Journal of Multimedia Information System VOL. 7, NO. 2, June 2020 (pp. 165-174): ISSN 2383-7632 (Online)
http://doi.org/10.33851/JMIS.2020.7.2.165

Improved Single Feistel Circuit Supporter by A Chaotic Genetic Operator

Abdellatif JarJar [1*]

Abstract

This document outlines a new color image encryption technology development. After splitting the original image into 240-bit blocks and modifying the first block by an initialization vector, an improved Feistel circuit is applied, sponsored by a genetic crossover operator and then strong chaining between the encrypted block and the next clear block is attached to set up the confusion-diffusion and heighten the avalanche effect, which protects the system from any known attack. Simulations carried out on a large database of color images of different sizes and formats prove the robustness of such a system.

Key Words: Chaotic map, Feistel-round diagram, genetic operator, S-Box, P-Box.

I. INTRODUCTION

With the rapid advance of innovative technologies in the field of information science and the digital world, all data are increasingly shared over Internet networks. On the other hand, unauthorized access to information or private information has become a problem in the virtual world. Security issues are increasingly coming to the forefront. Encryption and tattooing have become the most effective way to protect against unexpected attacks. However, traditional encryption standards, such as DES and AES, are generally designed only for encrypting text that does not have a high correlation, and are therefore considered unsuitable for images and video data sequences. The new vision of image-based encryption is the use of chaotic sequences for encryption key generation first prescribed by Friedrich in 1998. Since the image has been introduced and processed in digital form, its applications have been steadily increasing. It is now exploited by a wide public, both professional and amateur. However, given the extent of computer resources allowing the free circulation of information and the ease of transmission of confidential data, man has been pushed to increasingly improve encryption algorithms to secure his confidential data. To protect against known attacks, any new encryption system must agree to Shannon's recommendations [1]; (Permutation, confusion diffusion). The majority of techniques use static permutations such as Arnold's technique [2] or advanced Hill's technique [3]. For confusion the Xor operator is the most used [4]. Recently, in order to avoid differential attacks, most algorithms use different encryption methods. Given the advances in mathematical theory, for the generation of encryption keys, all methods use chaotic cards.

Chang'e Dong [5] offers color image encryption based on the construction of a coupled chaotic map. Xing-Yuan Wanga Sheng-Xian Gua Ying-Qian Zhangab [6] proposed a crypto system based on a multitude of chaotic maps that define an effective result. All these approaches use a Lyapunov exponent calculation [7] to check the installation of chaos and sensitivity to initial conditions. Most encryption algorithms operating on blocks used the Feistel scheme with several turns. RC4, RC6, DES used more than four towers [10]. The classical Feistel technique consists in separating a2n-bit block into two blocks of n bit each, this classical method is resumed by the scheme of the figure below.

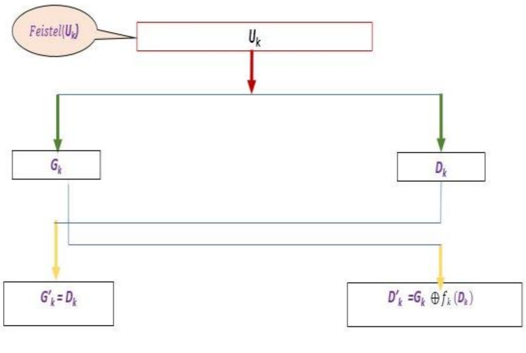


Fig. 1. Feistel's classic round scheme.

Manuscript received Apr 11, 2020; Revised Apr 23, 2020; Accepted May 16, 2020. (ID No. JMIS-20M-04-013)

Corresponding Author (*): High School Moulay Rachid, Taza, Morroco, abdoujjar@gmail.com.
[1] High School Moulay Rachid, Taza, Morroco, abdoujjar@gmail.com

This figure could be understood by the following evaluation function. Let t denote the quantity of blocks to be encrypted.

$$Equation\ 1\begin{cases} g_i(G_{i,}\ ,D_i) = (G'_i,\ D'_i) \\ \begin{cases} G'_i = D_i \\ D'_i = G_i \oplus f_i(D_i) \end{cases} \end{cases} \quad With\ i \in [\![1\quad t]\!]$$

$(f_i)$is a n-bit pseudo-random function.

In the absence of the diffusion, this method stays exposed to differential attacks. As a result, this scheme was expanded to include a new scheme by a bijection construction from purely random functions to produce a new encryption scheme [8], encapsulating confusion-diffusion. Genetic algorithms are based on the Darwinian evolution of biological populations, whose strongest individuals are the most suitable to survive and reproduce very powerful progeny. These algorithms have surfaced as pre-selected evaluation optimization tools for an assessment function. They are based on the following genetic operations: The inversion, the crossover the mutation and the insertion. Several tentative implementations for genetic algorithms for encrypting color images have surfaced [9-10]. Some use DNA sequences [11] for image encryption, others have used these genetic algorithms to upgrade some conventional encryption systems [12].

II. THE PROPOSED METHOD

Based on chaos, this technique implements one enhanced Feistel lap followed by a genetic crossover. This new color image encryption scheme focuses on six main axes All these measures are shown in a schematic diagram in the following figure.

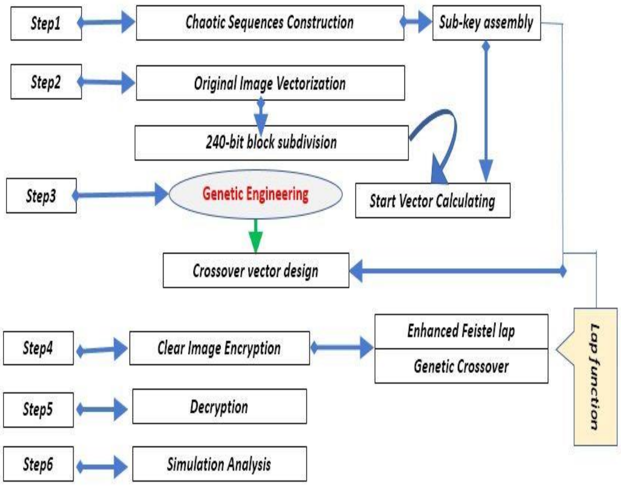


Fig. 2. Steps of realization of the algorithm.

2.1. Chaotic Sequences Development

All the encryption parameters necessary for the successful execution of our system are generated from three chaotic maps, the most frequently used in the color image encryption. This choice is due to the simplicity of their exploitation and configuration, as well as their extreme sensitivity to the initial parameters.

2.1.1. The logistics map

The logistic map is a recurrent sequence described by a simple polynomial of second degree defined by the following equation

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

This equation map guarantees that chaos is established to

$$\boldsymbol{u_0} \in \,]\mathbf{0,5}\ \ \mathbf{1}[\ \boldsymbol{as\ initial\ conditions}\quad \boldsymbol{and}\quad \boldsymbol{\mu} \in [\mathbf{3,76}\quad \mathbf{4}]\ \boldsymbol{as\ control\ parameters}$$

2.1.2. PWLCM map

It is a real linear sequence by pieces defined by the equation below.

$$\{\boldsymbol{w_n} = \boldsymbol{f}(\boldsymbol{w_{n-1}}) = \begin{cases} \frac{\boldsymbol{w_{n-1}}}{\boldsymbol{d}} & \boldsymbol{if}\ \ \mathbf{0} \le \boldsymbol{w_{n-1}} \le \boldsymbol{d} \\ \frac{\boldsymbol{w_{n-1}-d}}{\mathbf{0.5}-\boldsymbol{d}} & \boldsymbol{if\ d} \le \boldsymbol{w_{n-1}} \le \mathbf{0.5} \\ \boldsymbol{f}(\mathbf{1} - \boldsymbol{w_{n-1}}) & \boldsymbol{else} \end{cases} \qquad (2)$$

It is a very simple map to use in color image cryptography. It presents a chaotic aspect for $\mathbf{d} \in [\mathbf{0.5}\quad \mathbf{1}]$ as control parameters, and $\mathbf{w_0} \in \,]\mathbf{0}\quad \mathbf{1}[$ as initial conditions.

2.1.3. The skew tent map (SKTM)

The Skew tent map will be redefined as the next equation

$$\begin{cases} \boldsymbol{v_0} \in \,]\mathbf{0}\ \ \mathbf{1}[\qquad \boldsymbol{p} \in \,]\mathbf{0,5}\ \ \mathbf{1}[\\ \boldsymbol{v_{n+1}} = \begin{cases} \frac{\boldsymbol{v_n}}{\boldsymbol{p}} & \boldsymbol{if}\ \ \mathbf{0} \prec \boldsymbol{v_n} \prec \boldsymbol{p} \\ \frac{\mathbf{1}-\boldsymbol{v_n}}{\mathbf{1}-\boldsymbol{p}} & \boldsymbol{if\ p} \prec \boldsymbol{v_{n\prec 1}} \end{cases} \end{cases} \qquad (3)$$

The Skew tent map assures the installation of chaos under the conditions:

$$\boldsymbol{v_0} \in \,]\mathbf{0}\ \ \mathbf{1}[\ \boldsymbol{as\ initial\ conditions,}\ \ \boldsymbol{and}\ \ \boldsymbol{p} \in \,]\mathbf{0,5}\ \ \mathbf{1}[\ \boldsymbol{as\ control\ parameters}$$

2.2. Clear Image Preparation

Before transferred the original image to the encryption surgery center, it must be prepared in anticipation, for this it must include the following activities.

2.2.1. Original image vectoring

After the three (RGB) color channels extraction and their conversion into size vectors $(\mathbf{Vr}), (\mathbf{Vg}), (\mathbf{Vb})$ $(\mathbf{1}, \mathbf{nm})$ each, a concatenation is established to generate a vector $\mathbf{X}(\mathbf{x_1}, \mathbf{x_2}, \dots\dots\dots, \mathbf{x_{3nm}})$ of size $(\mathbf{1}, \mathbf{3nm})$

2.2.2. Vector size $(\mathbf{X})$ adaptation

The resulting vector $\mathbf{X}(\mathbf{x_1}, \mathbf{x_2}, \dots\dots\dots, \mathbf{x_{3nm}})$ must be divided into $\mathbf{240 - bit\ blocks\ -\ 30\ pixels}$, therefore its size must be accommodated. Let $(\mathbf{l})$ the new size calculated from the algorithm below.

```
Algorithm1 { let   $3nm \equiv r\ [30]$
             if $r = 0$ then
                 $l = 3nm$
             else
             $l = 3nm + 30 - r$
```

After, the vector $(\mathbf{X})$ will be transformed into an $(\mathbf{TX})$ vector of size $(\mathbf{1}, \mathbf{l})$ by $\mathbf{adding\ 30 - r}$ new chaotic components at the end of the vector $(\mathbf{TX})$, by applying the below algorithm.

```
Algorithm 2 {     if $r = 0$ then
                  $(TX) = (X)$
                  Else
                  for $i = 3nm + 1$ to $l$
             $TX(i) = mod(E(10^{10}u(i)), 253)) + 1$
                  end if
                  Next $i$
```

2.2.3. 240-bit blocks decomposition

The vector $(\mathbf{TX})$ is converted to binary and then a size $(\mathrm{t}, 240)$ binary matrix $(\mathbf{MC})$ with $(\mathbf{t} = \mathbf{l}/\mathbf{30})$. This decomposition can be illustrated by the following figure.

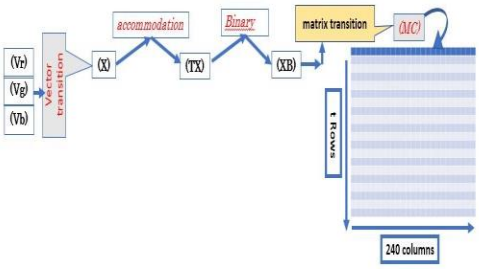


Fig. 3. Transition from clear image to matrix (MC).

2.2.4. $(\mathbf{IV})$ Initialization vector design

Ultimately, the $(\mathbf{IV})$ initialization vector of size $(\mathbf{1}, \mathbf{240})$ is provided by the next algorithm.

```
Algorithm3 {    for $i = 1$ to $240$
                  $IV(i) = 0$
                 for $k = 2$ to $t$
             $IV(i) = IV(i) \oplus MC(k, i)$
                  Next $k, i$
```

To surpass the uniform image problem (Black, White …) the vector $(\mathbf{IV})$ will be combined with the chaotic vector $(\mathbf{HT})$ specified by the following algorithm.

```
Algorithm4 {     for $i = 1$ to $2l$
               if $(u(i) > v(i))$then
                  $HT(i) = 1$
                    else
                  $HT(i) = 0$
                   Next $i$
```

The blending of the two vectors is performed by the next algorithm.

```
Algorithm5 {   for $i = 1$ to $240$
             $IV(i) = IV(i) \oplus HT(i)$
                 Next , $i$
```

This vector has the mission to only modify the value of the first block and start the diffusion confusion process.

```
Algorithm6 {       for $i = 1$ to $240$
             $MC(1, i) = MC(1, i) \oplus IV(i)$
                   Next , $i$
```

$(\mathbf{NB})$: In the absence of such an initialization vector, it is very difficult to follow the encryption scheme correctly.

2.3. Encryption Parameter Setting Architecture

2.3.1. Feistel's first round functions construction

Each MC(k:) block of order (k) will be subdivided into four identical 60 – bit blocks [MG(k:)GM(k:)MD(K:)DM(k:)] and projected to a first enhanced feistel loop described by the following figure.

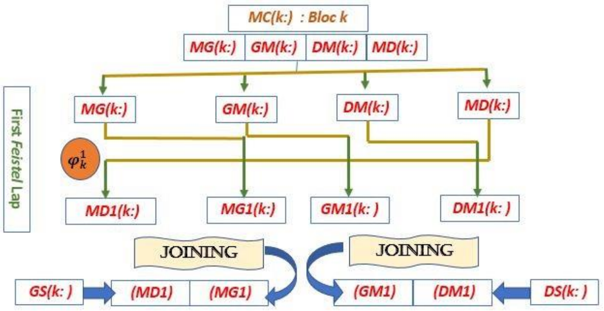


Fig. 4. First improved Feistel lap.

This pattern can be analytically expressed by the following mathematical statement.

$$\textbf{Equation 4}\ \ \varphi_k^1 \left\{\begin{array}{c} MD1(k:) = h_k(MD(k:)) \\ MG1(k:) = MG(k:) \oplus g_k(MD(k:)) \\ GS(k:) = MD1(k:)\ MG1(k:) \\ GM1(k:) = GM(k:) \oplus f_k(MG(k:)) \\ DM1(k:) = DM(k:) \oplus GM(k:) \\ DS(k:) = DM1(k:)\ GM1(k:) \end{array}\right.$$

- h_k : Chaotic permutation
- g_k : Chaotic displacement
- f_k : Chaotic confusion

2.3.1.1. Feistel function lap design

2.3.1.1.1. Permutation scheme $(\mathbf{h_k})$ *building*

Initially, a descending sort on the first 60 values of the logistics sequence generates a permutation $(\mathbf{PR})$ in $\mathbf{G_{60}}$. secondly, a chaotic vector $(\mathbf{DP})$ is constructed in parallel to serve as building the permutation matrix $(\mathbf{MP})$.

$$\textbf{Algorithm 7}\ \left\{\begin{array}{c} for\ i = 1\ \ to\ \ t \\ DP(i) = mod(E(10^{10}u(i)), 52)) + 3 \\ Next\ \ i \end{array}\right.$$

The first line of the permutation matrix $(\mathbf{MP})$ is the permutation $(\mathbf{PR})$; while line $(\mathbf{i \geq 2})$ is the displacement of line $(\mathbf{i-1})$ of step $\mathbf{DP(i)}$. This construction is described by the following algorithm.

$$\textbf{Algorithm8} \left\{\begin{array}{ll} \textit{The first line} & \left\{\begin{array}{c} for\ i = 1\ to\ 60 \\ MP(1,i) = PR(i) \\ Next\ i \end{array}\right. \\ \textit{The following lines} & \left\{\begin{array}{c} For\ i = 2\ to\ t \\ for\ j = 1\ to\ 60 \\ MP(i,j) = MP\begin{pmatrix} i-1, mod \\ (j + DP(i); 60) \end{pmatrix} \\ Next\ j, i \end{array}\right. \end{array}\right.$$

Therefore, the application of the permutation $\mathbf{h_k}$ on the block $(\mathbf{MD(k:)})$ given by the next algorithm.

$$\textbf{Algorithm9} \left\{\begin{array}{c} for\ \ i = 1\ to\ 60 \\ MD1(k,i) = MD(k, MP(k,i)) \\ Next\ , i \end{array}\right.$$

2.3.1.2. Function scheme g_k

$\mathbf{g_k}$: is a chaotic offset applied to the 60 bits of the $\mathbf{MG(k:)}$ block. This offset is performed by the chaotic vector $(\mathbf{DD})$ resulting by applying the next algorithm.

$$\textbf{Algorithm 10} \left\{\begin{array}{c} for\ i = 1\ \ to\ t \\ DD(i) = mod\left(E(10^{10}\frac{u(i) + v(i) + w(i)}{3}), 51\right) + 4 \\ Next\ \ i \end{array}\right.$$

The analytical expression of such a displacement is illustrated by the following algorithm.

$$\textbf{Algorithm11} \left\{\begin{array}{c} for\ \ i = 1\ to\ 60 \\ MG1(k,i) = MG(k, mod\ (DD(k) + i, 60) \\ Next\ , i \end{array}\right.$$

2.3.1.3. Function scheme f_k

The matrix $(\mathbf{MS})$ is the passage of the vector $(\mathbf{HT})$ resulting by applying the algorithm4 in matrix of size $(\mathbf{t, 60})$. As a result, the confusion function is given by the following algorithm.

$$\textbf{Algorithm12} \left\{\begin{array}{c} for\ \ i = 1\ to\ 60 \\ GM1(k,i) = MG(k,i) \oplus MS(k,i) \\ Next\ , i \end{array}\right.$$

2.3.2. Analytical expression of the function φ_k^1

The transformer of first-round diagram block $(\mathbf{MC(k:)})$ given by the algorithm below.

$$\textbf{Algorithm 13}\qquad \varphi_k^1 \left\{\begin{array}{c} for\ \ i = 1\ \ to\ \ 60 \\ MD1(k,i) = MD(k, MP(k,i)) \\ Next\ i \\ for\ \ i = 1\ \ to\ \ 60 \\ MG1(k,i) = GM(k,i) \oplus MG(k,i) \oplus MS(k,i) \\ Next\ i \\ for\ i = 1\ to\ 60 \\ GM1(k:i) = GM(k,i) \oplus MG(k, mod(i + DD(k), 60) \\ Next\ i \\ for\ i = 1\ to\ 60 \\ DM1(k:i) = DM(k,i) \oplus GM(k,i) \\ Next\ i \end{array}\right.$$

We affirm that the function for the first round is a bijection, its reciprocal is given by the following equation.

$$\textbf{Equation 5}\qquad \varphi_k^{-1} \left\{\begin{array}{c} \mathbf{MD(k:) = h_k^{-1}(MD1(k:))} \\ \mathbf{MG(k:) = MG1(k:) \oplus g_k\left(h_k^{-1}(MD1(k:))\right)} \\ \mathbf{GM(k:) = GM1(k:) \oplus f_k\left(MG1(k:) \oplus g_k\left(h_k^{-1}(MD1(k:))\right)\right)} \\ \mathbf{DM(k:) = DM1(k:) \oplus GM1(k:) \oplus f_k\left(MG1(k:) \oplus g_k\left(h_k^{-1}(MD1(k:))\right)\right)} \end{array}\right.$$

2.3.3. Crossover matrix design

A genetic crossover is a pseudo-random function applied to two genes of the same size to form another gene of double size. In our approach, it is a chaotic crossing between two 120-bit vectors to generate a 240-bit block. Firstly, for each block (k) two chaotic vectors $(\mathbf{HR})$ and $(\mathbf{RH})$ are generated by the following algorithm.

$$\textbf{Algorithm14} \left\{\begin{array}{c} for\ i = 1\ to\ 11 \\ HR(i) = mod\left(E\left(10^{10} * inf\left(v(i+k), u(i+k)\right)\right); 9\right) + 4 \\ RH(i) = mod\left(E\left(10^{12} * \left(\frac{u(i+2k) + 3 * v(i+3k)}{4}\right)\right); 9\right) + 3 \\ Next\ i \end{array}\right.$$

So

$$\textbf{Equation6} \left\{\begin{array}{l} HR = (h_1, h_2, \dots \dots . . h_{12})\ \ With\ \ h_{12} = 120 - \sum_{i=1}^{i=11} h_i \\ RH = (r_1, r_2, \dots \dots . . r_{12})\ \ With\ \ r_{12} = 120 - \sum_{i=1}^{i=11} r_i \end{array}\right.$$

The crossing function in our system is defined by the following mathematical formula.

$$\textbf{Equation7} \left\{\begin{array}{c} Cr(L,R) = Q \\ L\ size\ block\ (1, 120)\ bit \\ R\ size\ block\ (1, 120) bit \\ Q\ size\ block\ (1, 240)\ bit \end{array}\right.$$

The vector $(\mathbf{Q})$ obtained by the following formula.

$$\begin{cases} Q = (Q_1)(Q_2)(Q_3)(Q_4) \dots\dots\dots.(Q_{24}) \\ \text{With} \\ \textbf{Equation8} \begin{cases} (Q_1) = \text{Contains } h_1 \text{ components extracted from } L \text{ from the first position} \\ (Q_2) = \text{Contains } r_1 \text{ components extracted from } R \text{ from the first position} \\ (Q_3) = \text{Contains } h_2 \text{ components extracted from } L \text{ from the } h_1 \text{ position} \\ (Q_4) = \text{Contains } r_2 \text{ components extracted from } R \text{ from the } r_1 \text{ position} \\ \vdots \\ (Q_{24}) = \text{Contains } r_{12} \text{ components extracted from } R \text{ from the } r_{11} \text{ position} \end{cases} \end{cases}$$

Example:

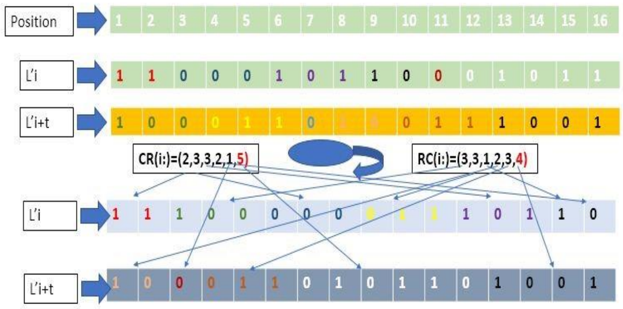


Fig. 5. Crossing of two individuals.

The genetic crossing function $(\mathbf{Cr})$ defined by the equation7 is a bijection. Indeed, we have the block.

$$\boldsymbol{Q = (Q_1)(Q_2)(Q_3)(Q_4) \dots\dots\dots.(Q_{24})}$$

$$\textbf{Equation 9} \quad \boldsymbol{Cr^{-1}} \begin{cases} \boldsymbol{L = (Q_1)(Q_3)(Q_5) \dots\dots\dots.(Q_{23})} \\ \boldsymbol{R = (Q_2)(Q_3)(Q_6) \dots\dots\dots.(Q_{24})} \end{cases}$$

2.4. Clear Image Encryption

Let's assume $(\emptyset)$ the clear image $(\mathbf{MC})$ encryption function, we have

$$\textbf{Equation 10} \quad (\boldsymbol{MO}) = \emptyset(\boldsymbol{MC}) \ \textbf{With} \ \emptyset = \boldsymbol{\varphi^2 \ o \ Mt \ o \ \varphi^1}$$

Therefore

$$\textbf{Equation 11} \quad \emptyset(\boldsymbol{MC}(\boldsymbol{k}:)) = \begin{cases} \forall \boldsymbol{k} \in [\![\mathbf{1} \ \boldsymbol{t}]\!] \ \textbf{We have} \\ (\boldsymbol{MO}(\boldsymbol{k}:)) = \emptyset(\boldsymbol{MC}(\boldsymbol{k}:)) \\ (\boldsymbol{MO}(\boldsymbol{k}:)) = \boldsymbol{Cr_k \ o \ \varphi_k^1}(\boldsymbol{MC}(\boldsymbol{k}:)) \end{cases}$$

The encryption process can be illustrated by the following diagram.

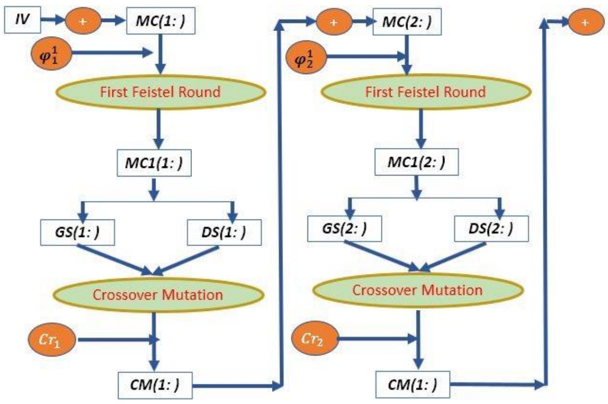


Fig. 6. Color image encryption.

2.4.1. Cryptographic function mathematical expression

This new color image encryption technique follows the steps of the algorithm below.

1) Clear image vectoring
2) Adapt the size of image vector
3) Split into t 240 – bit block
4) Extract the initialization vector
5) Do k=1
6) Confusion with the first block
7) Applying the rotation function $\boldsymbol{\varphi_k^1}$ the block
8) Perform a genetic crossover on the two output blocks to get the block (IS)
9) Do (IV) = (IS)
10) Do $\mathrm{k = k + 1}$
11) If $\mathrm{k \le t}$ then do $\mathrm{MC(k-1:) = MC(k:)}$
12) Return to 5
13) If $\mathrm{k > t}$ then restore the encrypted image

2.5. Encrypted Image Decryption

Our approach is a symmetrical chaos-based encryption system, so the secret encryption key is also the decryption key. After decomposing the encrypted image into 240-bit blocks and regenerating all encryption parameters, the decryption process starts with the last block by applying the inverse turn function to the second block, then the initialization vector is recalculated to retrieve the first block and restore the original image.

$$\textbf{Equation 13} \quad \emptyset^{-1}(\boldsymbol{MO}(\boldsymbol{k}:)) = \begin{cases} \forall \boldsymbol{k} \in [\![\boldsymbol{t} \ \mathbf{1}]\!] \ \textbf{We have} \\ (\boldsymbol{MC}(\boldsymbol{k}:)) = \emptyset^{-1}(\boldsymbol{MO}(\boldsymbol{k}:)) \\ (\boldsymbol{MO}(\boldsymbol{k}:)) = \boldsymbol{\varphi_k^{-1} o \ Cr_k^{-1}}(\boldsymbol{MO}(\boldsymbol{k}:)) \end{cases}$$

2.6. Example and Simulations

A good system crypto must face all known attacks. For each statistical constant, 150 images are randomly selected according to a chaotic vector from a database of color images of different sizes and formats are tested by our algorithm, and a detailed statistical study has been developed.

2.6.1. Key space

If the precision of the computing is 10 decimal digits, then the size of the encryption key in our approach is $10^{60} \approx 2^{180} \gg 2^{100}$ which is more than enough to protect our method from brutal attacks.

2.6.2. Secret key's sensitivity analysis

The high sensitivity of the encryption keys used in our system indicates that a very slight degradation of the encryption key automatically leads to an image that is so different from the original image. This confirmation can be viewed below the scheme in the figure12.

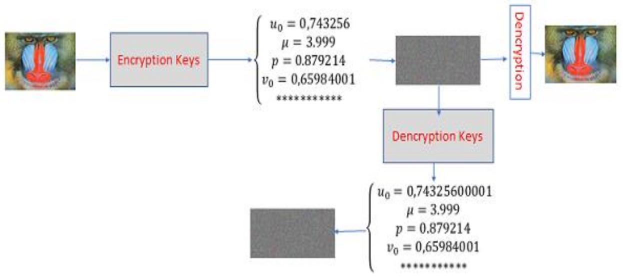


Fig. 7. Secret key's sensitivity

We note that a 10^{-12} change in a single encryption parameter of this technology is incapable of restoring the clear image by the same decryption process.

2.6.3. Entropy analysis

Entropy is the measure of the disorder diffused by a source without memory. The entropy is therefore maximal for a source whose symbols are all equiproable or presenting a flat histogram. The entropy is for an $(\mathbf{MC})$ image of size $(\mathbf{n}, \mathbf{m})$, we pose $(\mathbf{t} = \mathbf{nm})$, So

$$\textbf{\textit{Equation 14}} \qquad H(MC) = \frac{1}{t}\sum_{i=1}^{t} -p(i)\,log_2(p(i))$$

The entropy values on the 150 images tested by our method are represented graphically by the following figure.

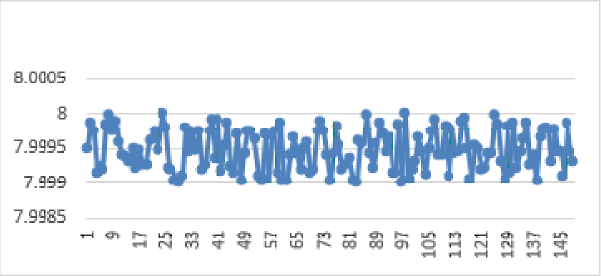


Fig. 8. Entropy of 150 images of the same size

The entropy values of the images encrypted by our algorithm are around 8, it is the maximum value for a color image encoded on 8 bits. It confirms the uniformity of the histograms. This proves that this approach is safe from entropy attack.

2.6.3.1. Entropy statistical analysis

2.6.3.1.1. Position parameter analysis

The values derived from the entropy by applying our approach to over 150 images in our image database, constitute a statistical series with position, dispersion and concentration parameters have been recalculated to verify the safety of our approach.

$$\textbf{\textit{Equation 15}} \begin{cases} Q_1 = & \textit{First quartile} \\ Q_2 = & \textit{Second quartile} \\ Q_3 = & \textit{Third quartile} \end{cases}$$

Average	Max	Min	Q1	Q2	Q3
7,999480662	7,999993872	7,999004283	7,99921863	7,99946043	7,999737

Table. 1. Position Parameter

The moustache box of the entorpy is illustrated in the diagram in Figure below.

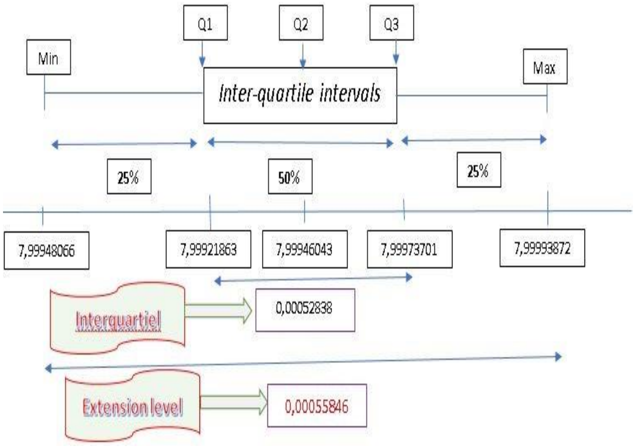


Fig. 9. Entropy moustache box.

2.6.3.1.2. Asymmetry coefficient

The Yule coefficient measures the asymmetry of the frequency curve of a statistical series. It is explained by the next equation.

$$\textbf{\textit{Equation 16}} \quad s = \frac{(Q_3 - Q_1) - (Q_2 - Q_1)}{(Q_3 - Q_1)} = \frac{Q_3 - 2Q_2 + Q_1}{(Q_3 - Q_1)}$$

Under these conditions, Yule has demonstrated that

$$\textbf{\textit{Equation 17}} \begin{cases} s = 0 & \textit{it has symmetry}. \\ s > 0 & \textit{right spreading} \\ s < 0 & \textit{left spreading} \end{cases}$$

In our entropy study, we found

$$s = \frac{Q_3 - 2Q_2 + Q_1}{(Q_3 - Q_1)} = 0{,}00221$$

We note that $s \approx 0$.

We can say that the frequency curve is symmetrical.

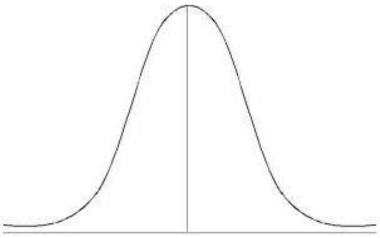


Fig. 10. Example of a symmetrical curve following a normal distribution $(\boldsymbol{s} \approx \boldsymbol{0})$.

2.6.3.1.3. Applatissment coefficient

Flattening is judged by reference to the normal distribution density curve model. We will say that the frequency curve is more or less flattened than the normal distribution model.

The coefficient for quantitatively measuring flattening is called the (Kurtosis). Pearson proposed the following coefficient:

$$Equation\ 18 \quad \beta_2 = \frac{\mu_4}{\sigma^4} \quad With \begin{cases} \mu_4 \text{ is the fourth order moment} \\ \sigma \text{ is the standard deviation} \end{cases}$$

Under these conditions, Pearson has demonstrated that

$$\begin{cases} \boldsymbol{\beta_2 = 3} & \textbf{Normal law.} \\ \boldsymbol{\beta_2 > 3} & \textbf{Flattened curve} \\ \boldsymbol{\beta_2 < 3} & \textbf{Sharp curve} \end{cases} \qquad (19)$$

In our entropy study, we found

$$\boldsymbol{\beta_2 = \frac{\mu_4}{\sigma^4} = 2,99725}$$

We note that $\boldsymbol{\beta_2 \approx 3}$.

We can say that our distribution is a normal distribution.

2.6.3.2. Correlation analysis

Correlation is a technique that compares two images to estimate the displacement of pixels in one image relative to another reference image. Adjacent pixels of a standard image of a clear image have a strong correlation. A good crypto image system must remove such correlation in order to avoid any statistical attack. The correlation expression is defined by equation below.

$$correlation \qquad r = \frac{cov(x,y)}{\sqrt{V(x)}\sqrt{V(y)}} \qquad (20)$$

2.6.3.2.1. Horizontal correlation

Simulations performed on 100 identical-sized color images choose from a wide database of images of various sizes, formats and correlated values are represented graphically by the next figure.

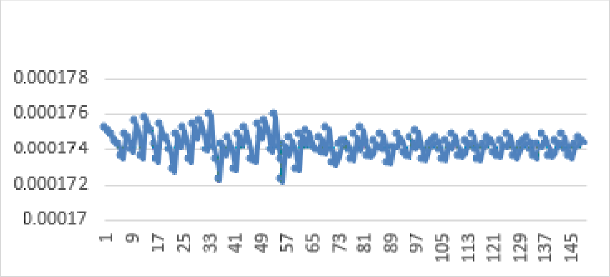


Fig. 11. Entropy of 100 images of the same size.

2.6.3.2.2. Vertical correlation

Simulations made on 150 images of the database gave the vertical correlation scores are displayed in Figure below.

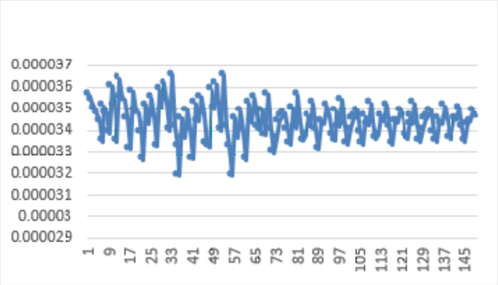


Fig. 12. Vertical correlation of 70 images of the varying sizes.

Figure 12 shows that the vertical correlation values of the encrypted images are close to zero. This ensures high security against correlation attacks.

2.6.3.2.3. Diagonal correlation

Simulations made on 150 images of the database gave the diagonal correlation scores are displayed in Figure 13

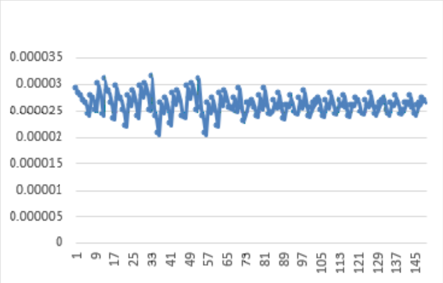


Fig. 13. Diagonal correlation of 100 images of the varying sizes.

Figure 11 shows that the diagonal correlation values of the encrypted images are close to zero. This ensures high security against correlation attacks.

2.6.3.3. Differential analysis

Let be two encrypted images, whose corresponding free-to-air images differ by only one pixel, from $(\boldsymbol{C_1})$ and $(\boldsymbol{C_2})$, respectively. The expressions of these two statistical constants $(NPCR) and\ (UACI)$ are given by equations 12 and 13, for an image size (n, m).

The $NPCR$ mathematical analysis of an image is given by the equation below.

$$Equation21 \qquad NPCR = \left(\frac{1}{nm} \sum_{i,j=1}^{nm} D(i,j) \right) * 100$$

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

The $UACI$ mathematicals analysis of an image is given by the equation 36

$$Equation\ 22 \qquad UACI = \left(\frac{1}{nm} \sum_{i,j=1}^{nm} Abs\big(C_1(i,j) - C_2(i,j)\big) \right) * 100$$

The study of the 150 selected images revealed the following diagram.

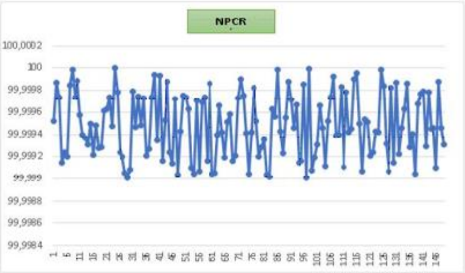


Fig. 14. NPCR of 150 images of the varying sizes.

All detected values are inside the confidence interval [99,63 99,95]. These values are largely sufficient to affirm that our crypto system is protected from known differential attacks.

The study of the 150 selected images revealed the following diagram.

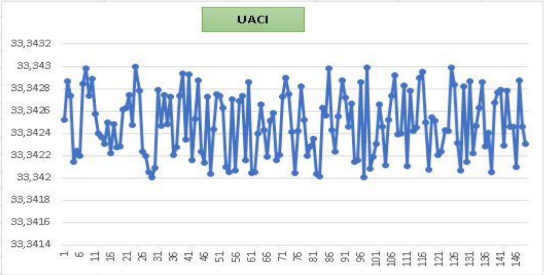


Fig. 15. UACI of 150 images of the varying sizes.

All detected values are inside the confidence interval [33,34 33,35]. These values are largely sufficient to affirm that our crypto system is protected from known differential attacks.

2.6.3.4. Avalanche effect

The avalanche effect is a required property in virtually all cryptographic hash functions and block coding algorithms. It causes progressively more important changes as the data is propagating in the structure of the algorithm. Therefore, by perturbing a single bit at the input, we can obtain a very different output, (about 1 bit our of 2 changed) explaining the name of this phenomenon. The avalanche effect makes it more difficult to reverse the function due to its chaotic properties (if well designed).

This constant determines the avalanche impact of the cryptographic structure in place. It is approximated by the next equation.

$$Equation23 \quad AE = \left(\frac{\sum_i bit\ change}{\sum_i bit\ total} \right) * 100$$

Figure below depicts the evaluation of the AE score for 150 images examined by our approach.

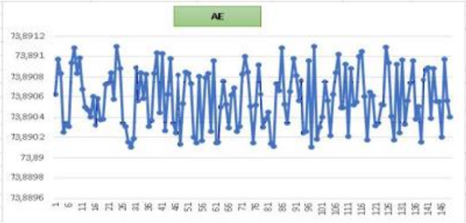


Fig. 16. Avalanche effect.

All values returned from the AE by our method are all in the range of residual values [73.96 74.02]. This guarantees that a one- bit change in the clear image will be reflected by a change of at least 78% of the encrypted image's bits.

2.6.3.5. Signal-To-Peak noise ratio (PSNR)

2.6.3.5.1 MSE

The image quality estimation to be based on the pixel change was obtained by processing the PSNR values and the MSE. These are the error metrics used to compare the image and the cipher image.

Mean Square Error MSE: This is the cumulative square deviation between the original image and the additional noise image. When the MSE level is reduced, the error is reduced.

This constant measure the distance between the pixels of the clear image and the encrypted image. It is calculated by the next equation.

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

$(\boldsymbol{P}(\boldsymbol{i},\boldsymbol{j}))$: pixel of the clear image

$(\boldsymbol{C}(\boldsymbol{i},\boldsymbol{j}))$: pixel of the cypher image

2.6.3.5.2 PSNR

The signal-to-peak noise ratio, often abbreviated PSNR, is a 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. Since many signals have a very large dynamic range, the PSNR is generally stated in terms of the logarithmic decibel scale. The PSNR mathematical analysis of an image is given by the next equation.

$$\boldsymbol{PSNR = 20Log_{10}\left(\frac{I_{max}}{\sqrt{MSE}}\right)} \qquad (25)$$

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. The acceptable PSNR values are the real numbers in the domain (5,10). Simulations made on over 150 images of various magnitudes and formats returned the same results as depicted in the figure 17.

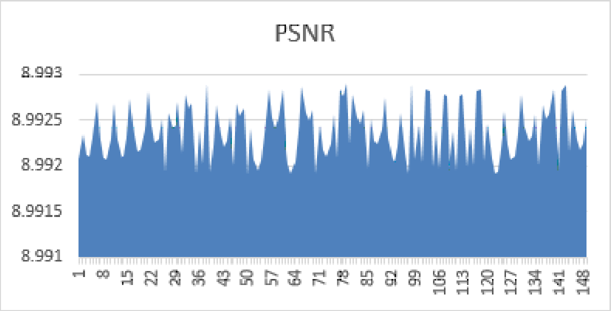
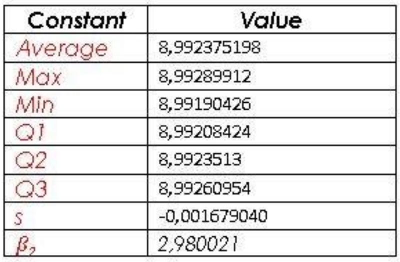


Fig. 17. PSNR of 150 images of the varying sizes.

All values returned from the PSNR by our method are all in the range of residual values [8,99 8,993]. A statistical study of the dispersions of the PSNR values of the 150 images analyzed by our algorithm reveals the scores presented in the following Table 1:

Table. 1. Parameter values.



Constant	Value
Average	8,992375198
Max	8,99289912
Min	8,99190426
Q1	8,99208424
Q2	8,9923513
Q3	8,99260954
s	-0,001679040
β_2	2,980021

This table ensures that there is a low dispersion and a high concentration of values in a length interval of 0.001. Moreover, the value of **s** close to 0 indicates that there is a symmetry in this dispersion and the value of $\boldsymbol{\beta_2}$close to 3 shows that our dispersion follows a normal distribution. More than 50% of the values achieved are within a longitudinal range of less than 0.0006.

2.6.3.6. Speed analysis

Assuming that the traditional DES and AES encryption algorithms operate in ECB mode, they are vulnerable to statistical attacks and selected plain text attacks. In addition, these two systems require no linking on clear and encrypted blocks and are consequently deficient in the face of differential attacks. In this sense, we will compare the time complexity for reference images with these two crypto systems. In addition to safety parameters, runtime is an important factor in evaluating image encryption system performance. To approve and document the quality of our methodology in a timely fashion. And finally, thanks to these properties, we have selected the "Lena" grayscale image with three different sizes (256×256) (512×512) and (1024×1024). The results are presented in the table below.

Table. 2. Execution time (in second).

Size	Our Method	DES	AES	Classic Hill	Hill [3]
256×256	0,097	0,639	0,568	0,192	0,081
512×512	0,172	7,449	0,354	0,214	0,201
1024×1024	0,201	29,112	1,152	0,921	0,852

We compare our results with the two classical algorithms AES and DES, Classic Hill and Improvement Hill, we can affirm that the time of execution is reasonable. The test was performed on other images of different sizes, and we obtained acceptable scores. This is due to the low algorithm complexity of the implemented algorithms in our strategy.

2.6.3.7 Math security

The large size of our encryption key ensures that the system is protected against any brute force attack. At the same time, the randomness of the genetic operator and the functions of Feistel's trick make it difficult to unlock the encryption system applied to a given block, increasing the difficulty of the statistical attack. In addition, the high sensitivity to the initial parameters of our three chaotic maps, and the statistical constants calculated in simulation make it difficult to reconstruct the encryption key.

III. CONCLUSION

Taking security as its primary objective, this document develops a new encryption framework for color images of arbitrary size. Based on chaos, this technique relies on 24-bit blocks by applying an improved Feistel round accompanied by genetic crossover, followed by chaining to install protection against any known attack. Simulations performed on more than 150 randomly selected images from a large database of color images of different sizes and formats confirm the robustness of our system.

References

[1] C. E Shanon, Communication theory of security systems, Bell syst Tech J., pp 656-715, 1949

[2] Hillion, Les theories mathematiques des population, P.U.F.1986.coll

[3] A.Jarjar, Improvement of hill's classical method in image cryptography, *International Journal of Statistics and Applied Mathematics*, vol. 2, no. 3, Part A, 2017.

[4] Hraoui S., Gmira F., Jarar A. O., Satori K. and Saaidi A., Benchmarking AES and chaos based logistic map for image encryption, in *Proceeding of International Conference Computer Systems and Applications (AICCSA)*, 2013.

[5] G. Zhang, Q. Liu, "A novel image encryption method based on total shuffling scheme", *Opt. Commun.*, pp. 284-2775, 2011.

[6] Xiao Feng, Xiaolin Tian and Shaowe iXia, "An Improved Image Scrambling Algorithm Based On Magic Cube Rotation and Chaotic Sequences", in *Proceeding of IEEE 4th International Congress on Image and Signal Processing*, pp. 1021-1024, 2011.

[7] Chang'e Dong, Color image encryption using one-time keys and coupled chaotic systems, *Signal Processing: image Communication*, vol. 29, no. 5, pp. 628-640, 2014.

[8] Xing-Yuan Wanga, Sheng-Xian Gua and Ying-Qian Zhangab, “Novel image encryption algorithm based on cycle shift and chaotic system,” *Signal Processing: Image Communication*. vol. 68, pp. 126-134, 2015.

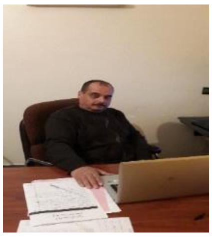
[9] G. Zhang and Q. Liu, “A novel image encryption method based on total shuffling scheme”, *Opt. Commun.*, pp. 2775-2780,2011.

[10] Tomoyasu Suzaki Kazuhiko Minematsu, “Improving the Generalized Feistel,” in *Proceeding of International Workshop on Fast Software Encryption FSE 2010*, pp. 19-39, 2010.

[11] Abdellatif JarJar, “Improvement of Feistel method and the new encryption scheme,” *Optik*, vol. 157, pp. 1319-1324, 2018.

[12] Jacques Patarin, “Security of Random Feistel Schemes with 5 or More Rounds,” in *Proceeding of* Annual International *Cryptology Conference CRYPTO 2004: Advances in Cryptology – CRYPTO*, pp. 106-122, 2004.

Author



ABDELLATIF JARAJAR is Cryptography Researcher. His major concern is a proof mathematics and he is a member of Moulay Rachid Hight School, Taza, in Morroco.