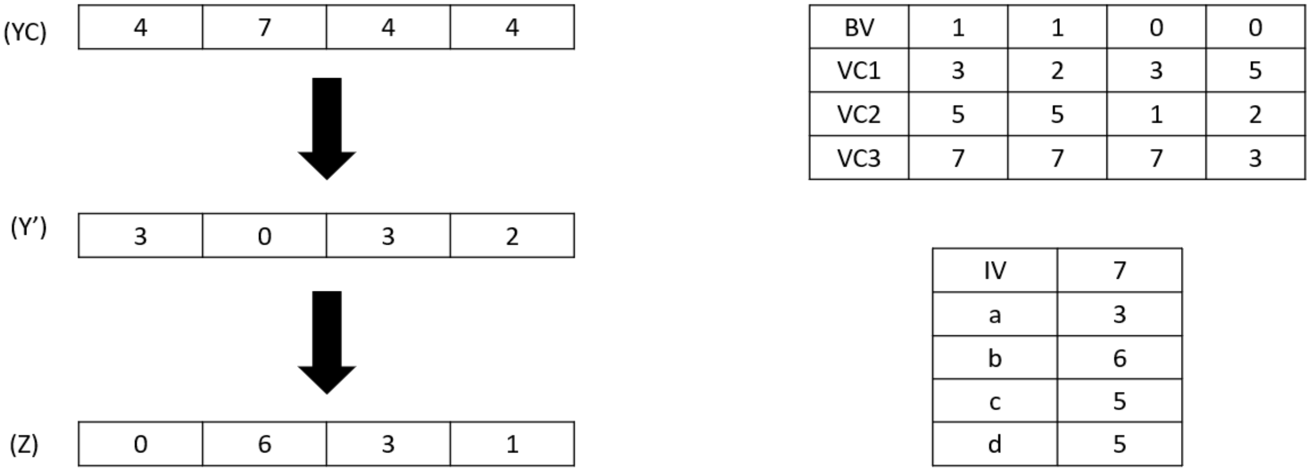


Figure 9: Second round of Vigenère

Example:

Let's consider the first 4 pixels of the vector (YC) taking values in G8. By using Algorithm 10 and the 2 S-boxes (SM1) and (SM2) mentioned in [Figure 6], we obtain the corresponding encrypted pixels stored in the vector (Z) after calculating the initialization value (IV). This process is controlled by the pseudo-random vectors (BV), (VC1), (VC2), and (VC3), along with the 4 pseudo-random parameters a, b, c, and d. [Figure 10] illustrates the progression of this process.



(YC)	4	7	4	4
(Y')	3	0	3	2
(Z)	0	6	3	1

BV	1	1	0	0
VC1	3	2	3	5
VC2	5	5	1	2
VC3	7	7	7	3

IV	7
a	3
b	6
c	5
d	5

Figure 10: Encryption Using Second Round of Enhanced Vigenère

The following operations are used to calculate the encrypted pixel corresponding to the original pixel with index 2 (YC(2)).

$$\begin{cases} Y'(2) = SM2\big(VC1(2), Z(1) \oplus YC(2)\big) \\ \quad = SM2(2, 0 \oplus 7) \\ \quad = SM2(2,7) \\ \quad = 0 \\ Z(2) = SM1\Big(VC3(2), SM2\big(VC1(2), mod(c * Y'(2) + d, 8) \oplus VC2(2)\big)\Big) \\ \quad = SM1\big(7, SM2(2, mod(5 * 0 + 5, 8) \oplus 5)\big) \\ \quad = SM1\big(7, SM2(2,5)) \oplus 5)\big) = SM1(7,6) \oplus 5 = 3 \oplus 5 = 6 \end{cases}$$

f) Applying a Global Permutation

To increase the time complexity of any attack, the vector obtained (Z) will be subjected to a global permutation (GP) algorithm 10.

```
Algorithm 10: global permutation
Input: (Z) vector of size (3nm),
Output: (ZC) vector of size (3nm)
Begin
    For i =1 to 3nm
        $ZC(i) = GP(Z(i))$
    End For
end
```

The vector (ZC) represent the image encrypted by our algorithm.

b. Decryption Process

Our technique is a symmetric encryption algorithm, uses the same secret key and the same encryption parameters. The decryption process begins with the last encryption operations with the use of inverse functions and must follow the following steps:

- Global inverse permutation.(PG).
- Inverse of the second Vigenère round ensured by the inverse substitution matrices (MS1) and (MS2).
- Recalculate initialization value (IV).
- Application of reverse genetic crossing ensured by the same table (CT).
- Inverse of the first Vigenère round ensured by the inverse substitution matrices (MS1) and (MS2).
- Reconstruction of the original image.

a) Generation of Reverse Encryption Parameters

i) Reverse Permutation:

The inverse permutation (PG) of (GP) is calculated from Algorithm 11.

```
Algorithm 11: global permutation design
Input: Global permutation (GP)
Output: Global inverse permutation (PG)
Begin
    For i =1 to 3nm
        $PG\big(GP(i)\big) = i$
    End For
end
```

ii) Inverse of the Substitution Matrix

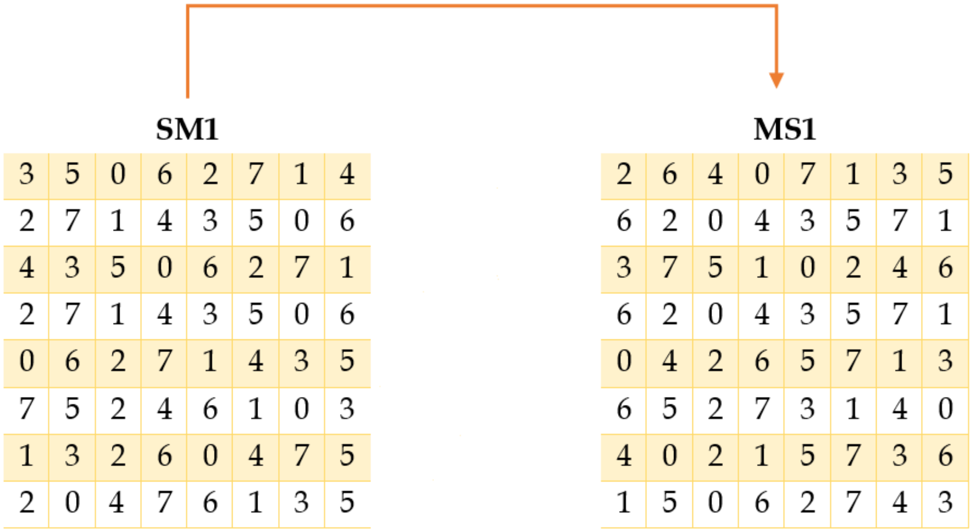
Each row of the substitution matrix is a permutation in (G_{256}), so the decryption matrix will consist of inverse permutations. For this reason, two generations of substitution matrices are provided by Algorithm 12:

Algorithm 12: inverse substitution matrix

```
Input: Two s-boxes (SM1) and (SM2)
Output: (MS1) and (MS2) two s-boxes
Begin
    For i =1 to 256
        For j=1 to 256
            $MS1(i, SM1(i, j)) = j$
            $MS2(i, SM2(i, j)) = j$
        End For
    End For
end
```

Example:

[Figure 11] represents the S-box SM1 taking values in G8 and its inverse MS1.



SM1

3	5	0	6	2	7	1	4
2	7	1	4	3	5	0	6
4	3	5	0	6	2	7	1
2	7	1	4	3	5	0	6
0	6	2	7	1	4	3	5
7	5	2	4	6	1	0	3
1	3	2	6	0	4	7	5
2	0	4	7	6	1	3	5

MS1

2	6	4	0	7	1	3	5
6	2	0	4	3	5	7	1
3	7	5	1	0	2	4	6
6	2	0	4	3	5	7	1
0	4	2	6	5	7	1	3
6	5	2	7	3	1	4	0
4	0	2	1	5	7	3	6
1	5	0	6	2	7	4	3

Figure 11: Substitution Matrix and Its Inverse in G_8

iii) Reverse Substitution

Following the same approach as the classical technique of Vigenère, the inverse substitution of the first round is given by equation 12, likewise for the second is defined by equation 11.

$$\begin{cases} \text{if } z = SM1(y, a * x \oplus b) \\ \text{Then} \\ x = a^{-1} * (MS1(y, z) \oplus b) \end{cases} \quad (11) \qquad \begin{cases} \text{if } z = SM2(y, x) \\ \text{Then} \\ x = MS2(y, z) \end{cases} \quad (12)$$

b) Decrypting the Encrypted Image

The algorithm we propose is a symmetric encryption system, meaning that the same key will be utilized during decryption. Furthermore, due to the diffusion function's implementation, the decryption process starts from the last pixel and proceeds towards the first pixel, necessitating the recalculation of the initialization value to obtain the precise value of the first pixel.

i) Application of Permutation (PG)

The vector (Z) is obtained by applying the inverse permutation (PG) to the vector (ZC), as illustrated in Algorithm 13:

```
Algorithm 13: global permutation
Input: (ZC) vector of size (3nm), (PG) the inverse permutation
Output: (Z) vector of size (3nm)
Begin
    For i =1 to 3nm
        $Z(i) = PG(ZC(i))$
    End For
end
```

ii) The Inverse Function of the Second Round

The inverse function of the second round is given by Algorithm 14:

```
Algorithm 14: inverse of the second round of Vigenère
Input: (Z) vector of size (3nm), two S-boxes (MS1) and (MS2), chaotic vectors (VC1), (VC2) and (VC3).
Output: (YC) vector of size (3nm)
Begin
    $a^{-1}$is the reverse of a in $G_{256}$
    $c^{-1}$is the reverse of c in $G_{256}$
    For i =3nm to 2
        If VB(i)=0 then
            $YC(i) = MS2\left(VC1(i), a^{-1} * \left(MS1\left(VC3(i), MS2(VC2(i), Z(i))\right) \oplus VC1(i) \oplus b\right)\right) \oplus Z(i-1)$
        Else
            $YC(i) = MS2\left(VC1(i), c^{-1} * \left(MS2\left(VC1(i), MS2(VC3(i), Z(i))\right) \oplus VC2(i) \oplus d\right)\right) \oplus Z(i-1)$
        End If
    End For
end
```

The computation of the initialization value (IV) will allow us to recover the exact value of pixel YC(1).

iii) The Reverse of the Genetic Crossing Operation

The inverse of the genetic crossover operation is given by Algorithm 15.

Algorithm 15: genetic crossing

```
Input: (YC) vector of size (3nm), VC1 chaotic vector and (CT) crossover table
Output: (Y) vector of size (3nm)
Begin
  For i =1 to n
    For j=2 to 3m
      $Y((CT(i,1)-1)*3m+j) = VC1((CT(i,2)-1)*3m+j) \oplus YC((CT(i,3)-1)*3m+j)$
    End For
  End For
end
```

iv) The Inverse Function of the First Round

The inverse function of the first round is given by Algorithm 16:

Algorithm 16: inverse of the first round of Vigenère

```
Input: (Y) vector of size (3nm), two S-boxes (MS1) and (MS2),
chaotic vectors (VC1), (VC2) and (VC3).
Output: (X) vector of size (3nm)
Begin
  For i =3nm to 1
    If VB(i)=0 then
      $X(i) = MS2\left(VC2(i), MS1(VC1(i), Y(i))\right) \oplus VC3(i)$
    Else
      $X(i) = MS1\left(VC3(i), MS2(VC2(i), Y(i))\right) \oplus VC1(i)$
    End If
  End For
end
```

The original image (img) is retrieved through the XOR operation between the vector (X) and the chaotic vector (VC1). This operation is described by algorithm 17.

Algorithm 17: First confusion of the original image

```
Input: (X) vector and chaotic vector (VC1)
Output: original image (img)
Begin
  For i =1 to 3nm
    $img(i) = X(i) \oplus VC1(i)$
  end For
end
```

4. Examples and Simulations:

To measure the performance of our encryption system, we randomly select a large number of reference images and use them for testing our method. In this section, All simulations were performed using the Python programming language under Ubuntu 20.04, on a basic personal computer equipped with an i5 processor, 8 GB of RAM, and a 500 GB hard drive.

1) Key Space Analysis

The chaotic sequences used in our method ensure a high sensitivity to initial conditions and can protect against brute-force attacks. The secret key of our system is composed of...

- u_0=0,8753418021, μ_1=3.97451652, for the logistics map.
- v_0=0,356000001,v_1=0,956100001 b=1.09231541 for the map of Henon.

If we use single-precision real numbers **10^{-10}** for the calculation, the total key size will greatly exceed ≈2^{180} 2^{110}, which is enough to avoid brute force attacks.

2) Secret Key Sensitivity Analysis

Our encryption key has high sensitivity, which means that a slight modification of a single parameter will automatically lead to a significant difference compared to the original image. [Figure 12] illustrates this property, ensuring that in the absence of the correct encryption key, the original image cannot be restored.

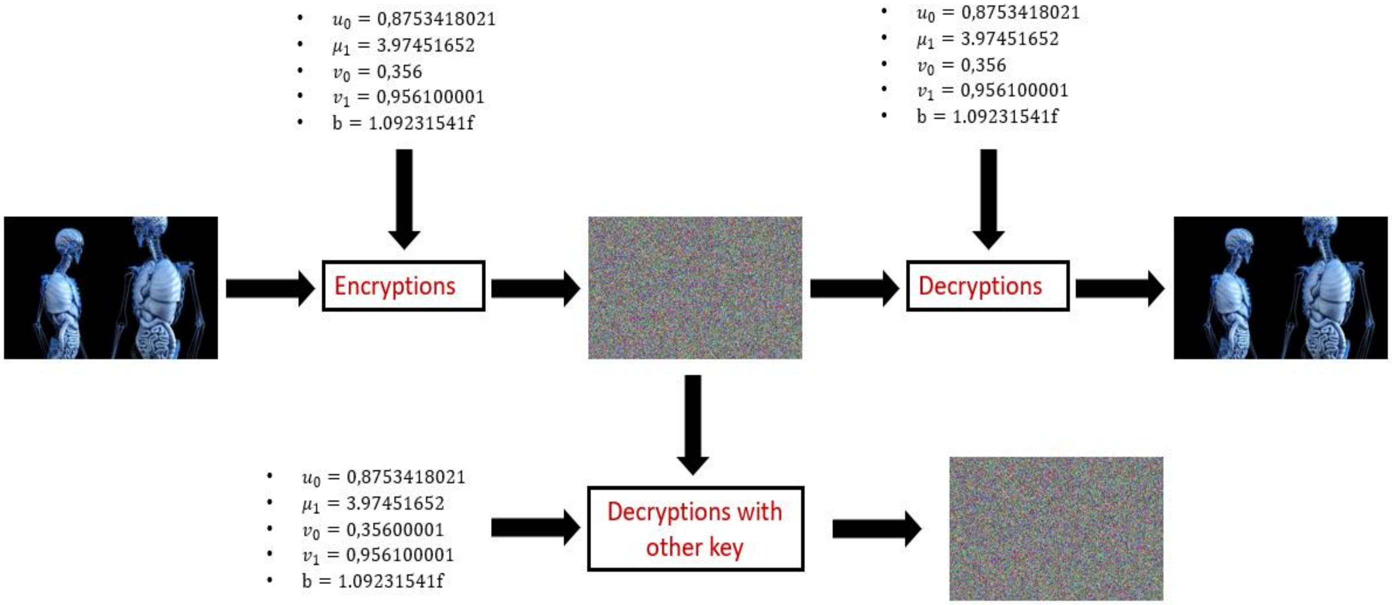


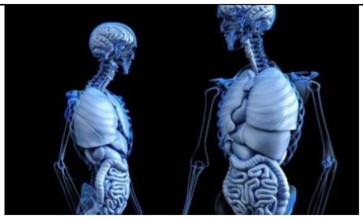
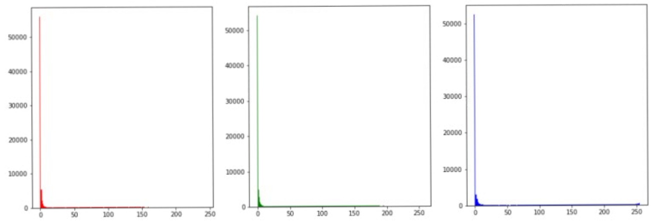
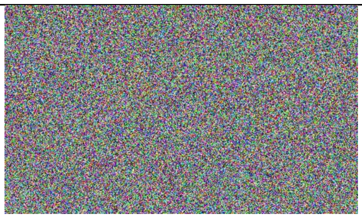
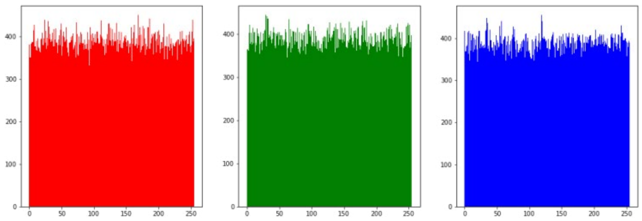
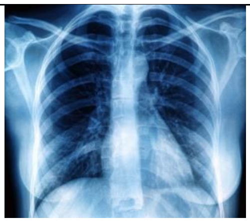
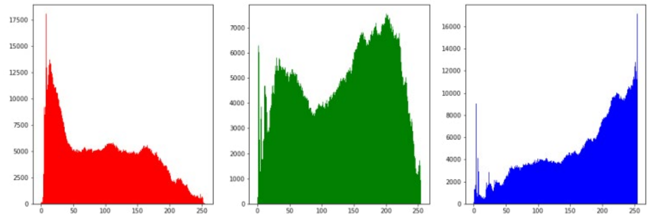
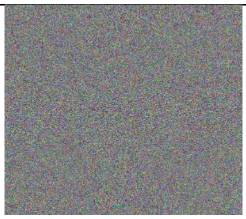
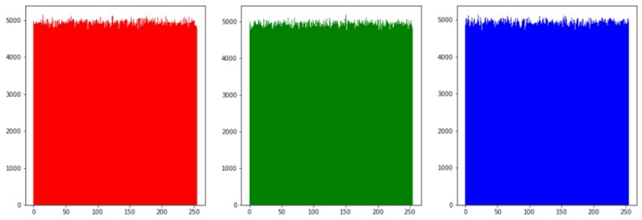
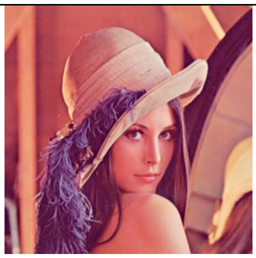
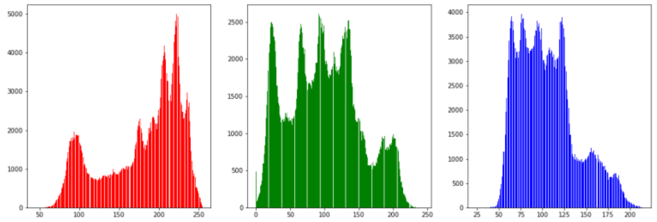
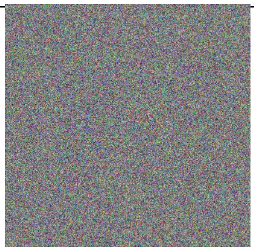
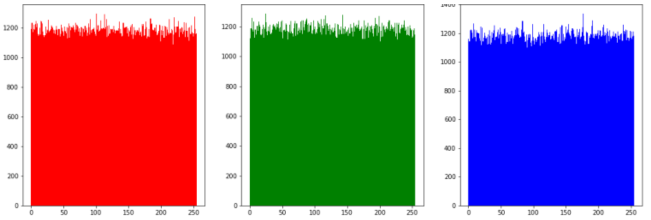
Figure 12: Encryption Key Sensitivity

3) Security Against Statistical Attacks:

a) Simulation and Histogram

[Table 4] provides examples of images before and after encryption. This illustration demonstrates that no visual information can be extracted from the encrypted image.

All the images tested by our algorithm have a uniformly distributed histogram [Table 4]. This indicates that the entropy of the encrypted images is around 8, making the system immune to statistical attacks.

name	size	images		Histograms
Img1	243x411	original iamge		
		Encrypted image		
img2		original image		
		Encrypted image		
img3	552x550	original image		
		Encrypted image		

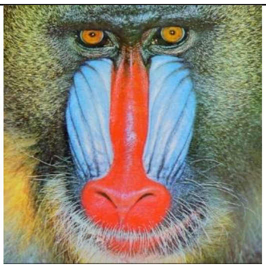
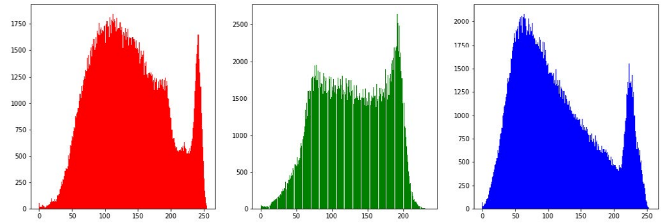
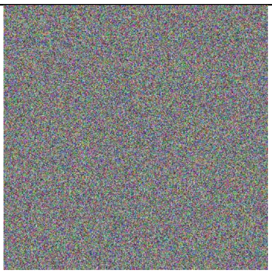
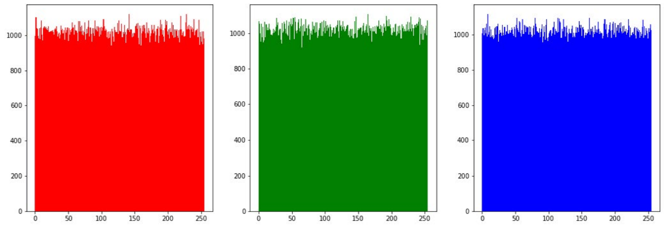
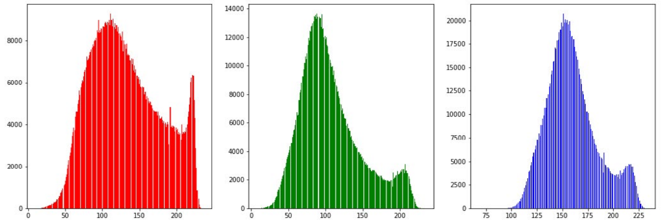
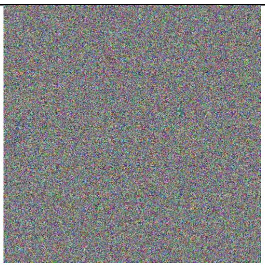
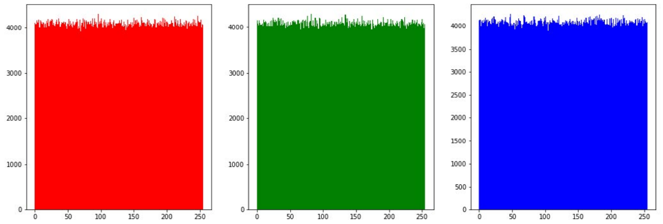
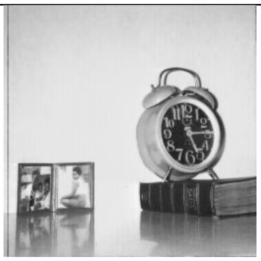
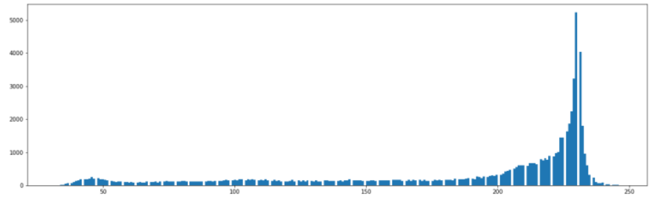
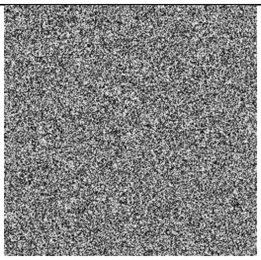
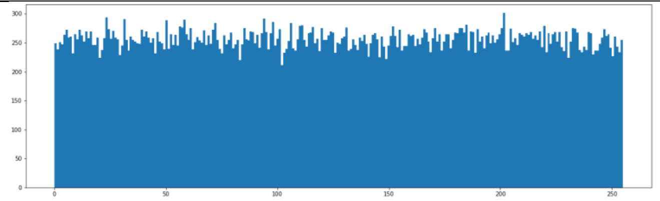
Img4	512x512	original image		
		Encrypted image		
Img5	1024x1024	original image		
		Encrypted image		
Img6	256x2565	original image		
		Encrypted image		

Table 4: Original images and Cipher images histograms ‘

b) Entropy Analysis

The entropy of a size image (n,m) is given by equation 14:

$$H(MC) = \frac{1}{256}\sum_{i=1}^{256} -p(i)\log_2\big(p(i)\big) \qquad (14)$$

p(i)is the probability of occurrence of level (i)in original image. All the images that our method tested have an entropy very close to 8 [Table 5], which is the maximum value. This parameter ensures that entropy attacks are avoided by our system.

c) Correlation Analysis

The correlation of a size image (n,m)is given by equation 15.

$$r = \frac{\mathrm{cov}(x, y)}{\sqrt{V(x)}\sqrt{V(y)}} \qquad (15)$$

Pixel correlation measures the independence of neighboring pixels. the correlation values obtained are very close to zero for all the images encrypted by our method [Table 5]. This parameter ensures that statistical attacks are avoided by our system.

Pictures		Vertical correlation	Diagonal correlation	Horizontal correlation	Entropy
Img1	R	0.00024	-8.37e-05	-0.00213	7.998032
	G	-0.00146	0.00062	0.00160	7.998075
	B	-0.00204	-0.00302	0.00130	7.998090
img2	R	0.00012	-0.00038	7.06e-05	7.999635
	G	-0.00182	-0.00286	-0.00052	7.999646
	B	0.00241	-0.00314	0.00162	7.999617
img3	R	0.00015	-0.00074	0.00237	7.999454
	G	-0.00208	0.00295	-0.00014	7.999393
	B	0.00223	0.00187	-4.01e-05	7.999333
Img4	R	-0.00323	0.00386	-0.00293	7.999295
	G	0.00084	0.00061	0.00104	7.999275
	B	-0.00059	-0.00286	0.00137	7.999422
Img5	R	-0.00059	-0.00052	-0.00020	7.999825
	G	0.00049	-0.00142	-0.00127	7.999831
	B	0.00081	-0.00155	-0.00035	7.999843
Img6		0.00051	-4.96e-05	0.00122	7.997465

Table 5: Correlation and Entropy of some tested images

4) Security Against Differential Attacks

In cryptography, the analysis of differential attacks is managed by the following parameters:

a) The NPCR Constant

the Number of Pixels Change Rate (NPCR) is determined by equation 16.

$$\mathrm{NPCR} = \left(\frac{1}{\mathrm{nm}}\sum_{\mathrm{i,j=1}}^{\mathrm{nm}} \mathrm{D(i,j)}\right) * 100 \qquad (16)$$

$$\text{With} \quad \mathrm{D(i,j)} = \begin{cases} 1 & \text{if} \quad \mathrm{C_1(i,j) \neq C_2(i,j)} \\ 0 & \text{if} \quad \mathrm{C_1(i,j) = C_2(i,j)} \end{cases}$$

b) The UACI Constant

Unified Average Changing Intensity (UACI) [24] mathematical analysis of an image is given by equation 17.

$$\mathrm{UACI} = \left(\frac{1}{\mathrm{nm}}\sum_{\mathrm{i,j=1}}^{\mathrm{nm}} \mathrm{Abs}\big(\mathrm{C_1(i,j) - C_2(i,j)}\big)\right) * 100 \qquad (17)$$

c) Avalanche Effect

Our approach is based on a CBC operating mode. Therefore, any small change in the original image results on a larger change in the pixels of the encrypted image. The avalanche effect corresponds to the number of bits that have been modified if a single bit of the original image is modified [25]. The mathematical expression of this avalanche effect is given by equation 18.

$$\mathrm{AE} = \left(\frac{\sum_{\mathrm{i}} \text{bit change}}{\sum_{\mathrm{i}} \text{bit total}}\right) * 100 \qquad (18)$$

[Table 6] shows that the differential parameters results obtained in the desired ranges (NPCR close to 100%, UACI > 33% and EA > 50%).

Pictures	NPCR	UACI	EA
Img1	100.0 %	33.6861%	50.2609%
Img2	100.0 %	33.5820%	50.2015%
Img3	100.0 %	33.5439%	50.1484%
Img4	100.0 %	33.5682%	50.2117%
Img5	100.0 %	33.5971%	50.2013%
Img6	100.0 %	33.5470%	50.2260%

Table 6: Differential Parameters

5) Encryption Time Complexity

The encryption time is a crucial benchmark for assessing the efficiency of an image encryption algorithm. Effectively encrypting substantial data, such as images, within a reasonable timeframe has become a challenging aspect of algorithm development. In our study, we present the encryption times for images of sizes 256x256 and 512x512 in [Table 7], along with a comparison to other recent works. Furthermore, the time complexity of our method for an image of size (N, M) is O(NM).

images	Our algorithm	Ref [35]	Ref [36]	Ref [37]
256x256	0.6956	0.65	8.22	0.156
512x512	1.7551	----	----	0.406

Table 7: Encryption Time

6) Comparison With Other Approaches

In [Table 8], we provide a comparison between the entropy and differential parameters of several images encrypted using our approach and the same images encrypted using other algorithms.

images	entropy				NPCR				UACI			
	Our method	Ref [38]	Ref [39]	Ref [40]	Our method	Ref [38]	Ref [39]	Ref [40]	Our method	Ref [38]	Ref [39]	Ref [40]
baboon	7,9993	7.9984	----	---	100.0%	99.60%	----	----	33.56%	33.26%	----	-----
lena	7,9993	----	7.9976	7.9992	100.0%	----	99.61%	99,65%	33.54%	----	33.51%	33,48%

Table 8: Comparison with other methods

5. Conclusion

In this article, we have proposed a novel medical image encryption scheme based on two Vigenère encryption rounds using large S-boxes, separated by a genetic crossover adapted for image encryption and a permutation applied to the output vector [35-40]. The results obtained by testing several randomly chosen images from a database have shown very promising and encouraging outcomes, ensuring better protection against any known attacks. As the time complexity of our approach is polynomial, the execution time in the encryption and decryption processes is reasonable. In our perspective, we plan to integrate genetic algorithms acting at the level of DNA and RNA to enhance the security of our system.

Declarations

Conflict of Interest

All authors of this article confirming the absence of any conflict between them, and there are no private or public organizations or laboratories to fund this research, thus avoiding any expected conflicts.

This document does not contain any research or experiments conducted on animals or humans

Ethical Approval

This declaration is “not applicable”.

Competing Interests

The authors declare that they have no competing interests.

Authors’ Contributions

All authors: carried out the experiment.

Mariem Jarjar; Faiq Gmira; Said Hraoui: wrote the manuscript with support from F.S.

All authors: helped supervise the project and conceived the original idea.

Abdellatif Jarjar and Abdelhamid Benazzi: reviewed the manuscript.

Funding

No funding was received to assist with the preparation of this manuscript.

Availability of Data and Materials

This declaration is “not applicable”.

References

1. Qobbi, Y., Jarjar, A., Essaid, M., & Benazzi, A. (2022). New image encryption scheme based on dynamic substitution and hill cipher. In *WITS 2020: Proceedings of the 6th International Conference on Wireless Technologies, Embedded, and Intelligent Systems* (pp. 797-808). Springer Singapore.
2. 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.
3. 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.
4. Boussif, M., Aloui, N., & Cherif, A. (2020). Securing DICOM images by a new encryption algorithm using Arnold transform and Vigenère cipher. *IET Image Processing*, 14(6), 1209-1216.
5. Jarjar, M., Hraoui, S., Najah, S., & Zenkouar, K. (2022). New technology of color image encryption based on chaos and two improved Vigenère steps. *Multimedia Tools and Applications,* 81(17), 24665-24689.
6. Uniyal, N., Dobhal, G., Rawat, A., & Sikander, A. (2021). A novel encryption approach based on vigenere cipher for secure data communication. *Wireless Personal Communications,* 119, 1577-1587.

7. JarJar, A. (2022). Vigenere and genetic cross-over acting at the restricted ASCII code level for color image encryption. *Medical & Biological Engineering & Computing*, 60(7), 2077-2093.
8. Feng, W., Qin, Z., Zhang, J., & Ahmad, M. (2021). Cryptanalysis and improvement of the image encryption scheme based on Feistel network and dynamic DNA encoding. *IEEE Access, 9*, 145459-145470.
9. Fang, P., Liu, H., Wu, C., & Liu, M. (2021). A secure chaotic block image encryption algorithm using generative adversarial networks and DNA sequence coding. *Mathematical Problems in Engineering,* 2021, 1-26.
10. Razaq, A., Iqra, Ahmad, M., Yousaf, M. A., & Masood, S. (2021). A novel finite rings based algebraic scheme of evolving secure S-boxes for images encryption. *Multimedia Tools and Applications, 80,* 20191-20215.
11. Zhu, S., Wang, G., & Zhu, C. (2019). A secure and fast image encryption scheme based on double chaotic S-boxes. *Entropy*, 21(8), 790.
12. Brahim, A. H., Pacha, A. A., & Said, N. H. (2023). A new image encryption scheme based on a hyperchaotic system & multi specific S-boxes. *Information Security Journal: A Global Perspective,* 32(2), 59-75.
13. Belazi, A., Khan, M., El-Latif, A. A. A., & Belghith, S. (2017). Efficient cryptosystem approaches: S-boxes and permutation–substitution-based encryption. *Nonlinear Dynamics,* 87, 337-361.
14. Ullah, A., Jamal, S. S., & Shah, T. (2017). A novel construction of substitution box using a combination of chaotic maps with improved chaotic range. *Nonlinear Dynamics,* 88, 2757-2769.
15. Zhang, Y. (2018). The unified image encryption algorithm based on chaos and cubic S-Box. *Information Sciences*, 450, 361-377.
16. Qobbi, Y., Jarjar, A., Essaid, M., & Benazzi, A. (2021, January). Development of large chaotic S-boxes for image encryption. In *International Conference on Digital Technologies and Applications* (pp. 847-858). Cham: Springer International Publishing.
17. Arif, J., Khan, M. A., Ghaleb, B., Ahmad, J., Munir, A., Rashid, U., & Al-Dubai, A. Y. (2022). A novel chaotic permutation-substitution image encryption scheme based on logistic map and random substitution. *IEEE Access*, 10, 12966-12982.
18. Kumar, M., & Gupta, P. (2021). A new medical image encryption algorithm based on the 1D logistic map associated with pseudo-random numbers. *Multimedia Tools and Applications,* 80(12), 18941-18967.
19. Khan, J., Ahmad, J., & Hwang, S. O. (2015, May). An efficient image encryption scheme based on: Henon map, skew tent map and S-Box. In *2015 6th International conference on modeling, simulation, and applied optimization (ICMSAO)* (pp. 1-6). IEEE.
20. Ping, P., Xu, F., Mao, Y., & Wang, Z. (2018). Designing permutation–substitution image encryption networks with Henon map. *Neurocomputing*, 283, 53-63.
21. Umbarkar, A. J., & Sheth, P. D. (2015). Crossover operators in genetic algorithms: a review. *ICTACT journal on soft computing,* 6(1).
22. Kaur, M., & Kumar, V. (2020). A comprehensive review on image encryption techniques. *Archives of Computational Methods in Engineering,* 27, 15-43.
23. Abraham, L., & Daniel, N. (2013). Secure image encryption algorithms: A review. *International journal of scientific & technology research*, 2(4), 186-189.
24. Wu, Y., Noonan, J. P., & Agaian, S. (2011). NPCR and UACI randomness tests for image encryption. Cyber journals: multidisciplinary *journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT)*, 1(2), 31-38.
25. Budiman, F., Andono, P. N., & Setiadi, M. (2022). Image Encryption using Double Layer Chaos with Dynamic Iteration and Rotation Pattern. *International Journal of Intelligent Engineering & Systems*, 15(2).
26. Ali, F. M. S., & Sarhan, F. H. (2014). Enhancing security of vigenere cipher by stream cipher. *International Journal of Computer Applications,* 100(1), 1-4.
27. Arqub, O. A., & Abo-Hammour, Z. (2014). Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. *Information sciences,* 279, 396-415.
28. Abo-Hammour, Z. E., Alsmadi, O., Momani, S., & Abu Arqub, O. (2013). A genetic algorithm approach for prediction of linear dynamical systems. *Mathematical Problems in Engineering.*
29. Abo-Hammour, Z., Abu Arqub, O., Momani, S., & Shawagfeh, N. (2014). Optimization solution of Troesch's and Bratu's problems of ordinary type using novel continuous genetic algorithm. Discrete Dynamics in Nature and Society.
30. Abu Arqub, O., Abo-Hammour, Z., Momani, S., & Shawagfeh, N. (2012, January). Solving singular two-point boundary value problems using continuous genetic algorithm. *In Abstract and applied analysis* (Vol. 2012). Hindawi.
31. Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering*, 157, 107250.
32. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering,* 376, 113609.
33. Abualigah, L., & Diabat, A. (2021). Advances in sine cosine algorithm: a comprehensive survey. *Artificial Intelligence Review,* 54(4), 2567-2608.
34. Abualigah, L. M. Q. (2019). Feature selection and enhanced krill herd algorithm for text document clustering.
35. Khan, J. S., & Ahmad, J. (2019). Chaos based efficient selective image encryption. *Multidimensional Systems and Signal Processing,* 30, 943-961.
36. Ayoup, A. M., Hussein, A. H., & Attia, M. A. (2016). Efficient

selective image encryption. *Multimedia tools and applications,* 75, 17171-17186.

37. Laiphrakpam, D. S., & Khumanthem, M. S. (2017). Medical image encryption based on improved ElGamal encryption technique. *Optik*, 147, 88-102.
38. Mahmud, M., Lee, M., & Choi, J. Y. (2020). Evolutionary-based image encryption using RNA codons truth table. *Optics & Laser Technology*, 121, 105818.
39. Mondal, B., & Mandal, T. (2020). A secure image encryption scheme based on genetic operations and a new hybrid pseudo random number generator. *Multimedia Tools and Applications,* 79(25-26), 17497-17520.
40. Es-Sabry, M., El Akkad, N., Merras, M., Saaidi, A., & Satori, K. (2022). A new color image encryption algorithm using multiple chaotic maps with the intersecting planes method. *Scientific African*, 16, e01217.