International Journal of Advanced Science and Computer Applications (2024); 3(2):
DOI: 10.47679/ijasca.v3i2.41
https://ijasca.org/index.php/ijasca

RESEARCH ARTICLE

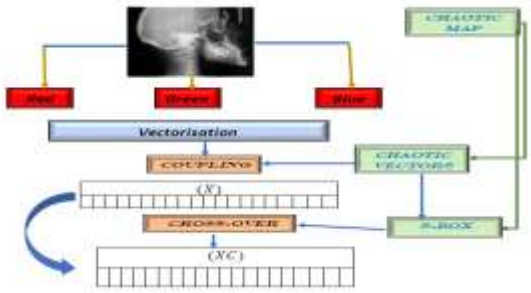
Utilizing a pseudo-random Linear Congruential Generator (LCG) S-Box for Encoding Color Images Through Genetic Crossover

Mariem Jarjar [1], Said Hraoui [2], Abid Abdellah[1], Faiq Gmira[3], Abdellatif JarJar [1], Abdellhamid Benazzi [1*)]

Published online: 26 September 2024

Abstract

This article introduces a novel algorithm crafted for encrypting color images. The algorithm leverages chaotic principles and harnesses fresh substitution tables derived from independent linear congruence generators. These generators are dynamically sized based on pseudo-random vectors used in this technology. The proposed method initiates with the original image, vectorization, employing selected chaotic maps. The primary goal revolves around implementing a genetic operator tailored for image encryption, who integrates an enhanced Vigenère technique, incorporating novel confusion and diffusion functions derived from the previously established substitution tables. To gauge the effectiveness of this approach, numerous color images of varying dimensions and formats underwent testing using our algorithm. The yielded outcomes are both promising and gratifying, furnishing heightened security against recognized attacks. Graphical abstract



Keyword: Vigenere grid; Chaotic map; Broadcast function; confusion function

Introduction

Due to the swift progress in chaos theory within the realm of mathematics, researchers now possess ample opportunities to improve a variety of traditional encryption methods. In addressing the vital security concerns, many techniques for encrypting color and medical images have surfaced in the digital domain. These methods predominantly make use of numerical approaches and the fundamental concepts of chaos $[1-2]$. While certain approaches intend to enhance established methods like Hill $[3-4]$, Cesar, Vigenere [5-6], and Feistel $[7-8]$ by integrating suggested enhancements, other strategies endeavor to introduce their own distinct methodologies.

Earlier works

The renewed interest in chaos theory within cryptography can be attributed to its inherent traits, such as its responsiveness to initial conditions and its ability to generate pseudo-random number sequences. These attributes render chaotic systems exceptionally appropriate for the encryption of images. As a result, there has been a notable increase in the creation of cryptographic techniques based on chaos, specifically designed for image encryption. These protocols are gaining popularity online. A majority of these algorithms are built upon the foundational principles laid out by Shannon.

In recent decades, several image encryption methods have been proposed in literature, and improved, namely:

The paper [9] presents a novel approach for encrypting color images, which involves combining the logistic map and sine map – both chaotic maps – to generate a more effective map. The technique entails taking the four most significant bits of each pixel in the image's channels, merging them, and subjecting them to a circular shifting pattern, both horizontally and vertically. This is done before applying pixel-

Mohamed First University, MATSI Laboratory, Oujda, Morocco

*) corresponding author

Email: abdoujjar@gmail.com

Received 21 May 2024 | Received in revised form 20 Juli 2024 | Accepted 22 August 2024
Available online 26 September 2024

level scrambling within each channel. The scrambling involves interchanging rows and columns across channels, following the sequences generated solely by the new map.

During the diffusion process, controlled diffusion sequences guided by control sequences are employed to spread the scrambled pixels within each channel. In a separate study by the authors [10], a fresh hybrid approach for encrypting color images is detailed. This method combines the concepts of Latin squares and chaotic systems within a permutation-substitution framework. The fusion of these elements allows for the incorporation of beneficial traits such as confusion and diffusion. Moreover, the approach demonstrates resilience in accommodating noise during the decryption process, owing to the inherent similarities between the two systems.

In the context of the paper [11], researchers have introduced an innovative strategy for encrypting color images. This method revolves around a linear function, assured by an invertible multiplier, in conjunction with a 2D hyperchaotic map that results from the fusion of two distinct 1D chaotic maps. This approach employs simple row and column shifts to disarrange the image's structure. Subsequently, a broadcasting function is applied to recover the encrypted image.

Ritesh Bansal and colleagues introduced a fresh approach to encrypting color images in their work [12]. This system integrates various chaotic maps with a Vigenère strategy. The encryption procedure encompasses a sole cycle of diffusion and confusion. The initial phase is comprised of three steps: reverse spreading, the Vigenère scheme's matching operation, and forward spreading. Following this, pixel coordinates undergo swapping facilitated by a chaotic map via position interchange.

Problematic

The majority of conventional systems remain vulnerable to frequency and statistical attacks, moreover, in the absence of any chaining between encrypted blocks and the next clear blocks, these systems are also exposed to differential attacks. Furthermore, this approach is particularly well-suited for encrypting large datasets characterized by substantial redundancy and strong correlations.

Our Contributions

This manuscript, will elaborate on the procedure for creating fresh substitution tables with varying sizes, achieved through the utilization of multiple (LCG). Furthermore, it will introduce a genetic crossover method tailored for encrypting extensive datasets. This method incorporates an enhanced Vigenère technique version, integrating the newly generated substitution tables to bolster the preservation of the original image's integrity.

Our approach Describe

Based on chaos $[13-14]$, this method is articulated on several main axes, cited below:

Axe1: chaotic maps Selection.

Our system is built upon the utilization of the two most extensively employed chaotic maps in cryptography $[15-16]$. These maps are renowned for their extreme sensitivity to initial conditions. They are generated using a sufficiently large key, rendering them immune to brute-force attacks. The two chaotic maps chosen for our approach are described below:

(1). The Logistics Map

The logistic map$(\boldsymbol{U_n})$ $[17-18]$.is a recurrent sequence described by a simple polynomial of second degree defined by the following equation

Logistics Map (u_n)

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

It is the most popular and widely used chaotic map in cryptography due to its high sensitivity to initial conditions and easy configuration in any cryptosystem, confirmed by its Lyapunov exponent value.

(2). The Skew Tent Map (SKTM)

The Skew tent map ($\boldsymbol{V_n}$) $[19-20]$.will be redefined as the next equation

The Skew Tent Map (v_n)

$$\begin{cases} v_0 \in]0\ \ 1[\quad p \in]0{,}5\ \ 1[\\ v_{n+1} = \begin{cases} \dfrac{v_n}{p} & \text{if } 0 < v_n < p \\ \dfrac{1-v_n}{1-p} & \text{if } p < v_{n<1} \end{cases} \end{cases} \qquad (2)$$

The integration of two chaotic maps will be employed to derive all the essential parameters required for the efficient and effective operation of our innovative technology.

Axe2: Constructing encryption settings

This phase, operates at the pixel level by applying a new, improved Vigenere technique, requiring the parameters construction below

- (CT) Tables for confusion and diffusion process.
- (BT)Binary tables for control and decision process.
- $(SW1)$; $(SW2)$ two big substitution tables.

(3). $(\boldsymbol{CT})$ Table Generation

The table (CT) of size $(3nm;\ 5)$ with coefficients in (G_{256}) is intended to act as aliasing and diffusion acting on the original image pixels level. Building such a table is described by the following algorithm:

Alg 1: (CT) design

1. $For\ \ i = 1\ to\ 3nm$
2. $CT(i;1) = \ mod(E(|u(i) - v(i)| * 10^{10}),\ 253) + 3)$
3. $CT(i;2) = \ mod(E((u(i) + v(i)) * 10^{10}), 254) + 2)$
4. $CT(i;3) = mod(E(Sup(u(i); v(i) * 10^{11}),\ 255) + 1$
5. $CT(i;4) = mod(E\left(\left(\frac{u(i)+2*v(i)}{3} * 10^{11}\right),\ 253\right) + 1$
6. $CT(i;5) = mod(E\left((u(i) * 10^{6} + v(i) * 10^{7}),\ 253\right) + 3$
7. $Next\ i$

Every individual column within the table (CT) signifies an independent pseudo-random vector distinct from the remaining vectors.

(4). (BT) Binary tables construction

The (BT) binary table of size $(3nm; 2)$, employed to control any cryptographic operations used in our new system, is developed by the following algorithm:

Alg2 (BT) design

1. $For\ i = 1\ to\ 3nm$

First column

2. $if\ u(i) > v(i)\ Then$
3. $BT(i;1) = 0\ Else\ BT(i;1) = 1$
4. $End\ if$

Second column

5. $if\ CT(i;2) \geq \mathrm{CT(i;4)} Then$
6. $BT(i;2) = 0\ Else\ BT(i;2) = 1$
7. $end\ if$
8. $Next\ i$

Every column within this table depicts a binary vector that will function as a control for one or multiple encryption procedures.

The $(CT)\ and\ (BT)$ table construction is viewed by the figure below:

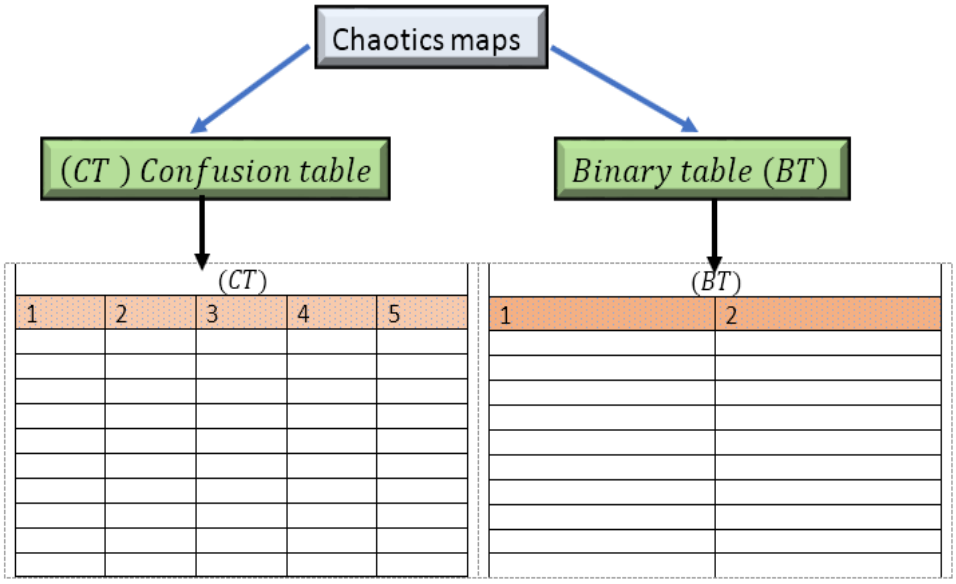


Fig 1. tables generation

Axe3: New S-Bos design

In this section, we will develop the process of generating two new substitution tables which will be used for the integration of a genetic crossing technique adapted to the encryption of medical images.

(5). S-Box construction

An $S - Box$, is a table used to modify the grayscale pixel value through a specific mathematical procedure. Consequently, each row within any S-Box embodies a pseudo-random permutation within (G_{256}). In our innovative methodology, the augmentation of these permutations is achieved through the utilization of (LCG), as indicated by our approach.

a) Mathematics reminder

A (LCG)can be described as a mapping defined by:

LCG definition

$LCG(s_0; a; b; m)$ (3)

1. $s_0 = u$
2. $s_{n+1} = mod(as_n + b, m)$

$(s_0; a; b; m)$ Is called, generator parameters. A good (LCG) is a map of maximal period equal to (m). Furthermore, displaying a random characteristic of the produced numbers is achieved, with Hull & Dobell's theorem offering the essential and complete requirement for a (LCG) to exhibit the pseudo-random attribute.

b) $Hull - Dobell\ theorem\ (1962)$

All (LCG) whose parameters adhere to the aforementioned theorem have (m), like maximum periodicity of these prerequisites are met by the connection depicted below.

Hull & Dobell Terms	
1. s_0 *unspecified integer in* $[\![0 \ ; \ m-1]\!]$ 2. $b \wedge m = 1;\ a \wedge m = 1$ 3. *if there exists a prime divisor p of m, then* a. *p divides the quantity* $a-1$. b. *if* $4/m$ *then* $4/a-1$	*(4)*

c) Particular case (m=256)

Consider the following (LCG), with $m = 2^k$

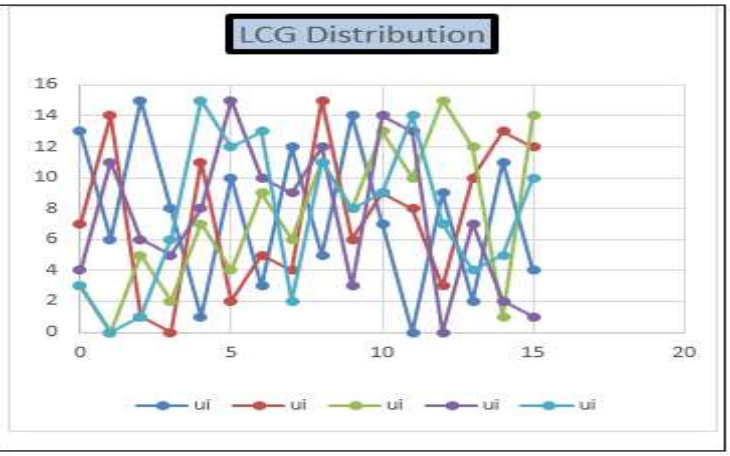
(LCG)	
$LCG(s_0; a; b; 256)$ $s_0 = u$ $s_{n+1} = mod(as_n + b\ ); 2^k)$	(5)

This generator has a period of 255 , if and only if:

(LCG)	
$LCG(s_0; a; b; 256)$ $s_0 = u\ \epsilon\ G_{256}$ $b = mod(2h+1; 2^k)$ $a = mod(4k+1; 2^k)$	(6)

Example in (G_{16}) :

Rank	ui	ui	ui	ui	ui
0	13	7	3	4	3
1	6	14	0	11	0
2	15	1	5	6	1
3	8	0	2	5	6
4	1	11	7	8	15
5	10	2	4	15	12
6	3	5	9	10	13
7	12	4	6	9	2
8	5	15	11	12	11
9	14	6	8	3	8
10	7	9	13	14	9
11	0	8	10	13	14
12	9	3	15	0	7
13	2	10	12	7	4
14	11	13	1	2	5
15	4	12	14	1	10



We notice that the distribution of (LCG) generators has a pseudo-random aspect. These generators will be used to create two substitution tables $(SW1)\ and\ (SW2)$, whose size will be calculated from the pseudo-random vectors used.

In all that follows, we will adopt the case:

a) S-Box size computation

Let (t) be the height of each S-Box, which represents the number of rows. The integer (t) is selected from the pseudo-random table (CT) while being influenced by the table (BT). The value of this constant (t) is determined by the following algorithm:

Alg3 (t) calculation	
1. $t = 0$ 2. *For* $i = 1$ *to* $2nm$ 3. *If* $BT(i; 2) = 0$ *then* 4. $x = CT(i; 1) + CT(i; 2) * CT(i; 4)$ 5. $x = mod(x; n)$	7. $x = CT(i; 1) * CT(i; 2) + CT(i; 3)$ 8. $x = mod(x; m)$ 9. *end if* 10. $t = mod(t + x; n)$ 11. *Next i*

6. *Else*	12. $t = mod(t; n) + m$

We note that, the constant (t) is framed by $m \leq t \leq n + m$.

For each S-Box, we will need (t), (LCG) satisfying Hull & Dobell's theorem. As each generator has a parameter $(s_0; a; b; 256)$ satisfying the Hull & Dobell theorem defined by equation (6),we will store them in a table (LG) $of\ size$ $(3; t)$.

b) (LG) parameter table design

A(LCG) is determined by the recurrence relation below, meeting the Hull & Dobell theorem's criteria

(LCG)

1. $s_0 = u$
2. $s_{n+1} = mod(as_n + b; 256)$
3. s_0 $unspecified\ integer\ in\ [\![0\ ;\ 255,]\!]$
4. $b \equiv 1\ [2]\quad b = 2k + 1$
5. $a \equiv 1\ [4],\ \ a = 4k + 1$

(8)

The three parameters (s_0, a, b) of a (LCG) used in our algorithm for the development of two substitution tables $(SW1)$ and $(SW2)$ will be stored in arrays (LG) of size $(3; t)$, by the following process:

- The initial value of the generator in (G_{256}), denoted as (s_0), will be present in the first line as pseudo-random seed.
- The second line will contain the values of the multiplier parameter (a=4k+1)
- The third line will contain the values of the bias parameter (b=2k+1).

The developments table (LG) is given by the following algorithm:

Alg4: (LG) Table design

1. *For* $j = 1$ *to* t
2. *If* $BT(j; 2) = 0$ *Then*
3. $LG(1, j) = CT(j; 3)$
4. $LG(2, j) = mod(4 * CT(j; 2) + 1{,}256)$
5. $LG(3, j) = mod(2 * CT(j; 3) + 1{,}256)$
6. *Else*
7. $LG(1, j) = CT(j; 2)$
8. $LG(2, j) = mod(4 * CT(j; 1) + 1{,}256)$
9. $LG(3, j) = mod(2 * CT(j; 4) + 1{,}256)$
10. *Next* j

This architecture, governed by pseudo-random vectors, is extremely sensitive to any mutation of any private key component; This reinforces the robustness of a system.

Example: in(G_{16}) and $(t = 12)$, we obtain for example the following table (LG)

Rank	1	2	3	4	5	6	7	8	9	10	11	12
(s_0)	5	2	4	5	8	12	16	9	7	2	1	3
$(a = 4k + 1)$	5	9	1	1	9	5	13	5	13	9	5	9
$(b = 2k + 1)$	3	5	3	9	7	13	15	11	3	9	5	7

An additional set (LCG) becomes an example.

Example:

$LCG(LG(1; 3); LG(2; 5); LG(3; 6); 16)$

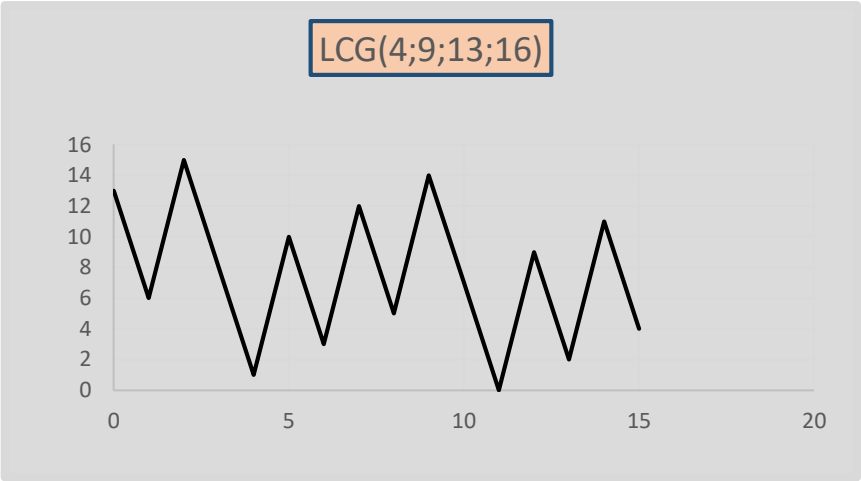
1. $s_0 = LG(1; 3) = 4$
2. $s_{n+1} = mod(LG(2; 5)s_n + LG(3; 6); 16)$
   a. $s_{n+1} = mod(9s_n + 13; 16)$

(9)

The values of the generated (LCG) are:

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s_i	13	6	15	8	1	10	3	12	5	14	7	0	9	2	11	4

The graphical representation of this $(LCG)$is given below:



To give the pseudo-random aspect to the S-Boxes construction; we are going to use two new tables $(TX)\ and\ (TY) of\ size\ (t; 4)$.

c) (TX) design

To achieve the pseudo-random $(SW1)$ construction, we will create a control table (TX) with dimensions $(t;\ 4)$. Each column of this table will correspond to a permutation within the group $(\boldsymbol{G}_t)$, and this table will be generated through the following outlined process.

1. *The first column is a permutation* $(P1)$ *obtained by sorting on the* (t) *first values of the vector* $CT(:4)$
2. *The second column is a permutation* $(P2)$ *obtained by sorting on the* (t) *first values of the vector* $CT(:3)$
3. *The third column is a permutation* $(P3)$ *obtained by sorting on the* (t) *first values of the vector* $CT(:1)$
4. *The fourth column is a permutation* $(P4)$ *obtained by sorting on the* (t) *first values of the vector* $CT(:2)$

The algorithm that establishes the table (TX) containing the parameters of the (t) linear congruence generators for crafting these two S-Boxes is as follows:

$Alg5:\ \ (TX)\ developpement$

1.
2.
3.
4. $TX(i;3) = P3(i)$
5. $TX(i;4) = P4(i)$
6. $Next\ i$

d) Table description (TX)

The construction of the table $(SW1)$ is described by the following steps:

1. *The first column indicates the index of the parameter* (s_0) *as initial condition of the generator*
2. *The second column indicates the index of the parameter* (a) *as multiplier of the generator*
3. *The third column indicates the index of parameter* (b) *as the bias of the generator*
4. *The fourth column the number of the line where to place the generator in the S-Box.*

This can be described by: the line $LG\big(TX(i;4)\big)$of the table $(SW1)$ is defined by the (LCG) given by:

$$SW1(TX(i;4):) = LCG\big(LG\big(TX(i;1)\big); LG\big(TX(i;2)\big); LG\big(TX(i;3)\big); 256\big)$$

e) (TY) design

To achieve the pseudo-random $(SW2)$ construction $[19-20]$, we will create a control table (TY) with dimensions $(t;\ 4)$. Each column of this table will correspond to a permutation within the group $(\boldsymbol{G}_t)$, and this table will be generated through the following outlined process.

1. *The first column is a permutation* $(P5)$ *obtained by sorting on the* (t) *last values of the vector* $CT(:2)$
2. *The second column is a permutation* $(P6)$ *obtained by sorting on the* (t) *last values of the vector* $CT(:4)$
3. *The third column is a permutation* $(P7)$ *obtained by sorting on the* (t) *last values of the vector* $CT(:5)$
4. *The fourth column is a permutation* $(P8)$ *obtained by sorting on the* (t) *last values of the vector* $CT(:1)$

The algorithm that establishes the table (TY) containing the parameters of the (t) linear congruence generators for crafting these two S-Boxes is as follows:

Alg6: (TY) developpement

1.
2.
3.
4.
5. $TY(i;4) = P8(i)$
6. *Next i*

1.	*The first column of* (TY) *indicates the index of the parameter* (s_0) *as initial condition of the generator*
2.	*The second column of* (TY) *indicates the index of the parameter* (a) *as multiplier of the generator*
3.	*The third column of* (TY) *indicates the index of parameter* (b) *as the bias of the generator*
4.	*The fourth column of* (TY) *indicate the number of the line where to place the generator in the S-Box.*

This can be described by: the line $LG(TY(i;4))$of the table $(SW2)$ is defined by the (LCG) given by:

$$SW2(TY(i;4):)LCG(LG(TY(i;1)); LG(TY(i;2)); LG(TY(i;3)); 256)$$

1) $(SW1)$ and $(SW2)$ Construction

The two S-Boxes constructed aim to modify the value of each pixel according to a well-defined confusion function. This new development technique is described by the following algorithm:

Alg7: S – Boxes Development

1.
2. $s = LG(TX(i,1);\ a = LG(TX(i,2);\ b = LG(TX(i,3)$
3. $h = LG(TY(i,4);\ c = lG(TY(i,5);\ d = LG(TY(i,6)$
4.
5.
6.
7.
8. *Next j, i*

Example: In (G_{16})

Rank	1	2	3	4	5	6	7	8	9	10	11	12
s_0	3	2	0	5	7	9	11	3	5	7	10	15
$a = 4k + 1$	1	5	5	9	13	9	1	5	13	9	13	1
$b = 2k + 1$	7	5	7	9	11	13	3	15	11	13	7	5

$(SW1)$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	7	14	9	8	11	2	13	12	15	6	1	0	3	10	5	4
2																
3	3	14	9	4	15	10	5	0	11	6	1	12	7	2	3	8
4																
5																
6	2	13	12	15	6	1	0	3	10	5	4	7	14	9	8	11
7																
8																
9																
10																
11																
12																

(TX)

	1	2	3	4
1	2	5	6	6
2	5	3	3	1
3	1	7	5	3
4	9	10	10	8
5	10	6	7	9
6	6	12	12	7
7	8	9	8	12
8	12	4	2	4
9	4	8	9	11
10	6	1	1	5
12	3	2	7	2

Line6	*Line1*	*Line3*
$\begin{cases} s_0 = 2 \\ s_{n+1} = mod(13s_n + 3; 16) \end{cases}$	$\begin{cases} s_0 = 7 \\ s_{n+1} = mod(5s_n + 7; 16) \end{cases}$	$\begin{cases} s_0 = 3 \\ s_{n+1} = mod(s_n + 11; 16) \end{cases}$

At this point, all the requisite parameters to ensure the smooth advancement of the encryption procedure have been generated. The next step involves preparing the original image for testing within our system.

Axe4: Preparing the image for encryption

After extracting the three (RGB) color channels and converting them to magnitude vectors $(Vr), (Vg), (Vb)$ $(1, nm)$, create a confound connected to a table (TC) to generate the vector $X(X_1; X_2 \dots\dots; X_{3nm})$. This operation is described by the following figure:

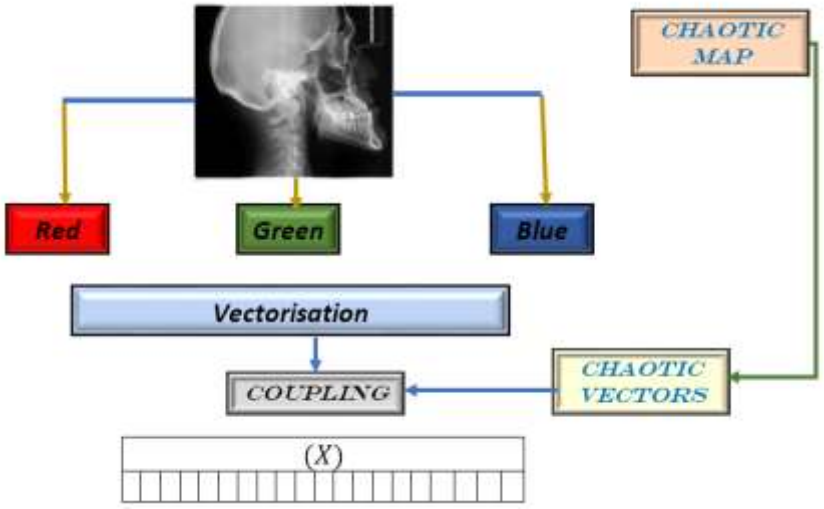


Fig 2. Original image vectoring

This figure is described by the algorithm below:

Alg8: Switching to vector (X)

1. $for\ i = 2\ to\ nm$
2. $If\ BT(i;2) = 0\ Then$
3. $X(3i-2) = Vr(i) \oplus CT(3i-2;2)$
4. $X(3i-1) = Vg(i) \oplus CT(3i-1;1)$
5. $X(3i) = Vb(i) \oplus CT(3i;4)$
6. $Else$
7. $X(3i-2) = Vr(i) \oplus CT(3i-2;3)$
8. $X(3i-1) = Vg(i) \oplus CT(3i-1;5)$
9. $X(3i) = Vb(i) \oplus CT(3i;2)$
10. $End\ if$
11. $Next\ i$

Fig 3. This new original image method vectorization, involving pseudo-random vectors, makes it possible to affirm that the system is protected from any statistical and frequency attack.

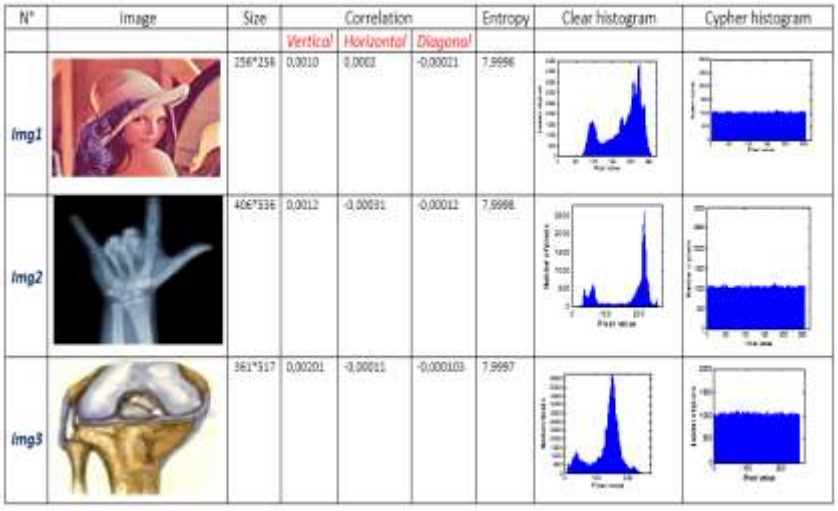


Fig 4. Calculated first-round correlation

It is important to observe that all statistical measures associated with this initial operation conform to global standards. This guarantees the security of our system against statistical and frequency-based vulnerabilities.

Axe5: second intervention

To enhance resistance against differential attacks, a second operation, is implemented, utilizing an upgraded Vigenère and two newly generated pseudo-random S-boxes. This approach incorporates the use of binary vectors and a cross table controlled by a pseudo-random generator, combined with chaotic vectors. This new intervention requires the initialization value calculation for starting the encryption process

The second color image encryption process involves the following steps:

(1). Initialization Value design

To change the seed pixel's value and launch the encryption process, an initialization value is computed from the original image. The following algorithm identifies this initialization value:

Alg9: Fist initialization value

1. $IV = 0$
2. $For\ i = 2\ to\ 3nm$
3. $If\ BT(i;1) = 0\ Then$
4. $IV = X(i) \oplus IV \oplus CT(i;4)$
5. $Else$
6. $IV = X(i) \oplus IV \oplus CT(i;1)$
7. $Next\ i$

We find that this computed value correlates strongly with the original image and the control vector (BT). A simple glitch in the original image or in one of the private key parameters creates a new, different initialization value and results in a different encrypted image.

(2). New Vigenere's mathematical expression

The advanced use of S-Boxes requires the development of a pseudo-random table $(GM)\ of\ size\ (3nm;2)$ with coefficient in (G_t). This construction is given by the following algorithm:

Alg10: (GM)table

1. $For\ i = 1\ to\ 3nm$
2. $If\ TB(i;1) = 0\ Then$
3. $GM(i;1) = mod(CT(i;4);t) + 1$
4. $GM(i;2) = mod(CT(i;5);t) + 1$
5. $Else$
6. $GM(i;1) = mod(CT(i;3);t) + 1$
7. $GM(i;2) = mod(CT(i;1);t) + 1$
8. $Next\ i$

(3). confusion function expression

The expression for the image of pixel $X(i)$ represented by two matrices $(SW1)$ and $(SW2)$ is given by the following formula:

Confusion function

$$VG\big(X(i)\big) = Y(i) \ = SW1\big(GM(i;1), SW2(GM(i;2), X(i) \oplus CT(i;2))\big) \oplus CT(i;4) \quad (10)$$

A modification of the initialization value by the new Vigenère function will make it possible to modify the value of the first pixel and to start the encryption process.

(4). Diffusion function expression

(φ) The new diffusion function using matrix $(SB1) and\ (SB2)$ is described by the following formula:

Broadcasting function

$$\varphi\big(X(i+1)\big) \ = SW1\big(GM(i;2); Y(i) \oplus X(i+1)\big) \oplus CT(i;3) \quad (11)$$

This second phase of this encryption method, will put the vector (X) in genetic crisis with the table (CT) for the breeding an image inheriting the properties of the two tables, and be able to overcome any known attack.

Axe6: Genetic cross-breeding suitable for cryptography

(1). Genetic crossing

Genetic crossbreeding is a breeding strategy used in zoology and botany that involves mating animals from several populations or species to create hybrids. It enables the exploitation of dominance's genetic variation. Crossings incorporate the idea of races and interbreeding rather than the concept of species, in contrast to hybridization strictly speaking.

In our approach genetic crossover is an interaction between the vector (X) and a confounding table vector (CT) under the control of the vector (BT). This genetic crossing will be entirely governed by a pseudo-random table (RC), called a crossing table.

a) Building the (RC)cross table

A table (RC) of size $(3nm;3)$, will be built by the following steps:
The first column is the permutation $(Pr1)$ obtained by an ascending sort on the $3nm$ values of the sequence (U),
The second is the permutation $(Pr2)$ obtained by a decreasing on the $3nm$ values of the sequence (V).
The third is the permutation $(Pr3)$ obtained by a decreasing on the $3nm$ values of the sequence $(CT(:5)$.

b) Description of the (RC) columns

- The first column of the table (RC) indicates the index of the pixel to be encrypted.
- The second column of the table (RC) indicates the index of the pixel f the $(CT(:3))$ Selected for crossbreeding.
- The Third column of the table (RC) The position of the pixel obtained by this crossing in the encrypted vector

The following graphic serves as an illustration of the genetic crossover described in our method:
This encryption scheme using both S-Box and broadcast functions, is translated by the algorithm below:

Algorithm11: New encryption scheme algorithm

1. *First pixel encryption*
2. $Y(RC(1;4)) = VG((X(RC(1;1))) \oplus CT(1;3))$
3. *Next Pixel Encryption*
4. *For* $i = 2$ *to* $3nm$
5. $x = \varphi\left(X(RC(1;1))\right)$
6. *If* $TB(1;i) = 0$ *Then*
7. $Y(RC(i;4)) = VG\left(\varphi\left(X(RC(i;1))\right)\right) \oplus CT(RC(i;1);2)$
8. *else*
9. $Y(i) = VG\left(\varphi\left(X(RC(1;1))\right)\right) \oplus CT(RC(i;1);3)$
10. *Next i*

We note that this first step uses only substitutions, which ensures an extreme speed in the execution. The output vector $Y(Y_1, Y_2, \dots\dots\dots, Y_{3nm})$, will considered as the encryption image

(2). Decryption function building

The system used is a symmetric encryption system, therefore the decryption process must start with the last encrypted pixel and use the calculated inverse functions. The decryption process follows the steps below

a) Reverse Vigenere application

The calculation of the inverse substitution table is given by the following algorithm

Alg12: Reverse S – Box

1. $for\ i = 1\ to\ t$
2. $for\ j = 1\ to\ 256$
3. $WS1(SW1(i,j),j) = i$
4. $WS2(SW2(i,j),j) = i$
5. $Next\ j, i$

b) Inverse of the Vigenere function

We note that

Expression of the inverse function

1. *If* $z = SW1(k;x)$
2. *Then*
3. $x = WS1(k;z)$

c) Inverse of confusion function

We have:

$$VG(X(i)) = Y(i) = SW1(GM(i;1), SW2(GM(i;2), X(i) \oplus CT(i;2))) \oplus CT(i;4)$$

So

(Ψ^{-1}) *expression*

1. $Y(i) \oplus CT(i;4) = SW1(GM(i;1), SW2(GM(i;2), X(i) \oplus CT(i;2)))$
2. $SW2(GM(i;2), X(i) \oplus CT(i;2)) = WS1(GM(i;1), Y(i) \oplus CT(i;4))$
3. $X(i) \oplus CT(i;2) = WS2(GM(i;2);\ WS1(GM(i;1), Y(i) \oplus CT(i;4)))$
4. $X(i) = WS2(GM(i;2);\ WS1(GM(i;1), Y(i) \oplus CT(i;4))) \oplus CT(i;2)$

d) Inverse of diffusion function

We have:

(φ^{-1}) *expression*

1. $\varphi(X(i+1)) = SW1(GM(i;2); Y(i) \oplus X(i+1)) \oplus CT(i;3)$
   So
2. $\varphi(X(i+1)) \oplus CT(i;3) = SW1(GM(i;2); Y(i) \oplus X(i+1))$
3. $Y(i) \oplus X(i+1) = WS1\left(GM(i;2);\ \varphi(X(i+1)) \oplus CT(i;3)\right)$
4. $X(i+1) = WS1\left(GM(i;2);\ \varphi(X(i+1)) \oplus CT(i;3)\right) \oplus Y(i)$

Axis 6: Examples and simulations

This section will provide and study the performance results of our new method on a large number of color and medical images that have been randomly sampled from a big database. We are aware that an effective algorithm is one that can defend against every known assault. The contents of the following graphic serve as a simplified extract from several reference images often used test our prediction model.

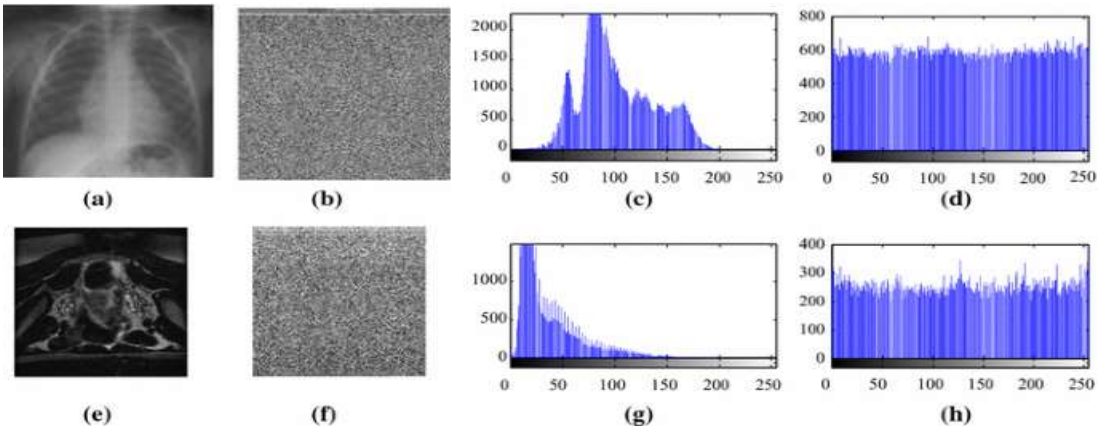


Fig 5. a-e: Original Image, b-f: Cypher Image, c-g: Original Histogram, d-h: Cypher Histogram

Fig 6. Simulation results of black image

(3). Brutal assaults

They consist in reconstructing the encryption keys in a random manner.

a) Key-space analysis

The brute force attack is a technique used in cryptanalysis to find the find the encryption key. It consists in reconstructing the encryption key using all possible combinations. As a result, a large encryption key makes brute force attacks impossible. In our algorithm, the size of the secret key is much larger than (2^{100}) $[18-19-20]$, which guarantees a better protection against brute force attacks. The secret key of our simulation is generated from the three most used chaotic maps in cryptography. As a result, the size of our key consists of six parameters written in 32 bits each, and thus, the global size is $(2^{6*32}) = (2^{192}) \gg (2^{100})$.

b) Secret key's sensitivity Analysis

All the chaotic maps used in our algorithm are very sensitive to the initial conditions. Therefore, any perturbation on at least one parameter for the regeneration of the secret key, will produce a different key, consequently will generate different chaotic vectors. This high sensitivity of the key of our new technology, can be seen in the following figure

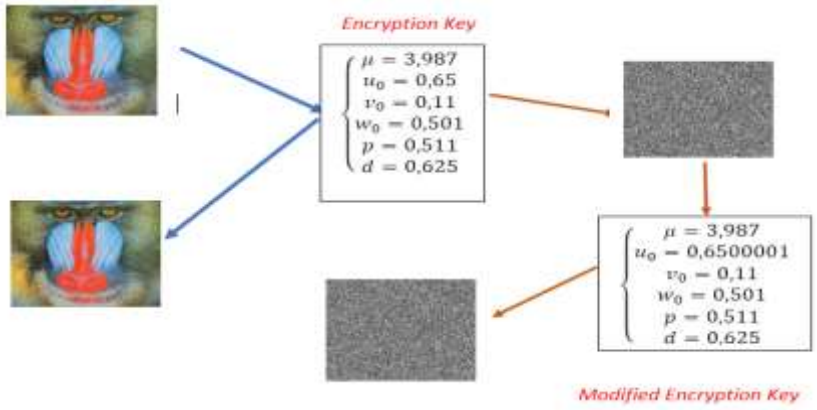


Fig 7. Encryption key sensitivity

We note that a perturbation of the order of (10^{-7}) on a single parameter, is not able to reconstruct the original image.

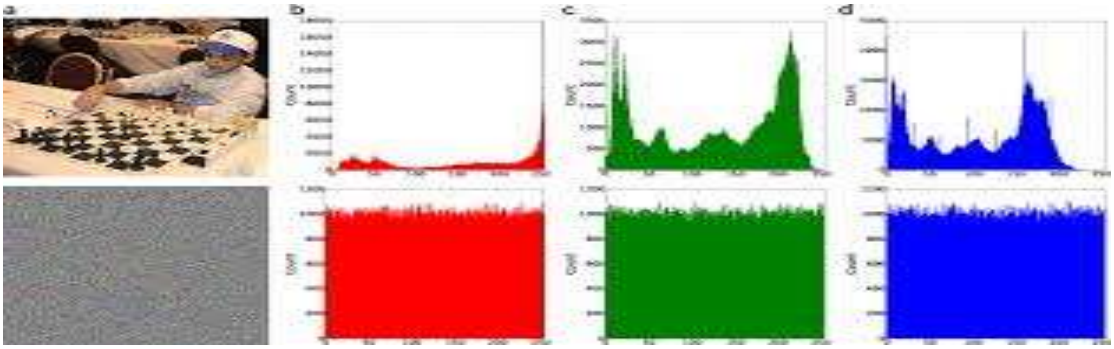
(4). Statistics Attack Security

In order to prove the robustness of our new medical and color image encryption technique against statistical attacks, many tests on images have been performed, of which we cite the most interesting.

a) Histogram analysis

An image histogram is a graphical representation of the number of pixels with the same level of gray. From a cryptographic point of view, this color distribution of an encrypted image is of great importance because it can reveal more information about the original image. On the other hand, if the histogram of the encrypted image is flat and has some uniformity, it may not provide any information about the original image or the relationship between it and the encrypted image. Some simulation results of our system are shown in Table 1.

Table 1. Histogram analisys



In this work, we notice that all the reference images tested by our algorithm, produce encrypted images whose histogram is flattened. This uniformity of the histogram, proves a great protection against any attack by histogram.

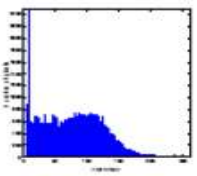
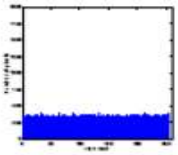
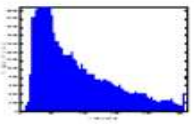
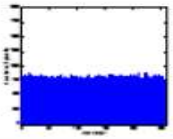
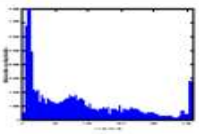
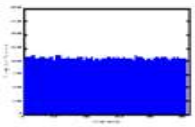
b) Entropy Analysis

between the pixels of an image is given by the following equation

$$H(MC) = \frac{1}{t}\sum_{i=1}^{t} -p(i)\, log_2(p(i))$$

$p(i)$ is the probability of occurrence of level (i) in the original image. Table 2 shows the entropy of the images tested by our technique

Table 2. Entropy analisys

IMAGE		HISTOGRAM		ENTROPY	
ORIGINAL	CYPHER	ORIGINAL	CYPHER	ROUND1	ROUND2
				7,9992	7,9996
				7,9995	7,9997
				7,9994	7,9998

c) Correlation analysis

Correlation is a technique that compares two images to estimate the displacement of pixels in one image relative to another reference image. The relevant expression is defined by the following equation

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

Table 3. Correlation coefficients

		Original Image			Encrypted Image		
Image	Size	H-C	V-C	D-C	H-C	V-C	D-C
	512x512	*0.9047*	*0.8520*	*0.8238*	*-0.0007*	*-0.0004*	*0.0001*
	1024x1024	*0.9774*	*0.9813*	*0.9668*	*-0.0001*	*-0.0002*	*-0.0010*
	512x512	*0.9786*	*0.9820*	*0.9694*	*-0.0002*	*0.0006*	*0.0002*

We notice that the value of the correlation is close to zero, which guarantees a better protection against correlation attacks.

Each pixel in the original image is highly correlated with its adjacent in the horizontal, vertical or diagonal direction. This correlation holds information that the attacker can use to reproduce the original image. A robust encryption system must reduce this correlation to the minimum possible. This value should be close to zero. The value of the correlation

(5). 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 $NPCR$ mathematical analysis of an image is given by the equation below

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

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

The $UACI$ mathematical analysis of an image is given by the below

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

Table 4: NPCR and UACI for different images

NPCR and UACI values.

Image						
NPCR	99.61	99.61	99.60	99.63	99.61	99.62
UACI	33.42	33.45	33.44	33.45	33.47	33.43

The differential values calculated for the reference images tested by our new technology are within the universal standards. It is then that the value of the $NPCR$ is close to 99.99% and that of the $UACI$ exceeds 34.65%. This ensures that our encryption system is safe from differential attacks. This protection is due to the installation of the first round.

(6). 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. This constant determines the avalanche impact of the cryptographic structure in place. It is approximated by the equation below

$$\left(\frac{\sum_i bit\ change}{\sum_i bit\ total}\right) * 100$$

Table5: Avalanche Effect

Image Name	Size	AE Value
Lena	(256x266)	78,25
Brain	(256x266)	77,95
Baboon	(256x266)	77,96
Oman	(256x266)	78,23
Girl	(256x266)	78,62
Men	(1024x1024)	77,26
Covid19	(512x512)	77,89
Girl	(256x256)	79,12
Cat	(128x128)	78,95
Aeroplain	(256x256)	79,20

The avalanche effect values from the images tested by our technology provide strong protection against differential attacks.

(7). Measures of central tendency and dispersion

a) The arithmetic mean

It is obtained by dividing the sum of all the values in the sample by the sample size (3nm). This measure is sensitive to extreme values.

b) The median

is the value such that 50% of the observations in the sample are below it.

On a histogram, the mode is the "highest point" of the distribution, the median is the value that divides the area in two and the mean is the center of gravity of the distribution, as in this illustration:

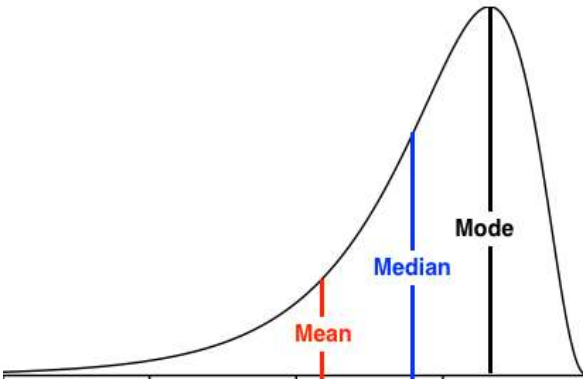


Fig 8. central tendency and dispersion

The mean, median, and standard deviation of the test images are calculated to measure dispersion and central tendency, respectively. Table 4 presents the comparative results which establishes that these measures have uniform values for the encrypted images while they varying values for the original images.

Table 4. Mean, median and mode of the original image and the cipher image

	Value					
	Mean		Medium		deviation	
Image Name	*Originale*	*Cypher*	*Originale*	*Cypher*	*Originale*	*Cypher*
Lena	124	127,6	129	128	47,9	74,3
Baboon	129	127,6	130	128	42,3	73,9
Peppers	104	127,5	108	128	57,2	72,9
Clock	186	127,5	215	128	57,5	73,2

(8). Math security

The algorithm implemented in our system employs a substantial symmetric key to generate chaotic sequences that possess a high degree of sensitivity to initial conditions and control parameters. This property ensures that even a minor alteration in the key will lead to the generation of new subject, calculation chaotic vectors, and an updated S-Box. Additionally, the inclusion of discrete logarithms in our calculations enhances the complexity, thereby significantly increasing the level of difficulty for potential attacks on our systems.

a) Advantages of this process

This method encapsulates several advantages of which we mention

- Encryption keys from chaotic maps are extremely sensitive to initial conditions, making it difficult to recover the real key used.
- Expanding arbitrarily sized substitution table.
- chaotic crossing with confusion and diffusion protects the system against any known attack
- A chaotic mutation for the generation of the encrypted image

c) Approach limitation

In essence, despite the classification of modern encryption systems as unbreakable, it is theoretically possible to break any encryption method with state-of-the-art technology and a significant investment of time and computational resources. The race between code-cracking capabilities and encryption algorithms continues due to the continuous advancements in computer speed. It is highly probable that even the most robust and secure encryption systems will be compromised and surpassed within a few years. However, these very technological advancements will also facilitate the development of the next generation of unbreakable encryption methods. It is crucial to acknowledge that the landscape remains volatile as computers constantly evolve, becoming more powerful and energy-efficient. The extreme calculation approach mentioned above can lead to exponential progress in these two aspects, rendering current encryption methods ineffective.

The limitations of our method primarily depend on the choices of chaotic maps, the construction of S-Boxes, and the size of the developed pseudo-random S-Boxes.

Conclusion

This article introduces an innovative method for encrypting color images of any size. The second encryption round involves the utilization of two substitution tables generated by multiple linear congruential generators, following the principles outlined in the Hull and Dobell theorem. These substitution tables are integrated with sophisticated replacement functions to enable crossover between the output vector from the first round and a randomly selected chaotic vector, controlled by a pseudo-random crossover table. Subsequently, a genetic mutation process is implemented, guided by a mutation table and a control binary vector, to enhance the system's complexity and generate the encrypted image. The conversion of the confusion table into a vector format facilitates the production of a fully protected encrypted image that is resistant to attacks.

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] L- Li, « Image encryption using chaotic map and cellular automata" Multimedia Tools and …, 2022 - Springer

[2] U Zia, » Survey on image encryption techniques using chaotic maps in spatial, transform and spatiotemporal domains » international Journal of information security, 2022 - Springer

[3] Y Qobbi « New image encryption scheme based on dynamic substitution and hill cipher » WITS 2020: Proceedings of the …, 2022 - Springer

[4] M Jarjar « New technology of color image encryption based on chaos and two improved Vigenère steps » Multimedia Tools and …, 2022 - Springer

[5] A JarJar « Vigenere and genetic cross-over acting at the restricted ASCII code level for color image encryption » Medical & Biological Engineering & Computing, 2022 - Springer

[6] Hraoui.S.; Gmira.F.; Jarar, A.O.; Satori.; Saaidi.A: Benchmarking AES and chaos based logistic map for image encryption. Computer Systems and Applications (AICCSA), 2013 ACS International Conference

[7] A Abid, « Two Enhanced Feistel Steps for Medical Image Encryption » 2022 IEEE 3rd …, 2022 - ieeexplore.ieee.org

[8] L Guo, « A quantum image encryption algorithm based on the Feistel structure » Quantum Information Processing, 2022 - Springer

[9] A Abdellah, « Vigenère Implemented in Two Chaotic Feistel Laps for Medical Images Encryption Followed by Genetic Mutation » Conference on Artificial …, 2022 - Springer

[10] Machkour, M., Saaidi, A., Benmaati, M.L.: A Novel Image Encryption Algorithm Based on the Two-Dimensional Logistic Map and the Latin Square Image Cipher. 3D Res. 6, 1–18 (2015). https://doi.org/10.1007/s13319-015-0068-1

[11] Gao, X.: Image encryption algorithm based on 2D hyperchaotic map. Opt. Laser Technol. 142, 107252 (2021). https://doi.org/10.1016/j.optlastec.2021.107252

[12] S Hraoui, M Gouiouez, F Gmira, M Berrada, A Jarjar « A novel cryptosystem for color images based on chaotic maps using a random controller » WITS 2020, 2022

[13] A JarJar "New chaotic map development and its application in encrypted color image" Journal of Multimedia Information System, 2021 - jmis.org

[14] Teng, L., Wang, X., Xian, Y.: Image encryption algorithm based on a 2D-CLSS hyperchaotic map using simultaneous permutation and diffusion. Inf. Sci. (Ny). 605, 71–85 (2022). https://doi.org/10.1016/j.ins.2022.05.032

[15] Huang, L., Cai, S., Xiao, M., Xiong, X.: A simple chaotic map-based image encryption system using both plaintext related permutation and diffusion. Entropy. 20, 1–20 (2018). https://doi.org/10.3390/e20070535

[16] Y Mao, » A novel fast image encryption scheme based on 3D chaotic baker maps » Journal of Bifurcation and chaos, 2004 - World Scientific

[17] Zhang, Y.: The unified image encryption algorithm based on chaos and cubic S-Box. Inf. Sci. (Ny). 450, 361–377 (2018). https://doi.org/10.1016/j.ins.2018.03.055

[18] Qobbi, Y., Jarjar, A., Essaid, M., Benazzi, A.: Image encryption algorithm using dynamic permutation and large chaotic S-box. Multimedia. Tools Appl. (2022). https://doi.org/10.1007/s11042-022-14175-2

[19] Hraoui, S., Gmira, F., Abbou, M.F., Oulidi, A.J., Jarjar, A.: A New Cryptosystem of Color Image Using a Dynamic-Chaos Hill Cipher Algorithm. Procedia Computer. Sci. 148, 399–408 (2019). https://doi.org/10.1016/j.procs.2019.01.048