Image Encryption Algorithm Based On Genetic Crossover And Chaotic DNA Encoding

Younes Qobbi (qobbi.younes@ump.ac.ma)
Mohammed I University Oujda: Universite Mohammed Premier Oujda https://orcid.org/0000-0002-3533-7960

Abdeltif jarjar
High School Moulay Rachid

Mohamed Essaid
Universite Sidi Mohamed Ben Abdallah

Abdelhamid Benazzi
Mohammed I University Oujda: Universite Mohammed Premier Oujda

Research Article

Keywords: Chaotic map, Genetic crossover, Image encryption, DNA encoding

Posted Date: August 9th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-672343/v1

Image Encryption Algorithm based on Genetic Crossover and Chaotic DNA Encoding

Younes Qobbi[1*], Abdeltif Jarjar[2] , Mohamed Essaid[3] and Abdelhamid Benazzi[1]

[1] Mohamed First University, HSTO, AMSPCS Laboratory, Oujda, Morocco.
[2] High School Moulay Rachid, Taza Morocco
[3] Sidi Mohamed Ben Abdellah University, LSI, Taza, Morocco

qobbi.younes@ump.ac.ma

Abstract

Based on the two-dimensional logistic map and a single improved genetic operator, a new image encryption system is proposed. The original image is transformed into DNA sequences, a subdivision into blocks of size calculated by using the chaotic map, with the intention to apply a crossover between blocks chaotically selected from a chaotic control vectors. For the installation of a diffusion phase, a strong link is established between the block resulting from a crossing operation and the next original block. Hoping to considerably increase the impact of the avalanche effect and protect the system against any differential attack. Simulations performed on a large number of images of different size and formats ensure that our method is not subject to any known attacks.

Keywords: Chaotic map, Genetic crossover, Image encryption, DNA encoding.

I. Introduction

With high speed development of information and communication technology, digital image encryption based on chaos theory has attracted the attention of researchers in the field of computer security. Therefore, many crypto-systems using chaotic systems have been proposed [1, 20, 21]. Chaotic maps, for example, tent map, Arnold map and logistic map are widely used in digital image encryption. These chaotic systems are characterized by a high sensitivity to initial conditions and control parameters and are used to generate the chaotic sequences which are exploited to create permutations and also establish the confusion and diffusion process [2, 6, 17, 18].

Based on chaos, author for article [3] propose a new image encryption scheme using chaotic permutation and diffusion process. In reference [4], the authors propose a new hybrid chaotic cryptosystem using new mathematical function to improve the chaotic behaviour of chaotic map to produce the key stream sequences with excellent pseudo-randomness characteristics. The obtained chaotic map is used in confusion-diffusion processes. In reference [5], the authors implemented a new S-box for image encryption; the authors affirm that the S-box is validate and consequently suitable for an efficient image encryption scheme**.** A combination between chaotic systems and genetic algorithm (GA) is a novel approach in the field of cryptography [6]. The authors of work [7] present an encryption system consists of three steps. Firstly a permutation of pixels of the plain image is established by using the Chen's chaotic map. At second, a diffusion process is performed by using the Logistic-Sine map to change the gray level values of pixels. Finally a genetic algorithm is used to select the best encrypted image. The authors of [8] propose a new image encryption algorithm based on the confusion-diffusion process, which are established a bit permutation and genetic operations, such as selection, crossover and mutation at the DNA nucleotides sequences. K.C. Jithin and al [9] proposed an image encryption scheme based on three major subsystems. In the first subsystem the selected algorithm is used to choose the good chaotic map. In the second a DNA encoding for three channels (R, G and B) applied to obtain three DNA sequences for (R, G and B) channels. In the third subsystem two main processes for any image encryption system are established, which are Confusion-diffusion. In the article [10], the authors proposed an encryption system combining the use of a double chaotic system and DNA encryption. The double chaotic system is composed of a CML (Coupled Map Lattice) and an optical chaos. The use of this new system makes it possible to obtain a chaotic system of great dimensions. DNA encryption is used to increase the security of the proposed system. The authors of [11] used genetic operations, such as mutation and crossover. These operations are controlled by a 2DNLCML system. This system makes it possible to generate better chaotic sequences. Saeed Noshadian and al [12] proposed a new encryption algorithm using a genetic algorithm controlled by the logistic map. The original image is subdivided into four equal parts. The first five pixels of each part are used to generate initial values. XOR Operation is used between the pixels of each block and the elements of the chaotic sequence to generate the starting population. Then genetic operations are applied to create new generations by using the correlation coefficient as fitness function. Some authors [13] propose two similar encryption and decryption systems, using hyper chaotic systems and mtDNA (mitochondrial DNA). In this system the image is decomposed into several cluster, the encrypted image is obtained by application of two main processes of each encryption system which are confusion and diffusion.

Some image encryption algorithms are applicable only on square images. To overcome this weakness, we propose an image encryption system suitable for any size of images. This image encryption using adapted genetic crossover and chaotic systems. After encoding the original image in DNA sequence, this image is decomposed into blocks of size calculated chaotically. In the next step, the operation of crossing between two randomly selected blocks from the plain image and chaotic DNA sequence generated by the chaotic systems used. Then a diffusion process is performed by connecting the encrypted block by the next original one using a new genetic operator.

II. Preliminaries.

In this part, we present the fundamental concepts of our work such as the logistic map, genetic operations and DNA encoding.

1. 2D Logistic map.

In order to build a new image encryption algorithm using a single encryption key, we will use the 2D logistic map [14-15]. This choice is due to the simplicity of its development and its high sensitivity to the initial values. It's mathematically expressed as:

$$\begin{cases} x_{n+1} = \mu_1 x_n(1 - x_n) + \mu_2 y_n^2 \\ y_{n+1} = \mu_3 y_n(1 - y_n) + \mu_4 x_n^2 \end{cases} \qquad (\mathbf{1})$$

Where x_0 , y_0 $\in]0 \quad 1]$ and $\mu_1 \in [2.75 \quad 3.4]$, $\mu_2 \in [0.15 \quad 0.21]$, $\mu_3 \in [2.75 \quad 3.45]$, $\mu_4 \in [0.13 \quad 0.15]$. All this conditions ensure the installation of chaotic behavior.

2. Genetic operations.

2.1. Mutation operation.

Genetic mutation is an exchange of nucleotides between two DNA genes from a randomly determined position to a stop codon described by RNA. In our situation, mutation is a change of bits between two blocks from a certain rank.

2.2. Inversing Operation.

Genetic inversion is the change in the values of the nucleotides in their complements from a random position to the meeting of a stop codon. In our approach, the genetic function Inversion acts on the concatenation (WB) of the two output blocks of the mutation by changing the bits from a position.

2.3. Reversion Operation.

Random process which consists in a change of value of nucleotides, it is a passage to the complimentary.

2.4. Crossover Operation.

Genetic Crossover operator is a random process that involves the integration of genes from both parents in randomly selected locations to facilitate the transfer of values from the parents to the new generation.

III. Proposed Method.

In this contribution we are going to develop an improved genetic crossover for color image encryption system. This dynamic crossover will be applied to the chaotically selected blocks from the control vector. This new cryptosystem is based on the following axes:

- Pseudo-random sequences generation.
- Plain Image Preparation.
- Key stream generation.
- Encryption Decryption Process.
- Simulation and Results.

1. Pseudo-random sequences generation.

The 2D logistic map is iterated for $12 \times nm$ times, to generate two sequences of pseudo-random numbers x and y of size$12 \times nm$. Where $n\, and\, m$ are respectively the height and width of the plain image.

1.1. Control Vector Development.

This vector noted $(\boldsymbol{Br})$ of size $(1\,, 12 \times nm)$ constructed by using the following algorithm:

```
For i=0 to 12nm-1
    If (x(i)>y(i))  then
      Br(i)=0
    Else
```

```
      Br(i)=1
    End
Next i
```

2. Plain Image Preparation.

After the three (RGB) color channels extraction and their conversion into three vectors $(\mathrm{Vr}), (\mathrm{Vg}), (\mathrm{Vb})$ of $\mathrm{size}(1, \mathrm{nm})$, a concatenation is established to generate a vector $(\mathrm{V1})$ of $\mathrm{size}(1{,}3\mathrm{nm})$. This operation is described by the following algorithm:

```
For i=0 to nm-1
  V1(3i)=Vr(i)
  V1(3i+1)=Vg(i)
  V1(3i+2)=Vb(i)
Next i
```

2.1. DNA Encoding

The DNA molecule is composed of four bases Adenine (A), guanine (G), cytosine (C) and thymine (T). In biology (A) is the complement of (T) and (C) is the complement of (G). In the binary system 00 is complement of 11 and 01 is complement of 10. To respect the complement rule [16], we have 8 possibilities to encode the four bases of DNA. DNA encoding rules is described by the table below (Table.1):

Table.1.DNA Encoding Ruls (*T*)

	A	T	C	G
Rule 1	00	11	01	10
Rule 2	00	11	10	01
Rule 3	01	10	00	11
Rule 4	01	10	11	00
Rule 5	10	01	00	11
Rule 6	10	01	11	00
Rule 7	11	00	01	10
Rule 8	11	00	10	01

2.2. Selected Rule.

Any chaos-based, fixed-encoding image encryption system is vulnerable to plain text attacks [19]. In our work, the DNA encoding rule of the original image is randomly selected. The result is stored in a vector (IV) of size $(1, 4)$ by using the following algorithm:

```
int k=mod((int)(x(n)×10^8),8)
For i=0 to 3
  IV(i)= T(k , i)
Next i
```

Example:

If $k = 6$ then: $A{:}11 \quad T{:}00, \quad C{:}01 \ and \ G{:}10.$

In our work, in order to provide a chaotic aspect to DNA encoding, we consider the following two chaotic tables:

Table.2. Chaotic table (TP1)

$x(2n)$	$x(n)$	$x(nm)$	$y(m)$
$x(2m)$	$y(n)$	$x(nm)$	$x(m)$
$y(nm)$	$x(n)$	$y(2n)$	$y(m)$
$x(nm)$	*x (2m)*	*y(m)*	*y(n)*

Table.3. Chaotic table (TP2)

y(2n)	*x (n)*	*y(nm)*	*y(m)*
x (2n)	*y(n+m)*	*x (nm)*	*y(n)*
x (2m)	*y(n)*	*x (n)*	*y(nm)*
x (2nm)	*x (n)*	*y(2nm)*	*y(n+m)*

- ➢ The table $(TP1)$ is arranged in descending order of rows to generate a $(TC1)$table.
- ➢ The table $(TP2)$ is arranged in ascending order of rows to generate a $(TC2)$table.

```
For i=0 to 3
  For j=0 to 3
    TC1(i , j)=IV(TP1(i ,j))
    TC2(i , j)=IV(TP2(i ,j))
Next j ,i
```

2.3. DNA Transformation.

The pixels of the original image are integer values in the range$[\![0\ \ 255]\!]$. Each pixel is encoded on four nucleotides. Then the transformation of a pixel $V1(i)$ into DNA sequence is carried out by the algorithm as bellow:

passage in hexadecimal

```
 int x=(int)(V1(i)/16)
int y=V1(i)-16×x
```

passage in nucleotides

```
int a=(int)(x/4)
int b=x-4×a
int c=(int)(y/4)
     Int d=y-4×c
```

The vector $(V2)$ of size $(12\ \times\ nm)$ is constructed by:

```
If (Br(i)==0)then
    V2(4i)=TC1(0,a)
    V2(4i+1)=TC1(1,b)
    V2(4i+2)=TC2(0,c)
    V2(4i+3)=TC2(1,d)
Else
    V2(4i)=TC1(2,a)
    V2(4i+1)=TC1(3,b)
    V2(4i+2)=TC2(2,c)
    V2(4i+3)=TC2(1,d)
```

This passage in nucleotides is controlled by the vector (Br).

3. Key stream generation.

In our encryption algorithm, we use two chaotic DNA sequences (XN) and (YN) of size$(12\times nm)$, generated by the following algorithm:

```
For i= 0 to 12nm-1
    For k=3 to 0
    If(y(i)-(k/4)>0 then
      If (Br(i)=0)
        XN(i)=TC2(2 ,k)
        YN(i)=TC1(2 ,k)
      Else
        XN(i)=TC2(1 ,k)
        YN(i)=TC1(3 ,k)
Next i
```

3.1. DNA algebraic operations.

By arranging each row in descending order, the matrix $(TP3)$ is transformed into $(GO1)$ the first table of algebraic operator$(\otimes)$.

Table.4. Chaotic table (TP3)

x(n)	x(nm)	y(nm)	y(m)
x(m)	y(n)	x(2m)	x(2n)
y(m)	x(nm)	y(2n)	y(nm)
x(nm)	x (2m)	y(m)	y(n)

Table.5. First algebraic operation $(GO1)$

⊗	**T**	**C**	**G**	**A**
T	IV(TP3(0,0))	IV(TP3(0,1))	IV(TP3(0,2))	IV(TP3(0,3))
C	IV(TP3(1,0))	IV(TP3(1,1))	IV(TP3(1,2))	IV(TP3(1,3))
G	IV(TP3(2,0))	IV(TP3(2,1))	IV(TP3(2,2))	IV(TP3(2,3))
A	IV(TP3(3,0))	IV(TP3(3,1))	IV(TP3(3,2))	IV(TP3(3,3))

In the same way, the chaotic table $(TP4)$ of size $(4\,,,4)$ is transformed into second algebraic operation table $(GO2)$ noted $(\boxtimes)$by passing each row arranged in descending order.

3.2. Block size Computing.

The encryption algorithm proposed in this manuscript is a block cipher system. The size of this block is calculated by the following expression:

$$r = 10 * \left(mod\left(\textstyle\sum_{i}^{n}\left(XN(i) + YN(i)\right), 15\right) + 10\right) \quad (\mathbf{2})$$

Then $\quad 100 \leq r \leq 240$

We note that; $\quad r = 10 \times q \quad (\mathbf{3})$

Where q is the sub-block, which contains 10 nucleotides?

3.3. Size Vector Adaptation.

The new vector size (VN) can be obtained by the following formula:

$$\begin{aligned} let\quad 12nm &\equiv s\ [\boldsymbol{r}] \\ L &= 12nm - s \\ t &= \tfrac{L}{r} \end{aligned} \quad (\mathbf{4})$$

By truncating the last (s) $(V2)$ pixels, and placing it in the (Vs) vector of size $(1, s)$ after modification, the (VN) adaptive vector can be obtained by the following algorithm:

```
For i= 0 to L
    VN(i)=V2(i)
Next i
If (s<>0) then
  For i=0 to s-1
```

```
If (Br(i+l)=0) then
    Vs(i)= V2(i+L)⊗ XN(i+L)
Else
    Vs(i)=V2(i+L) ⊠ YN(i+L)
Next i
```

3.4. *Initialization value Design*

The purpose of this part is to eliminate any differential attacks and to promote the implementation of genetic hybridization. The(Iv) initialization value must be recalculated to change the value of the starting block. It's provided by the algorithm bellow:

```
For i= 1 to L-1
If (Br(i+l)=0) then
    Iv= Iv ⊠ VN(i)⊗ XN(i)
Else
    Iv=Iv ⊗ VN(i) ⊠ YN(i+l)
Next i
```

4. Encryption Process.

4.1. *Matrix transition.*

- The vector (VN) of size $(1,L)$ is transformed to matrix MN of size $(t\,,r)$.
- The vector (XN) of size $(1,L)$ is transformed to matrix MX of size $(t\,,r)$.
- The vector (YN) of size $(1,L)$ is transformed to matrix MY of size $(t\,,r)$.

4.2. *Selection Vectors.*

Each row of these two dimensional array is a block of $\boldsymbol{r}$ nucleotides, which contains $\boldsymbol{q}$ sub-blocks of 10 nucleotides $[r = 10 \times q]$. Our technique uses a genetic crossing supervised by four chaotic permutations of size $(1\,,t)$ C1, C2, C3 and C4, with coefficients inZ/tZ.

- C1: indicates the index of the original block in the matrix MN, which will be crossing.
- C2: indicates the index of the chaotic block in the matrix MX, which will be crossing.
- C3: indicates the index of the row in the novel matrix MC of size $(t\,,r)$. This row is used to arrange the encrypted block after crossing.
- C4: this vector indicates the index of the row in the YN matrix. This row is used to establish a diffusion phase by linking the encrypted block with the next original block.

4.3. *Crossover Process*

After selection of the original block and the chaotic one, the crossover process is performed by using three vectors of size $(1,q)$, $Cr1$, $Cr2$ and $Cr3$, with coefficients in Z/qZ.

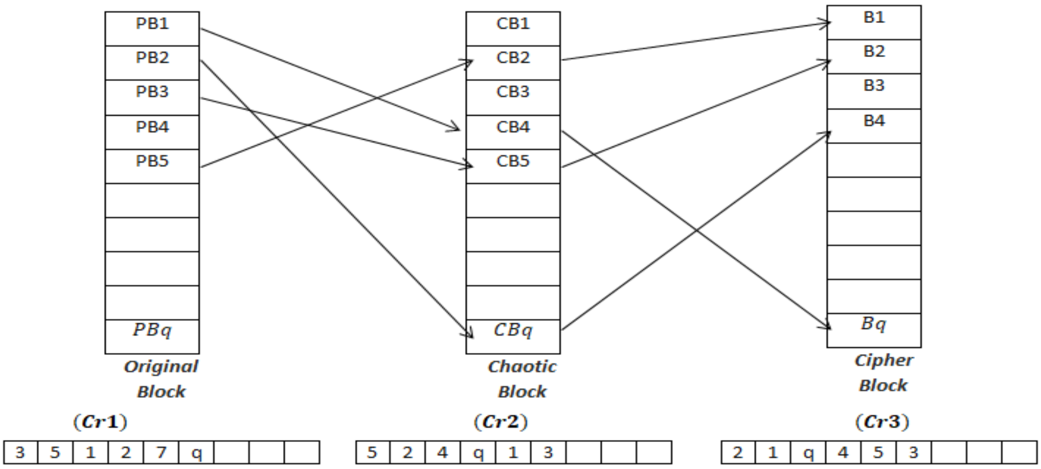


Fig.1. Crossover process

An enhanced crossover between two randomly selected blocks is ensured by the following expression:

$$\mathbf{MC(C3(i),Cr3(j)\times 10+k)= MN(C1(i),Cr1(j)\times 10+k) \boxtimes MX(C2(i),Cr2(j)\times 10+k)} \quad \mathbf{(5)}$$

Where **i=0 … t-1** , **j=0 … q-1** and **k=0 … 9**

4.4. *Diffusion Process.*

This process is established by the following formula.

$$\mathbf{MN(C1(i+1),Cr1(j)\times 10+k)= MC(C3(i),Cr3(j)\times 10+k)\otimes MN(C1(i+1),Cr1(j)\times 10+k)} \quad \mathbf{(6)}$$

Where **i=0 … t-2** , **j=0 … q-1** and **k=0 … 9**

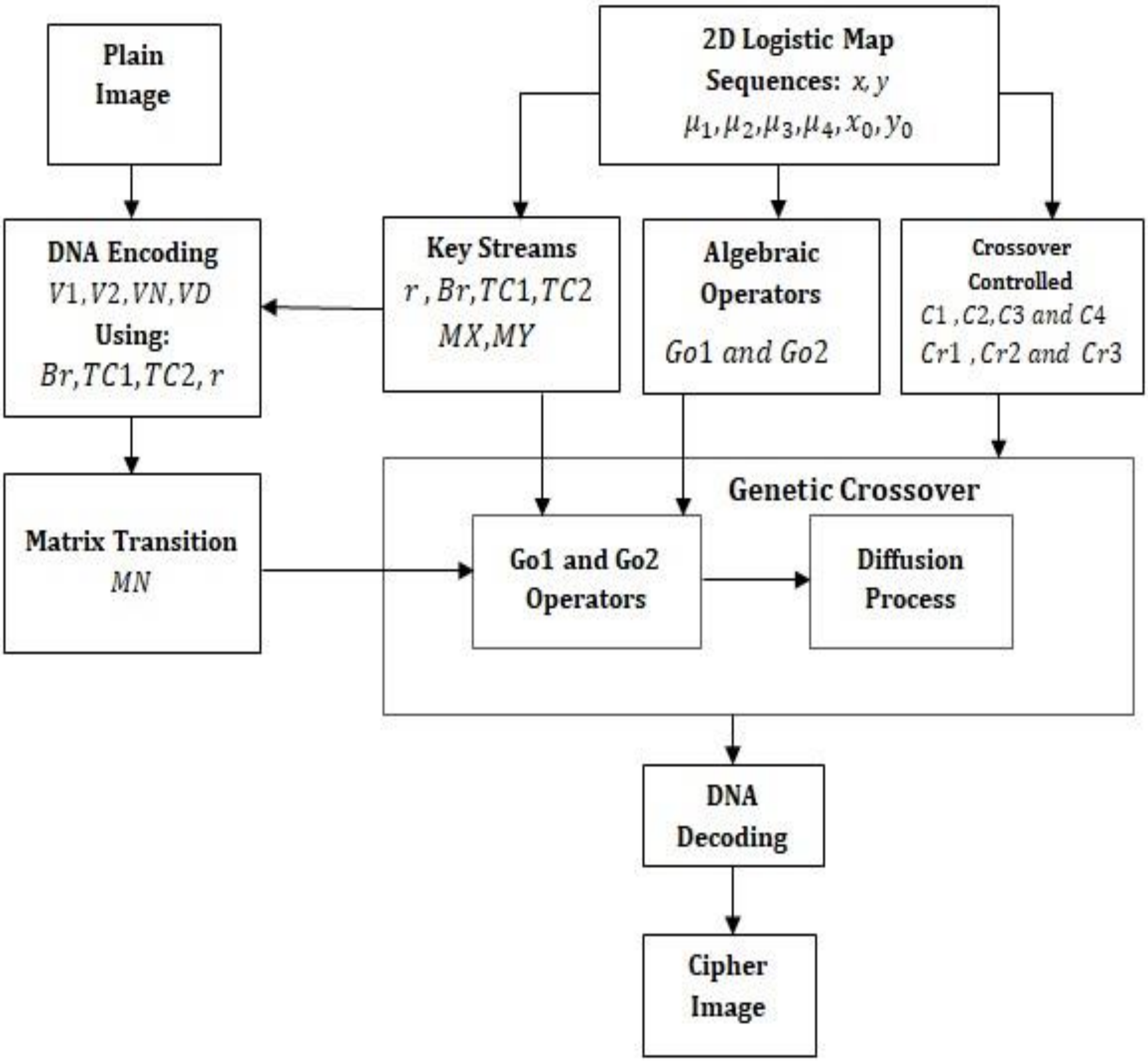
At the end of each crossing a modification of the $(Cr1)$, $(Cr2)$ and $(Cr3)$ is ensured by a chaotic displacement.

The Figure below (Fig.2) shows the encryption mechanism of our scheme.

5. *Decryption Process*.

Our algorithm is a symmetrical encryption system, therefore all encryption parameters will be used in the reverse process with reciprocal functions. The decryption process follows the following schedule.

- ✓ Encrypted image transformation.
- ✓ Recalculation of the block size (r)
- ✓ Image vector size adaptation
- ✓ Passage in nucleotide
- ✓ The reverse crossing is provided by the same controlled settings.
- ✓ Reconstruct the original image



Plain Image
2D Logistic Map Sequences: x, y
$\mu_1, \mu_2, \mu_3, \mu_4, x_0, y_0$
DNA Encoding $V1, V2, VN, VD$ Using: $Br, TC1, TC2, r$
Key Streams $r, Br, TC1, TC2$ MX, MY
Algebraic Operators $Go1$ and $Go2$
Crossover Controlled $C1, C2, C3$ and $C4$ $Cr1, Cr2$ and $Cr3$
Matrix Transition MN
Genetic Crossover
Go1 and Go2 Operators
Diffusion Process
DNA Decoding
Cipher Image

Fig.2. Encryption Process

IV. Experimental Results and Analysis.

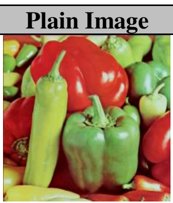
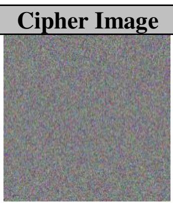
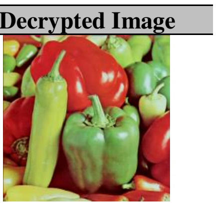
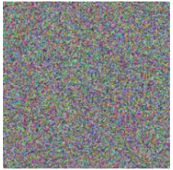
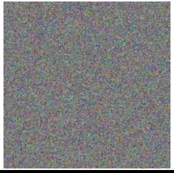
1. Simulation Result.

In order to demonstrate the performance of our image encryption system, we randomly select a large number of images at the USC-SIPI [27] and UCID [28] database to test our method. These experiments are performed using a computer with: 4 GB memory, Intel(R) Core(TM) i3-8130U CPU 2.20GHz and Java language is used as the compiling software. The secret key to our system consists of:

$$x_0 = 0{,}7655412001, \mu_1 = 3.89541$$
$$y_0 = 0.865421331, \quad \mu_2 = 0{,}56120$$
$$\mu_3 = 1{,}3561 \qquad \mu_4 = 0{,}56321$$

The obtained experimental results are shown in the table as follow:

Table.6. Experimental Results

Image Name	Image Size	Plain Image	Cipher Image	Decrypted Image
Peppers	**512×512**			
Baboon	**512×512**			
Barbara	**512×512**			
Lena	**256×256**			
Ucid0622	**384×512**			

2. *Performance and Security Analysis.*

2.1. Key Space.

The chaotic sequences used in our method ensure strong sensitivity to initial conditions, and can protect it from any brutal attacks. The key secret of our image encryption scheme consists of six real number μ_1 , μ_2, μ_3 , μ_4 ,x_0 and y_0 of single-precision. Then the total size of the key will greatly exceed≈ $\mathbf{2^{180}} \gg \mathbf{2^{128}}$ [24], which is enough to avoid any brutal attacks.

2.2. Histogram Analysis.

All images encrypted by our algorithm have a uniformly distributed histogram. This reflects that the entropy of the encrypted images is close to 8. The attackers can't extract any information about the distribution of the gray level of the cipher image, which makes the system strong against to histogram attacks. The figure 3 shows in the first column the plain image, histogram of the plain image in the second column and the histogram of the encrypted image in the third column.

2.3. Correlation analysis.

The correlation between adjacent pixels of an encrypted image is a technique used by cryptanalysts to decrypt this image. Therefore, a strong encryption system against this type of attack must reduce this correlation as much as possible. The table below gives the correlation between the adjacent pixels of several images encrypted by our algorithm in the three directions (horizontal, vertical and diagonal). The relevant expression is defined by the following equation.

$$r = \frac{cov(x,y)}{\sqrt{V(x)}\sqrt{V(y)}} \qquad (7)$$

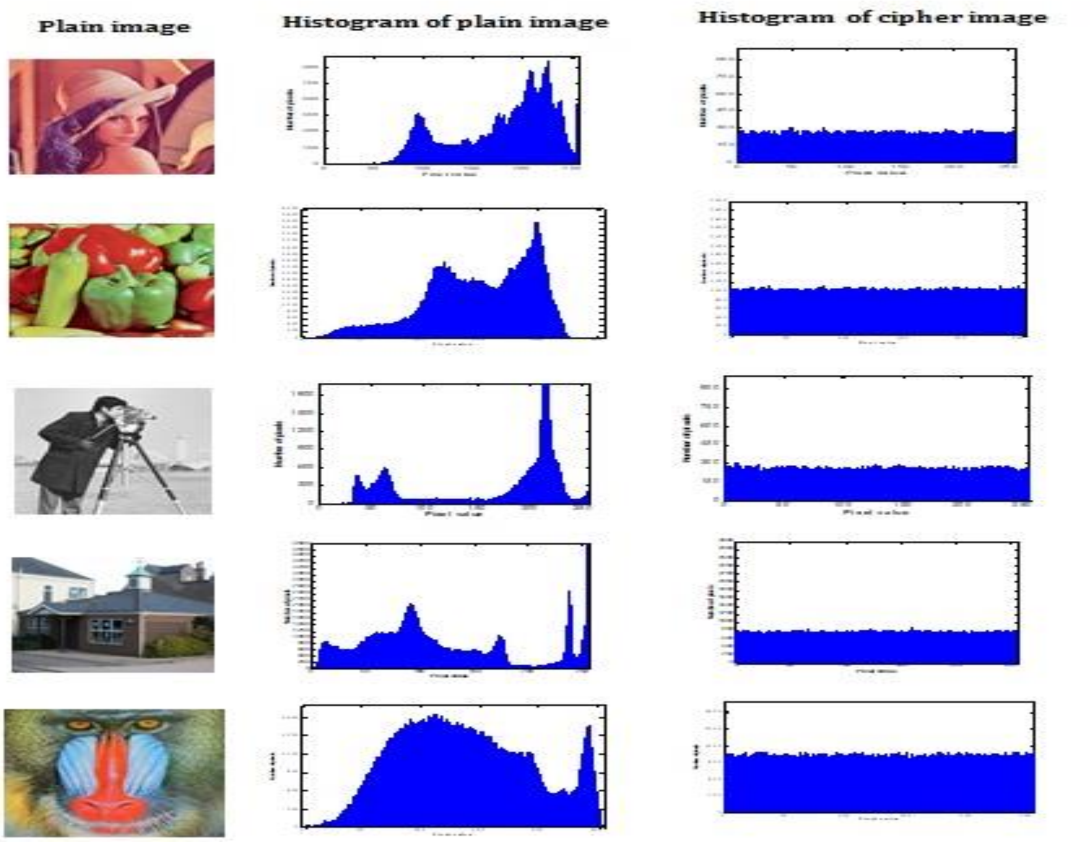


Fig.3. Histogram analysis

Table.7. Coefficient correlation analysis

Encrypted Image	Directions		
	Horizontal	**Vertical**	**Diagonal**
Lena	0.0007	0.0022	0.0041
Baboon	0.0031	0.0054	0.0031
Peppers	-0.0012	-0.0034	0.0053
Cameraman	-0.003	-0.0051	0.0032
Barbara	0.0025	0.0031	0.0012

2.4. Entropy Analysis

Entropy is the measure of the disorder diffused by a source without memory. The entropy expression is determined by the equation below.

$$H(m) = \sum_{i=0}^{255} p(m_i)\, log_2(p(m_i)) \qquad \textbf{(8)}$$

Where $p(m_i)$ is the probability of of occurrence for each m_i. For a perfect random information source with $256 = 2^8$ states, the ideal value of entropy equal 8. Table 8 gives the different entropy values obtained by our method.

2.5. Resistance to differentia attack

Sometimes the cryptanalyst make a small change in the original image to sees the effect of this small change on the cipher image. A strong encryption system against differential attacks should be very sensitive to a small change in the original image. This sensitivity analyzed by the number of pixels changed (NPCR) and the Unified Averaged Changed Intensity (UACI). The $NPCR$ and UACI mathematical analysis of an image is given by the equation below.

$$NPCR = \left(\frac{1}{nm}\sum_{i,j=1}^{nm} D(i,j)\right) * 100 \qquad \textbf{(9)}$$

$$D(i,j) = \begin{cases} 1 & if \quad C_1(i,j) \neq C_2(i,j) \\ 0 & if \quad C_1(i,j) = C_2(i,j) \end{cases} \qquad \textbf{(10)}$$

$$UACI = \left(\frac{1}{nm}\sum_{i,j=1}^{nm} Abs\big(C_1(i,j) - C_2(i,j)\big)\right) * 100 \qquad \mathbf{(11)}$$

The C1 and C2 are two encrypted images with a small change of the original image.

The Table 8 shows the values of the NPCR and UACI by applying our method on Baboon, Lena, Peppers, Cameraman and Barbara images.

Table.8. NPCR, UACI and Entropy analysis

	NPCR	UACI	Entropy
Baboon	99.654	33.454	7.9998
Peppers	99.745	33.784	7.9998
Lena	99.634	99.634	7.9992
Cameraman	99.632	33.541	7.9991
Barbara	99.625	33.461	7.9992

2.6. Signal-To-Peak Noise Ratio (PSNR)

MSE

The image quality estimation to be based on the pixel change was obtained by processing the $(PSNR)$ values and the (MSE). It is calculated by the following equation [25]

$$MSE = \sum_{i,j}(P(i,j) - C(i,j))^2 \qquad \mathbf{(12)}$$

Where $(P(i,j))$and$(C(i,j))$ represent respectively the pixel of i^{th} row and j^{th} column in the plain image and in the encrypted image.

PSNR

The transmission of digital images through communication channels can cause loss of information. So, a good encryption system must be strong against noise and information loss [22, 23]. The signal-to-peak noise ratio, often abbreviated $PSNR$, is an engineering term for the ratio between a signal's maximum possible power and the power of distorted noise that affects the precision of its display. The $PSNR$ mathematical analysis of an image is given by the next equation [26].

$$PSNR = 20Log_{10}\left(\frac{I_{max}}{\sqrt{MSE}}\right)\ dB \qquad \mathbf{(13)}$$

For (RGB) color images, the definition of $(PSNR)$ is the same except that the (MSE) is the sum of all square value changes. In the alternative, for color images, the image is transcoded into a separate color space and the $PSNR$ is displayed for each channel in that color space.

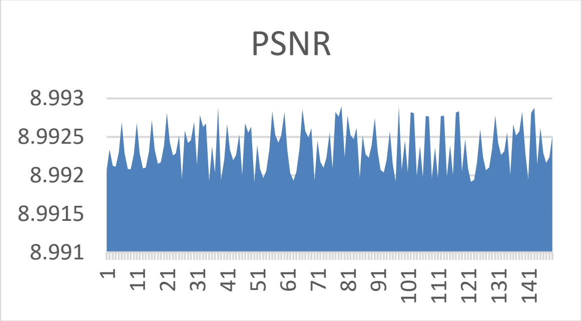


Fig.4.PSNR values of 150 images

All values returned from the $(PSNR)$ by our method are all in the range of residual values$[8.99\ ;\ 8.993]$.

2.7. Comparisons with some references Methods.

In table.9, we will present the results of the comparison of the performances of our method with several other approaches based on the results of the security analysis of each algorithm. This comparison proves the robustness of our crypto-system against all known attacks.

2.8. Discussion.

Due to the chaotic aspect of the encoding rule of the original image into DNA sequences and also to the control vectors used to select and manage the process of genetic crossover between blocks of random size, the results analysis of our algorithm show that the system proposed in this article is not a subject of all known attacks.

Table.9. Comparison of our scheme with other methods.

Measure	**Image Name**	**Our Method**	**Method [12]**	**Method [8]**	**Method [7]**
Entropy	Baboon(512×512)	7.9998	7.9944	---	7.9987
	Peppers(512×512)	7.9998	7.9973	---	7.9992
	Lena(256×256)	7.9992	7.9969	7.9967	7.9991
	Cameraman(256×256)	7.9991	7.9976	---	7.9991
NPCR	Baboon(512×512)	99.654	---	----	99.63
	Peppers(512×512)	99.745	99.63	----	99.60
	Lena(256×256)	99.634	66.63	99.61	99.57
	Cameraman(256×256)	99.632	99.54	----	99.56
UACI	Baboon(512×512)	33.454	----	----	33.40
	Peppers(512×512)	33.784	30.89	----	33.17
	Lena(256×256)	33.545	30.47	33.51	33.35
	Cameraman(256×256)	33.541	31.76	----	33.59
Vertical Correlation	Baboon(512×512)	-0.0031			----
	Peppers(512×512)	-0.0012			-0.0002
	Lena(256×256)	0.0007			0.0033
	Cameraman(256×256)	-0.0030			----

V. Conclusion

In this work, a system of encryption of color images of different sizes is proposed. On the first hand, this system based on the use of two-dimensional logistic map and a combination of novel operations on DNA sequences and an adapted and controlled genetic crossover. On the other hand, the DNA encoding rule and the size of the blocks selected for the crossing are randomly chosen. These chaotic characteristics increase the complexity of our algorithm. The results of security analysis includes, histogram, entropy, NPCR and UACI values, correlation coefficients and noise analysis prove the effectiveness of this crypto-system against all known attacks.

Compliance with Ethical Standards:

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

Funding: Not applicable

Conflicts of interest: Not applicable

Informed consent: Not applicable

References

1. Wang, X. Y., Zhang, Y. Q., & Bao, X. M. (2015). A novel chaotic image encryption scheme using DNA sequence operations. *Optics and Lasers in Engineering*, *73*, 53-61.

2. Parvaz, R., & Zarebnia, M. (2018). A combination chaotic system and application in color image encryption. *Optics & Laser Technology*, *101*, 30-41.

3. Fridrich, J. (1998). Symmetric ciphers based on two-dimensional chaotic maps. *International Journal of Bifurcation and chaos*, *8*(06), 1259-1284.

4. Farah, M. B., Farah, A., & Farah, T. (2019). An image encryption scheme based on a new hybrid chaotic map and optimized substitution box. *Nonlinear Dynamics*, 1-24.

5. Ullah, A., Jamal, S. S., & Shah, T. (2018). A novel scheme for image encryption using substitution box and chaotic system. *Nonlinear Dynamics*, *91*(1), 359-370.

6. Mahmud, M., Lee, M., & Choi, J. Y. (2020). Evolutionary-based image encryption using RNA codons truth table. *Optics & Laser Technology*, *121*, 105818.

7. Ghazvini, M., Mirzadi, M., & Parvar, N. (2020). A modified method for image encryption based on chaotic map and genetic algorithm. *Multimedia Tools and Applications*, *79*(37), 26927-26950.

8. Niu, Y., Zhou, Z., & Zhang, X. (2020). An image encryption approach based on chaotic maps and genetic operations. *Multimedia Tools and Applications*, *79*(35), 25613-25633.

9. Jithin, K. C., & Sankar, S. (2020). Colour image encryption algorithm combining Arnold map, DNA sequence operation, and a Mandelbrot set. *Journal of Information Security and Applications*, *50*, 102428.

10. Fu, X. Q., Liu, B. C., Xie, Y. Y., Li, W., & Liu, Y. (2018). Image encryption-then-transmission using DNA encryption algorithm and the double chaos. *IEEE Photonics Journal*, *10*(3), 1-15.

11. Zhang, Y. Q., He, Y., Li, P., & Wang, X. Y. (2020). A new color image encryption scheme based on 2DNLCML system and genetic operations. *Optics and Lasers in Engineering*, *128*, 106040.

12. Noshadian, S., Ebrahimzade, A., & Kazemitabar, S. J. (2020). Breaking a chaotic image encryption algorithm. *Multimedia Tools and Applications*, *79*(35), 25635-25655.

13. Mohamed, H. G., ElKamchouchi, D. H., & Moussa, K. H. (2020). A novel color image encryption algorithm based on hyperchaotic maps and mitochondrial DNA sequences. *Entropy*, *22*(2), 158.

14. Gan, Z. H., Chai, X. L., Han, D. J., & Chen, Y. R. (2019). A chaotic image encryption algorithm based on 3-D bit-plane permutation. *Neural Computing and Applications*, *31*(11), 7111-7130.

15. Khan, J. S., & Ahmad, J. (2019). Chaos based efficient selective image encryption. *Multidimensional Systems and Signal Processing*, *30*(2), 943-961.

16. Liu, H., & Wang, X. (2010). Color image encryption based on one-time keys and robust chaotic maps. *Computers & Mathematics with Applications*, *59*(10), 3320-3327.

17. Liu, L., Zhang, Q., & Wei, X. (2012). A RGB image encryption algorithm based on DNA encoding and chaos map. *Computers & Electrical Engineering*, *38*(5), 1240-1248.

18. Wei, Z. (2011). Dynamical behaviors of a chaotic system with no equilibria. *Physics Letters A*, *376*(2), 102-108.

19. Özkaynak, F., & Yavuz, S. (2014). Analysis and improvement of a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. *Nonlinear Dynamics*, *78*(2), 1311-1320.

20. Enayatifar, R., Abdullah, A. H., Isnin, I. F., Altameem, A., & Lee, M. (2017). Image encryption using a synchronous permutation-diffusion technique. *Optics and Lasers in Engineering*, *90*, 146-154.

21. Li, C., Luo, G., Qin, K., & Li, C. (2017). An image encryption scheme based on chaotic tent map. *Nonlinear Dynamics*, *87*(1), 127-133.

22. Hua, Z., Jin, F., Xu, B., & Huang, H. (2018). 2D Logistic-Sine-coupling map for image encryption. *Signal Processing*, *149*, 148-161.

23. Niyat, A. Y., Moattar, M. H., & Torshiz, M. N. (2017). Color image encryption based on hybrid hyper-chaotic system and cellular automata. *Optics and Lasers in Engineering*, *90*, 225-237.

24. Zheng, J., & Liu, L. (2020). Novel image encryption by combining dynamic DNA sequence encryption and the improved 2D logistic sine map. *IET Image Processing*, *14*(11), 2310-2320.

25. Boubaker, O., & Jafary, S. (2019). Recent advances in chaotic systems and synchronization.

26. Wang, J., & Li, X. (2015). A chaotic system with one line equilibria and image encryption with avalanche effects. In *Proceedings of the 2015 International Conference on Electronics, Electrical Engineering and Information Science-EEEIS*.

27. http://sipi.usc.edu/database/database.php?volume=misc.

28. Schaefer G , Stich M . UCID - an uncompressed color image database. In: storage and retrieval methods and applications for multimedia 2004, Proceedings of SPIE, 5307; 2004. p. 472–80.