The Journal of Applied Sciences Research

Journal homepage: http://www.journals.wsrpublishing.com/index.php/tjasr

Online ISSN: 2383-2215 Print ISSN: 2423-4400

Original Article

New cryptosystem using Vigenere and multiple bit-level permutations

Abdellatif JarJar
Moulay Rachid High School, Taza, IJCAT-326629, Morocco

Abstract

The largest part of the image encryption algorithms operates on the pixel as a central element by implementing diffusion confusion and eventually a permutation. On the other side, a permutation applied at the bit level changes not only the pixel value, but also its location within the image. In this work, we will propose a new technology for medical and color image encryption, which starts with an application of an improved Vigenere method acting at the pixel level to overcome any differential attack, followed by an implementation of several chaotic permutations acting at the bit level. Simulations performed on a large number of images of different sizes and formats ensure that our method is not subject to any known attacks.

Article Highlights

This new technology which applies Vigenere trick for protection against any differential attack followed by several permutations acting at bit level for color image encryption is based on the following novelties

1. New sequence chaotic design
2. Implementation of Vigenere's advanced technique
3. Switching to binary writing
4. Generation of chaotic permutations
5. New method setup

ARTICLE INFO

Corresponding Author: Abdellatif JarJar < abdoujjar@gmail.com >

How to Cite this Article: JarJar, A. (2021). New cryptosystem using Vigenere and multiple bit-level permutations. *The Journal of Applied Sciences Research,* 8(1), 28-50.

Article History: Received: 2021-04-26 Accepted: 2021-06-146

$$Notation$$

$$G_t = {}^{\mathbb{Z}}/_{t\mathbb{Z}}\ Ring$$
$$A(:j): Columm\ j\ of\ matrix\ A$$
$$A(j;): Line\ j\ of\ matrix\ A$$
$$\oplus: Xor\ operator$$
$$E(x): The\ whole\ part\ of\ x$$

Keywords: Vigenere grid; Chaotic map; Broadcast function; Chaotic permutation;

I.Introduction

The advancement of number theory in mathematics provides researchers with a new opportunity to generate new algorithms for image encryption based on chaos. Other researchers are happy to improve some classic techniques by injecting chaos. We cited some improvements acting at de pixel level from Vigenere, Hill and Feistel. [1 − 2 − 3 − 4].

1) Vigenere's classical technique

This technology is based on static (V)matrix defined by the following algorithm

$$algorithm1 \begin{cases} Fist\ Row \\ For\ \ i = 1\ \ to\ 26 \\ V(1,i) = i \\ Next\ i \\ folloying\ Rows \\ For\ \ i = 2\ \ to\ 26 \\ For\ \ j = 1\ \ to\ 26 \\ V(i,j) = V(i-1,(j+1),26) \\ Next\ j,i \end{cases}$$

Let (P): plain text, (C): cypher text; (K): Encryption key, (V) Vigenere matrix and (l): length of clear text. So

$$equation1 \begin{cases} C_i = V(P_i,K_i) = (P_i + K_i) \mod 26 \\ P_i = V(C_i,K_i) = (P_i - K_i) \mod 26 \end{cases}$$

The classic Vigenere technology is supported by a very public substitution matrix and a private key copied to the size of the plaintext. Until the arrival of ***Babagh***, the classic version of this method has withstood many tests. Some attempts to improve the method have been developed, but the same common matrix is always used. [5 − 6 − 7 − 8].

1) Our contribution

Our contribution in this work is to start with a deeply improved trick of Vigenere for a suppression of any differential attack and to switch to binary notation to apply multiple permutations to encrypt a color image.

I.The proposed method

Based on chaos $[9-10--11-12]$, This new cryptosystem is based on the following axes

Axe 1: *Chaotic Sequences Development*

The paper adopts a new two-dimensional chaotic map most $[13-14-15]$commonly used in the image encryption domain. This choice is based on the large size of the secret key and the high sensitivity to initial conditions on the one hand, and on the other hand on the simplicity of its configuration in the system.

1) 2D Logistics Map Decryption

It is a two-dimensional map $[16-17-18]$ defined by a second-degree polynomial. The installation of the chaos is described by the initial conditions. The equation below describes the expression of the 2D logistic map

$$Eq1 \quad \begin{cases} x_0, y_0 \ \epsilon]0 \quad 1[\begin{cases} x_{n+1} = \mu_1 x_n (1-x_n) + \mu_2 y_n^2 \\ y_{n+1} = \mu_3 y_n (1-y_n) + \mu_4 (x_n^2 + x_n y_n) \end{cases} \\ \mu_1 \epsilon [2{,}75 \quad 3{,}4] \\ \mu_2 \epsilon [0;15 \quad 0{,}21] \\ \mu_3 \epsilon [2{,}75 \quad 3{,}45] \\ \mu_4 \epsilon [0{,}13 \quad 0{,}15] \end{cases}$$

2) Chaotic vector design

Our algorithm requires the construction of chaotic vectors determined by the algorithm below

Control and chaotic vector creation

$$for \ \ i = 1 \ to 24nm$$

$$CL(i) = \ mod\left(E\left(\frac{x(i)+2y(i)}{3} * 10^{11}, 254\right) + 1\right) \qquad [2]$$

$$KL(i) = \ mod\left(E\left(\frac{x(i)+y(i)*y(i)}{2} * 10^{11}, 253\right) + 2\right)$$

$$KR(i) = E\left(\frac{KL(i)+CL(i)}{2}\right)$$

3) Binary control vector development.

For the supervision and monitoring of our algorithm, two binary vectors are created by the following algorithm

Control vector

$$\begin{array}{l} for\ \ i = 1\ to\ 12nm \\ if\ x(i) \geq y(i)\ then \\ BR(i) = 0\ Else \\ BR(i) = 1 \\ End\ if \end{array} \qquad \begin{array}{c} if\ CL(i) \geq KL(i) then \\ CR(i) = 0\ else\ CR(i) = 1 \\ end\ if \\ Next\ i \end{array} \qquad [3]$$

***Axe 2**: Vigenere upgrade*

In order to overcome any differential attack, we start our encryption process by applying the improved Vigenere technique $[19 - 20 - 21]$.

1) Original Image Vectorization

After the three (RGB) color channels extraction and their conversion into size vectors $(\boldsymbol{Vr}), (\boldsymbol{Vg}), (\boldsymbol{Vb})$ $(\boldsymbol{1, nm})$ each, a concatenation is established to generate a vector $\boldsymbol{X(x_1, x_2, \ldots\ldots, x_{3nm})}$ of size $(1,3nm)$. This operation is described by the following algorithm,

Original Image Vectorization

$$\begin{array}{l} For\ i = 1\ to\ \ nm \\ X(3i-2) = V_r(i) \\ X(3i-1) = V_g(i) \\ X(3i) = V_b(i) \\ Next\ i \end{array} \qquad [4]$$

a) Initialization value computation

The initialization value must be calculated in order to change the startup pixel in the future and start the encryption process correctly. This value depends closely on the original image and the control vector (CV). The calculation of this value is described below

Initialization value computation

$$\begin{array}{l} for\ i = 2\ to\ 3nm \\ If\ CR(i) = 0\ Then \\ V = IV \oplus X(i) \oplus CL(i) \\ Else \\ IV = IV \oplus X(i) \oplus KL(i) \\ End\ if \\ Next\ i \end{array} \qquad [5]$$

The chaotic vector participates in the calculation of the initialization value mainly to avoid the problem of uniform image color.

2) Vigenere's advanced methods

This technique requires the establishment of two substitution matrices (VG) and (VD) through the process described by the following steps

1. permutation (RP) obtained by descending ordering the first 256 *values* of the sequence (CL)
2. permutation (RR) obtained by increasing the ordering the first 256 *values* of the sequence (KL),

with the following restrictions

Lodging

$$If\ PR(i) = 256\ Then\ PR(i) = 0 \quad (3)$$
$$If\ RR(i) = 256\ Then\ RR(i) = 0$$

This new construction is entirely supervised by the vector (CR). It is given by the following algorithm

$$algorithm6 \begin{cases} Fist\ Row \\ For\ i = 1\ to\ 256 \\ VG(1,i) = RP(i) \\ VD(1,i) = RR(i) \\ Next\ i \end{cases} \begin{cases} For\ i = 2\ to\ 256 \\ For\ j = 1\ to\ 256 \\ if\ CR(i) = 1\ then \\ VG(i,j) = VG\left(i-1, RP\left(mod\left(j + CL(i)\right), 256\right)\right) \\ VD(i,j) = VD\left(i-1, RR\left(mod\left(j + KL(i)\right), 256\right)\right) \\ else \\ VG(i,j) = VG\left(i-1, RP\left(mod\left(j + KL(i)\right), 256\right)\right) \\ VD(i,j) = VD\left(i-1, RR\left(mod\left(j + CL(i)\right), 256\right)\right) \\ end\ if \\ next\ j,i \end{cases}$$

note that this building is completely done under the control of the vector (CR).

Example: in (G_8)

(VG)	1	2	3	4	5	6	7	0	CR	KL	CL
1	3	5	0	6	2	7	1	4			
2	2	7	1	4	3	5	0	6	1	5	4
3	4	3	5	0	6	2	7	1	1	3	5
4	2	7	1	4	3	5	0	6	0	3	4
5	0		2	7	1	4	3	5	1	4	2

(VD)	1	2	3	4	5	6	7	0
1	0	4	5	7	1	2	6	3
2	7	1	2	3	3	0	4	5
3	0	4	5	7	1	2	6	3
4	1	2	6	3	0	4	5	7
5	0	4	5	7	1	2	6	3

a) New Vigenere's mathematical expression

Based on Vigenere's classic formula, given by the following formula

Vigenere Expression

$$Y(i) = VG(CL(i), X(i)) \quad (4)$$

In this work the image $Y(i)$ of pixel $X(i)$ is given by the formula below

New Vigenere Expression

$$If\ BR(i) = 0\ Then$$
$$V_1(X(i)) = Y(i) = VG(CL(i), VD(KL(i), X(i) \oplus KR(i)))$$
$$Else$$
$$V_2(X(i)) = Y(i) = VD(KL(i), VG(KR(i), X(i) \oplus CL(i)))$$

[7]

This new expression is closely related to the control vector (BR)

2) First Encryption Process

The figure below describes in detail the new technique of Vigenere acting at the pixel level

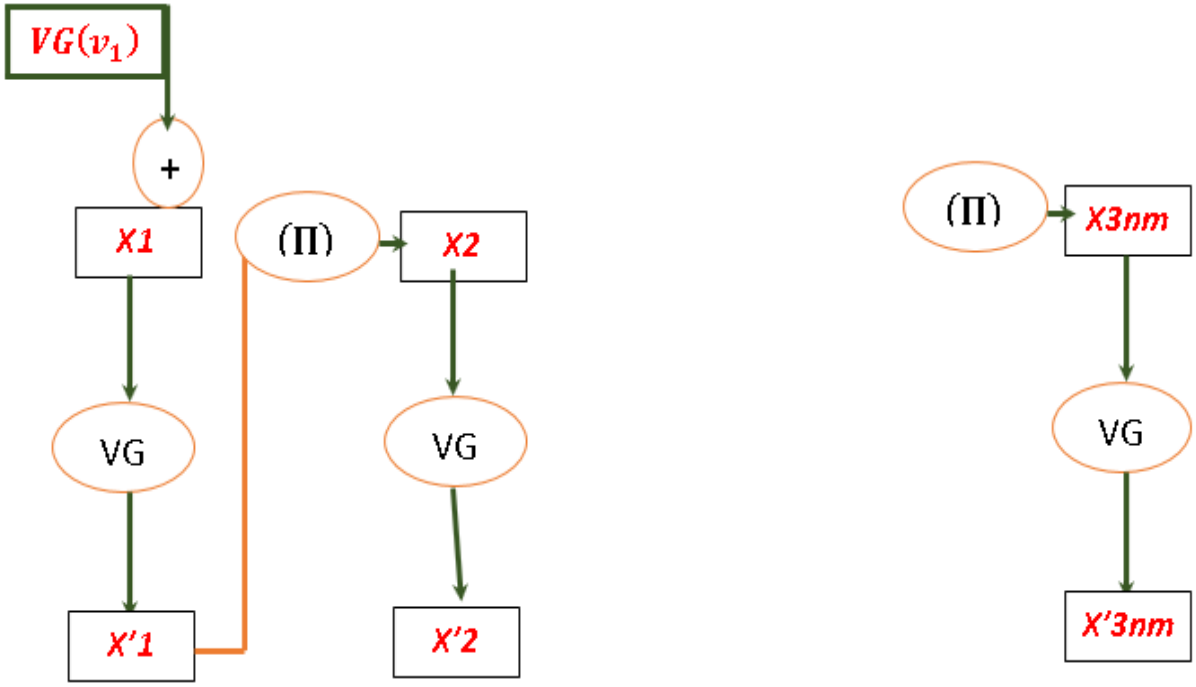


Figure2 : First Encryption

Encryption functions

(V): advanced substitution matrix (5)
(Π): New diffusion function

(Π) The new diffusion, defined by the following equation

Diffusion function

$$\Pi(X(i+1)) = VD(KR(i), VG(CL(i), X'(i) \oplus X(i+1))) \quad (6)$$

This function links the encrypted pixel with the next clear pixel. This schema is translated by the algorithm below

First process algorithm

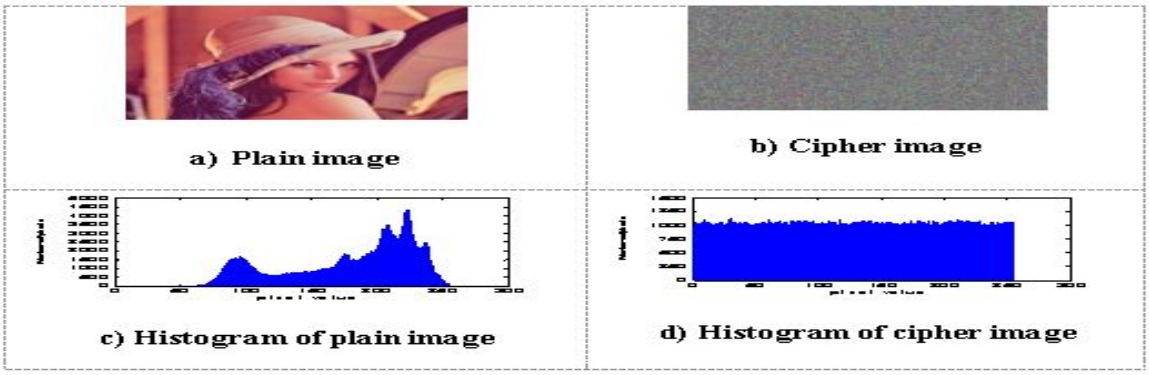
$$X'(1) = VG\big(KR(1), VD(CL(1), IV \oplus X(1))\big)$$
$$For\ i) 2\ to\ 3nm$$
$$\Pi\big(X(i)\big) = VD\left(KR(i), VG\big(CL(i), X'(i-1) \oplus X(i)\big)\right)$$
$$If\ BR(i) = 0\ Then \qquad [8]$$
$$X'(i) = V_1\left(\Pi\big(X(i)\big)\right)$$
$$Else$$
$$X'(i) = V_2\left(\Pi\big(X(i)\big)\right)$$
$$End\ if$$
$$Next\ i$$

We note that this first step uses only substitutions, which ensures an extreme speed in the execution. The output vector $\boldsymbol{X'(x'_1, x'_2, \ldots\ldots, x'_{3nm})}$, 'will undergo a second encryption attempt.

1) First round analysis

Several images have been tested by this first round of improved Vigenere, we quote

Example: "***Lena" encrypted in the first round***



Axis 4: Permutation generation

1) Binary Writing

The vector (X') is converted into binary to obtain a matrix (XM) of size $(8.3nm)$. This process follows the following path

$$\begin{cases} \text{Switching to hexadecimal} \\ x = E\left(\frac{X(i)}{16}\right) \\ y = X(i) - 16 * x \end{cases} \qquad \begin{cases} \text{Passage in } G_4 \\ \alpha = E\left(\frac{x}{4}\right) \\ \beta = x - 4\alpha \\ \gamma = E\left(\frac{x}{4}\right) \\ \delta = y - 4\gamma \end{cases}$$

We consider the table (T) of binary values lower than 4

(T)	1	2
0	0	0
1	0	1
2	1	0
3	1	1

Finally, the passage of the vector (X') in binary matrix (XM) of size $(8,3nm)$ is defined by the following algorithm

Binary Writing

$$\begin{aligned} &For\ i = 1\ to\ 3nm \\ &\quad x = E\left(\frac{X'(i)}{16}\right) \\ &\quad y = X'(i) - 16 * x \\ &\quad \alpha = E\left(\frac{x}{4}\right) \\ &\quad \beta = x - 4 * \alpha \\ &\quad \gamma = E\left(\frac{y}{4}\right) \\ &\quad \delta = y - 4 * \gamma \end{aligned} \qquad \begin{aligned} &XM(1,i) = T(\alpha,1) \\ &XM(2,i) = T(\alpha,2) \\ &XM(3,i) = T(\beta,1) \\ &XM(4,i) = T(\beta,2) \\ &XM(5,i) = T(\gamma,1) \\ &XM(6,i) = T(\gamma,2) \\ &XM(7,i) = T(\gamma,1) \\ &XM(8,i) = T(\gamma,2) \\ &Next\ i \end{aligned} \quad [9]$$

This passage in binary notation is determined by the following algorithm

Binary transition

$$\begin{aligned} &For\ i = 1\ to\ 8 \\ &\quad For\ j = 1\ to\ 3nm \\ &MX(i,j) = XM\left(i, Q_i(j)\right) \\ &\quad Next\ j,i \end{aligned} \quad [10]$$

2) The Two P-Box Construction

Two permutation matrices will be constructed. The matrix (PL) of size $(8,3nm)$ will be used for the permutation of rows, while the matrix (PC) of size $(3nm,8)$ will be used for the permutation of columns.

a) Construction of (PL)

The construction of the first P-Box requires the generation of three permutations in (G_{64})

1. The first row is the permutation $(P1)$ obtained by a descending sort in the broad sense on the first 3nm values of the vector (CL).
2. The second line is the permutation $(P2)$obtained by a descending sort in the broad sense on the first $(3nm)$ values of the vector (KL)
3. The third row is the permutation $(P3)$ obtained by a descending sort on the first 3nm values of the vector (U).

The determination of the $PL(i:)$ line $for\ i > 3$ by the following equation

$$(PL)\ design$$

$$If\ CV(i) = 0\ Then$$
$$PL(i:) = PL(i-1:)oPL(i-2:)$$
$$Else$$
$$PL(i:) = PL(i-2:)oPL(i-3:) \qquad [11]$$

b) Construction of (PC)

The construction of the second P-Box requires the generation of three permutations in (G_{64})

1. The first Column is the permutation $(C1)$ obtained by a descending sort in the broad sense on the first 8 values of the vector (CL).
2. The second Column is the permutation $(C2)$obtained by a descending sort in the broad sense on the first (8) values of the vector (KL)
3. The third Column is the permutation (8) obtained by a descending sort on the first 3nm values of the vector (V).

The determination of the $PL(:j)$ line $for\ j > 3$ by the following equation

$$(PC)\ design$$

$$If\ BR(i) = 0\ Then$$
$$PC(:j) = PC(:j-1)oPL(:j-2)$$
$$Else$$
$$PC(:j) = PC(:j-2)oPL(:j-3) \qquad [12]$$

First, we swap all the rows and then we swap the columns.

Example

Line swapping

(X)	$X(1)$	$X(2)$	$X(3)$	$X(4)$	$X(5)$	$X(6)$
Value	***209***	***42***	***140***	***226***	***49***	***101***

The vector (X) will be transited in binary by application of the table (NC). a build up of the two permutation tables is established. Firstly, all the rows will be permuted by the primary P-Box and then all the columns will be permuted by the secondary P-Box. This is illustrated by the following diagram

(PL) Permutations

	(Q_1)	(Q_2)	(Q_3)	(Q_4)	(Q_5)	(Q_6)	(Q_7)	(Q_8)
1	3	5	3	2	3	5	4	6
2	2	3	2	3	1	6	6	1
3	5	6	4	5	5	2	3	3
4	1	4	6	4	6	1	5	5
5	4	1	5	1	4	4	1	2
6	6	2	1	6	2	3	2	4

Clear Pixel

Cypher Pixel

	$X(1)$	$X(2)$	$X(3)$	$X(4)$	$X(5)$	$X(6)$
	209	42	140	226	49	101
1	1	0	1	1	0	0
2	1	0	0	1	0	1
3	0	1	0	1	1	1
4	1	0	0	0	1	0
5	0	1	1	0	0	0
6	0	1	1	0	0	1
7	1	1	0	1	0	0
8	0	0	1	1	1	1

	$Y(1)$	$Y(2)$	$Y(3)$	$Y(4)$	$Y(5)$	$Y(6)$
1	1	0	0	1	1	0
2	0	0	1	1	1	0
3	0	1	1	1	1	0
4	0	0	1	0	1	0
5	1	0	0	0	0	1
6	0	1	1	0	0	1
7	1	0	0	0	1	1
8	1	0	1	1	0	1
	130	36	117	224	141	15

Column swapping

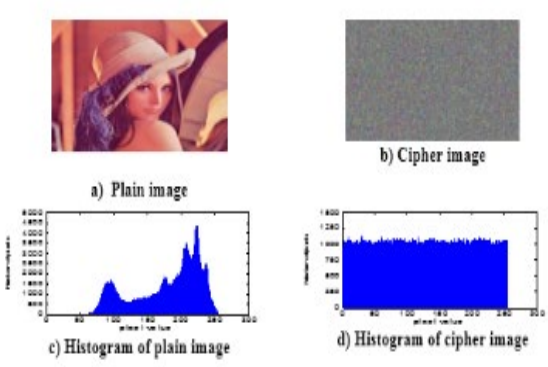
(PC)

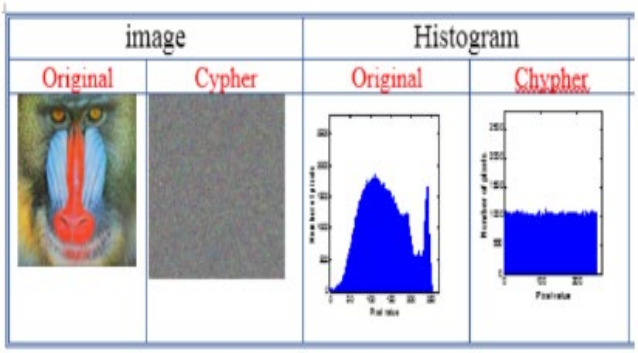
	(C_1)	(C)	(C_3)	(C_4)	(C_5)	(C_6)
1	4	3	5	6	2	6
2	5	2	2	3	3	1
3	3	8	8	5	8	8
4	7	5	1	2	7	5
5	2	6	6	8	6	7
6	6	1	4	4	2	4
7	8	7	7	7	1	2
8	1	4	3	1	5	3

	$Y(1)$	$Y(2)$	$Y(3)$	$Y(4)$	$Y(5)$	$Y(6)$
1	1	0	0	1	1	0
2	0	0	1	1	1	0
3	0	1	1	1	1	0
4	0	0	1	0	1	0
5	1	0	0	0	0	1
6	0	1	1	0	0	1
7	1	0	0	0	1	1
8	1	0	1	1	0	1
	130	**36**	**117**	**224**	**141**	**15**

	$Z(1)$	$Z2)$	$Z(3)$	$Z(4)$	$Z(5)$	$Z(6)$
1	0	1	0	0	1	1
2	1	0	1	1	1	0
3	0	0	1	0	0	1
4	1	0	0	1	1	1
5	0	1	1	1	0	1
6	0	0	1	0	1	0
7	1	0	0	0	1	0
8	1	0	1	1	0	0
	84	**136**	**207**	**89**	**206**	**184**

Example:





Step 5: Decryption encrypted images

Our approach is a symmetric encryption system with broadcast implementation. Therefore, in the decryption process, we apply the inverse encryption functions starting with the last block. The decryption process is based on the following steps

1. *Binary conversion*
2. *Reciprocal permutation generation*
3. *Switching to grayscale*
4. *Vigenere reciprocal matrix*
5. *Vigenere's reciprocal application*

1) Reciprocal permutation generation

Consider $(T_i)_1^8$the reciprocal permutation of $(Q_i)_1^8$, it defines by the following algorithm

Reciprocal permutation

$$For\ j = 1\ to\ 3nm$$
$$T_i\big(Q_i(j)\big) = j \qquad [13]$$
$$Next\ j$$

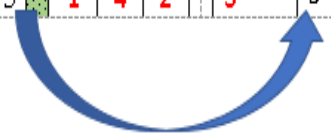
2) Vigenere Reciprocal matrix

Vigenere's reciprocal matrix

$$for\ \ i = 1\ \ to\ \ 256$$
$$for\ j = 1\ to\ \ 256$$
$$GV\big(i, VG(i,j)\big) = j \qquad [14]$$
$$DV\big(\big(i, VD(i,j)\big) = j$$
$$Next\ j, i$$

(VG)	1	2	3	4	5	6	7	0	CR	KL	CL
1	3	5	0	6	2	7	1	4			
2	2	7	1	4	3	5	0	6	1	5	4
3	4	3	5	0	6	2	7	1	1	3	5
4	2	7	1	4	3	5	0	6	0	3	4
5	0		2	7	1	4	3	5	1	4	2

(GV)	1	2	3	4	5	6	7	0
1	7	5	1	0	2	4	6	3
2	3	1	5	4	6	0	2	7
3	0	6	2	1	3	5	7	4
4	3	1	5	4	6	0	2	7
5	5	3	7	6	0	2	4	1



By following the same logic of Vigenere's traditional technique, we obtain

Vigenere Reciprocal

$$if\ \ z = VG(y,x)$$
$$Then \qquad [15]$$
$$x = GV(y,z)$$

3) Vigenere's inverse expression

New Vigenere Reciprocal

$$if\ BR(k) = 0\ then$$
$$V_1^{-1}(X'_k) = DV\big(KL(k), GV(CL(k), X'_k)\big) \oplus KR(k)$$
$$else$$
$$V_2^{-1}(X'_k) == GV\big(KR(k), DV(KL(k), X'_k)\big) \oplus CL(k) \qquad [16]$$
$$end\ if$$

4) Reverse diffusion

$$\Pi^{-1}(X'(k)) = \mathrm{GV}\big(CL(k), DV(KR(k), X'(\mathrm{k})\big) \oplus X(k-1) \qquad (7)$$

Axis 5: Examples and simulations

In this section we will show the performance of our technology compared to other algorithms.

1) Brutal assaults

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

a. Key-space analysis

The $2D\ logic\ map$ used alone in this work is attached to two initial values and four control parameters, which greatly increases the size of the private encryption key and provides strong protection against any brute force attacks.

Secrete Key used

$$x_0 = 0{,}7655412001\ , \mu_1 = 3.89541$$
$$y_0 = 0.865421331,\ \ \mu_2 = 0{,}56120 \qquad (8)$$
$$\mu_3 = 1{,}3561 \qquad \mu_4 = 0{,}56321$$

If we use single-precision real numbers $\mathbf{10^{-10}}$ to operate, the total size of the key will greatly exceed $\approx \mathbf{2^{180}} \gg \mathbf{2^{110}}$, which is enough to avoid any brutal attacks.

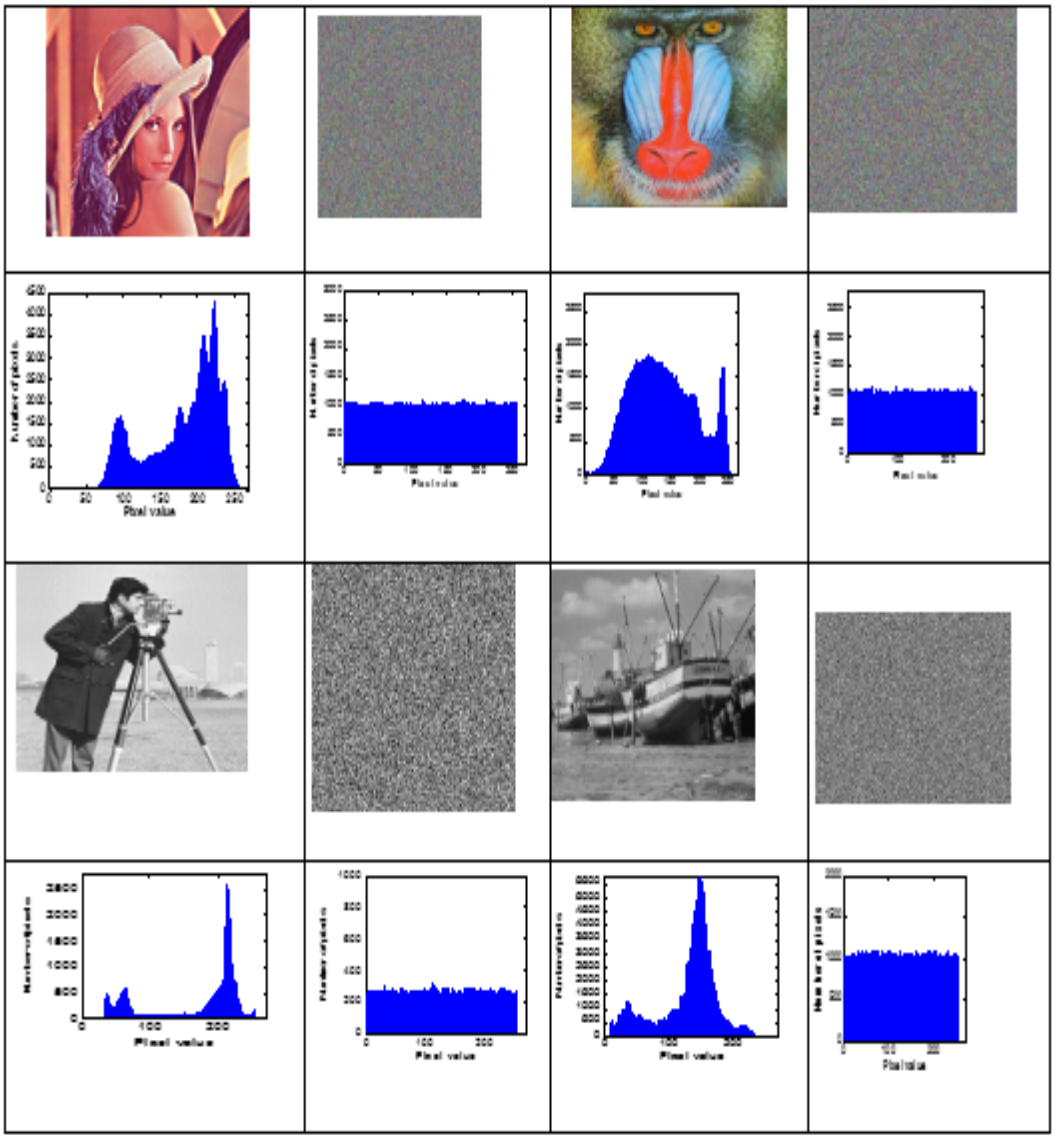
b. Secret key's sensitivity Analysis

Our map is extremely sensitive to initial conditions and control parameters, ensuring that minimal disturbance to at least one parameter of the key will generate another chaotic sequence.

2) Histogram analysis

The application of a cryptographic system on a large number of randomly selected images to extract all the statistical constants reveals the robustness of the technology used. In our case, randomly selecting 150 images from a database of more than 6000 images produced the following results. An example of the reference images chosen and tested by our algorithm

Table 1: Encrypted image histogram



We noticed that all images encrypted by our system have a uniform and flat histogram, which can resist any histogram attack.

3) Statistics Attack Security

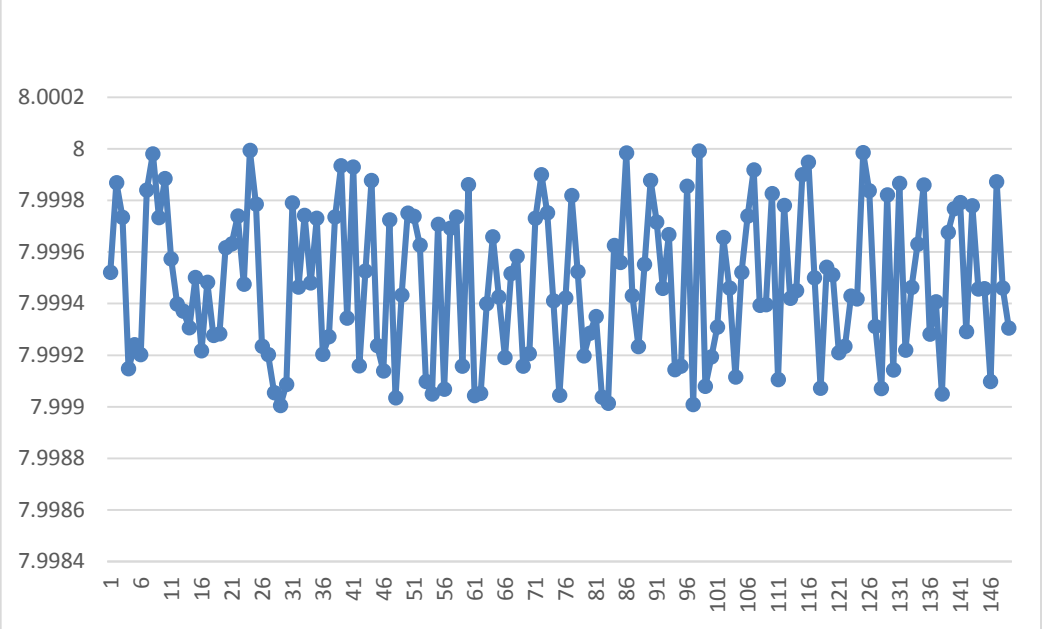
a. Entropy Analysis

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

Entropy

$$H(MC) = \frac{1}{t}\sum_{i=1}^{t} -p(i)\, log_2(p(i)) \qquad (9)$$

The entropy measured after encryption of the 150 images by our algorithm, describe the following curve



Fugure3: Entropy of 150 images

We note that all the cleared values revolve around the maximum value 8, which proves the uniformity of the histograms and ensures the protection against any attack by entropy.

b. Correlation analysis

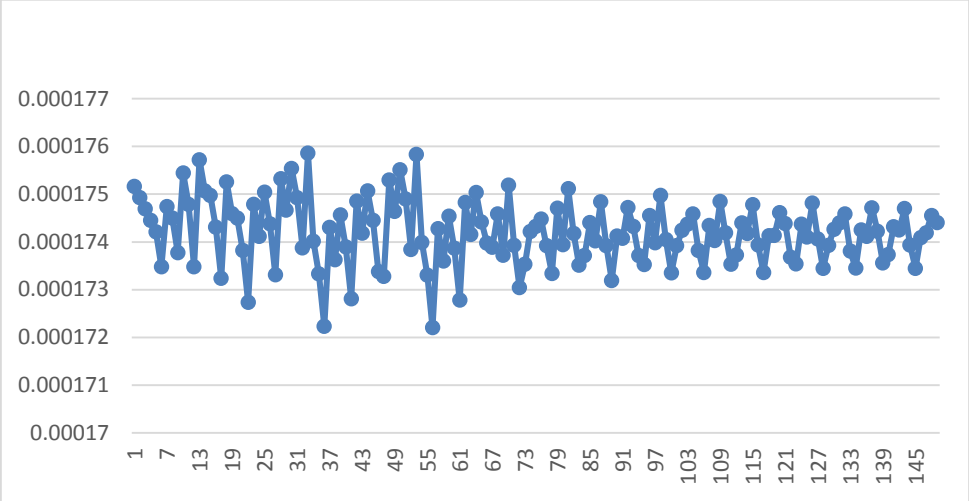
Correlation is a technique that compares two images to estimate the displacement of pixels in one image relative to another reference image. The relevant expression is defined by the following equation

Correlation

$$r = \frac{cov(x,y)}{\sqrt{V(x)}\sqrt{V(y)}} \qquad (10)$$

i.Horizontal Correlation

Another 150 images were randomly selected from the same database to record the horizontal correlation values represented in the curve in the following figure The following figure



Fugure4: Entropy of 150 images

ii.Vertical Correlation

The curve in the following figure shows the vertical correlation values of the same images

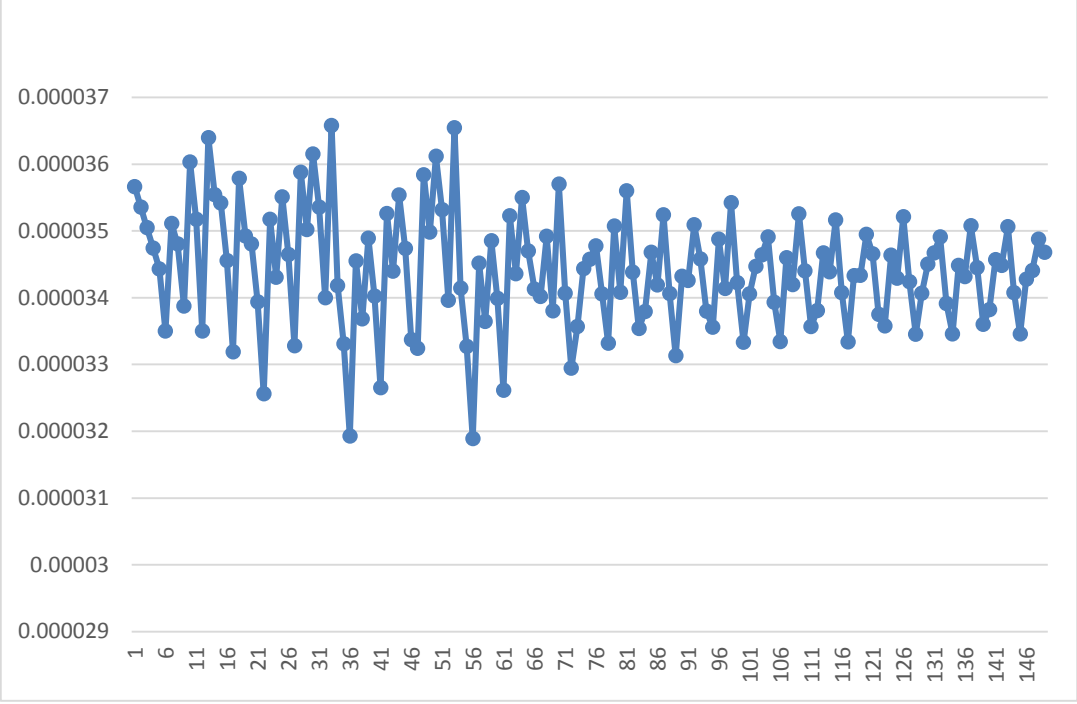


fIgure5: Vertical correlation of 150 images

Figure 5 shows that the diagonal correlation values of the encrypted images are close to zero. This ensures high security against correlation attacks.

iii.Diagonal Correlation

The curve in the following figure shows the vertical correlation values of the same images

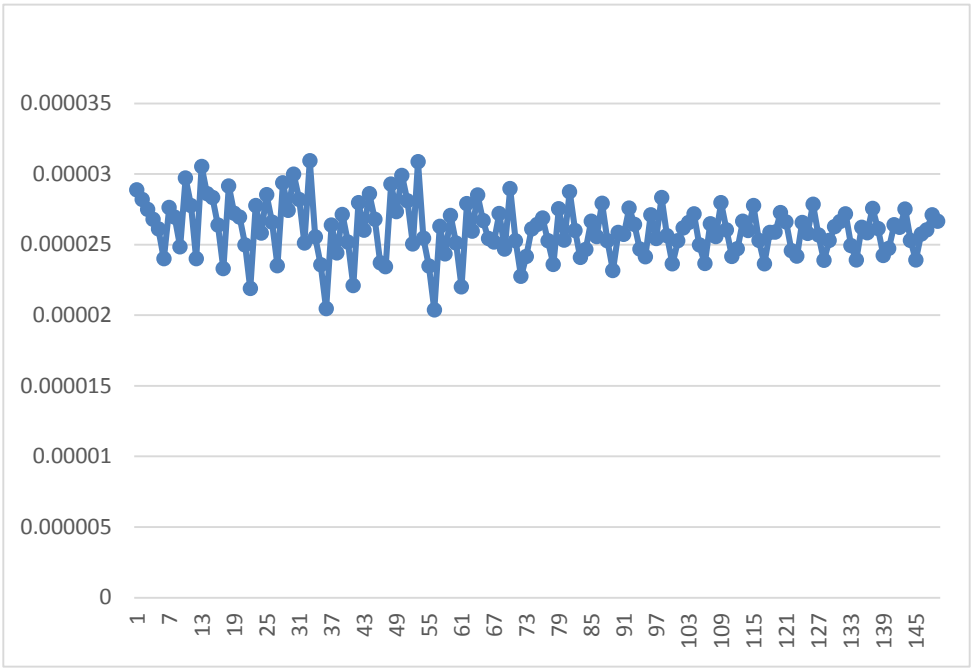


fIgure6: Diagonal correlation of 150 images

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

4) Differential analysis

If we change one bit in an image and compare the difference between the two encrypted images, we get the following curve on 150 images otherwise chosen. Let$(\boldsymbol{C_1})$and$(\boldsymbol{C_2})$, be the two images whose difference is one bit only

$$NPCR \left(\frac{1}{nm}\sum_{i,j=1}^{nm} D(i,j)\right) * 100 \quad (12)$$

$$With \quad D(i,j) = \begin{cases} 1 & if \quad C_1(i,j) \neq C_2(i,j) \\ 0 & if \quad C_1(i,j) = C_2(i,j) \end{cases} \quad (13)$$

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

$$UACI \left(\frac{1}{nm} \sum_{i,j=1}^{nm} Abs\big(C_1(i,j) - C_2(i,j)\big) \right) * 100 \quad (14)$$

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

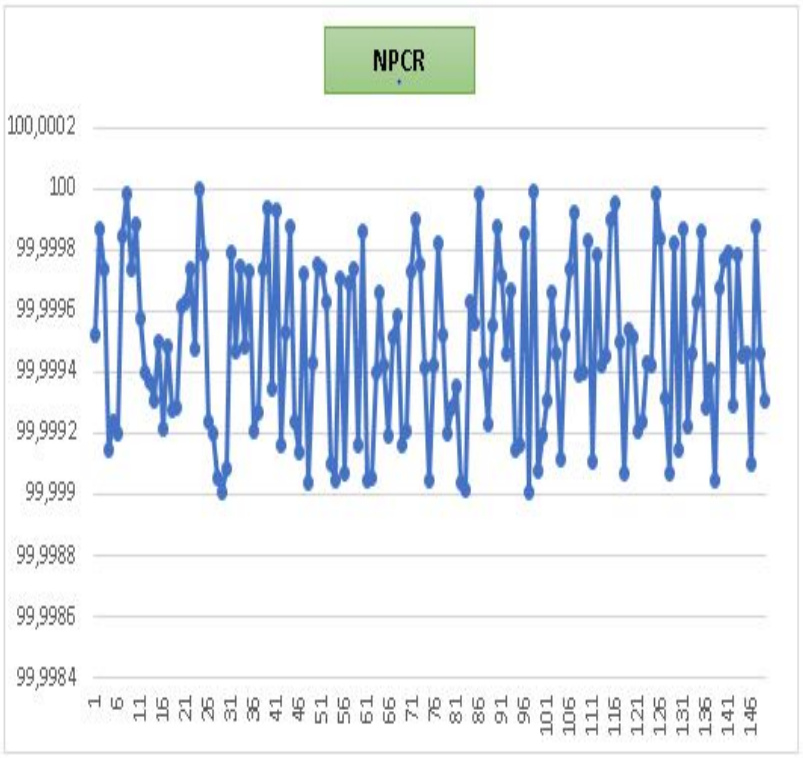


fIgure7: NPCR of 150 images

The $(NPCR)$ values found are in the range [99.63 99.95], which provides strong protection against any differential attack.

For the calculation of the statistical constant $(UACI)$, the test performed on the same 150 images is illustrated by the curve in the following figure

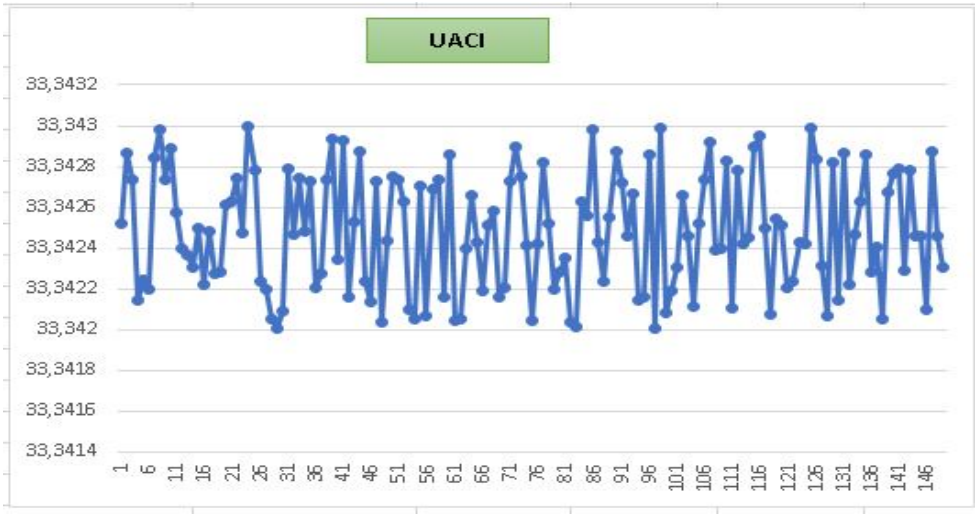


fIgure8: UACI of 150 images

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.

5) Avalanche effect

The diffusion function installed in the first encryption process using Vigenere's advanced technique ensures a better increase in the impact of the avalanche effect and gives strong protection against any differential attack

$$AE \qquad \left(\frac{\sum_i bit\ change}{\sum_i bit\ total}\right) * 100 \qquad (15)$$

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

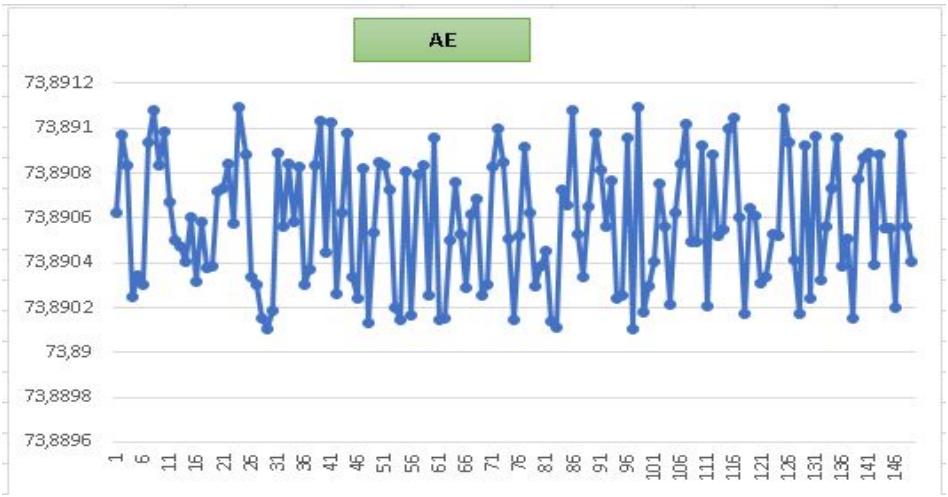


Figure9: Avalanche effect

All values returned from the (AE) by our method are all in the range of residual values [73,89 73,892]. 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.

6) Signal-To-Peak Noise Ratio (PSNR)

(a) 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

$$MSE \qquad \sum_{i,j}(P(i,j)-C(i,j))^2 \qquad (16)$$

$(P(i,j))$; pixel of the clear image

$(C(i,j))$: pixel of the cypher image

(b) PSNR

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

$$PSNR \qquad 20Log_{10}\left(\frac{I_{max}}{\sqrt{MSE}}\right) \qquad (17)$$

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.

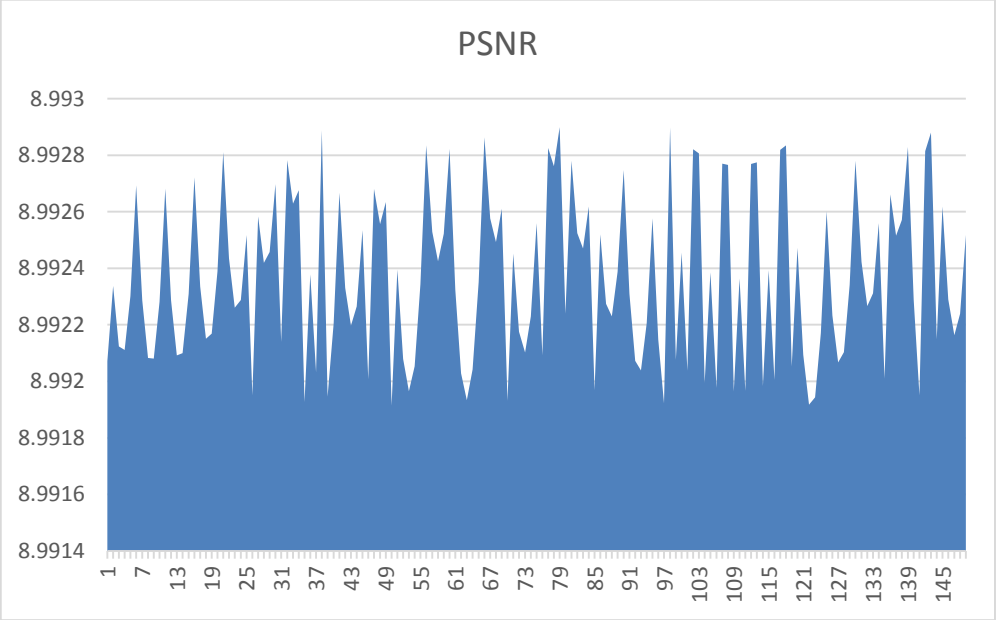


fIgure10: ¨PSNR of 150 images

All values returned from the $(PSNR)$ by our method are all in the range of residual values [8,99 8,993].

b) Speed analysis

For an evaluation of the execution time, our algorithm is tested on a personal computer "Intel core $i5\ 3337\ U\ 1.86\ Gz\ CPU\ 8\ GB\ ram$. We use Matlab as programming software. We measure the encryption and decryption time of the tested images.

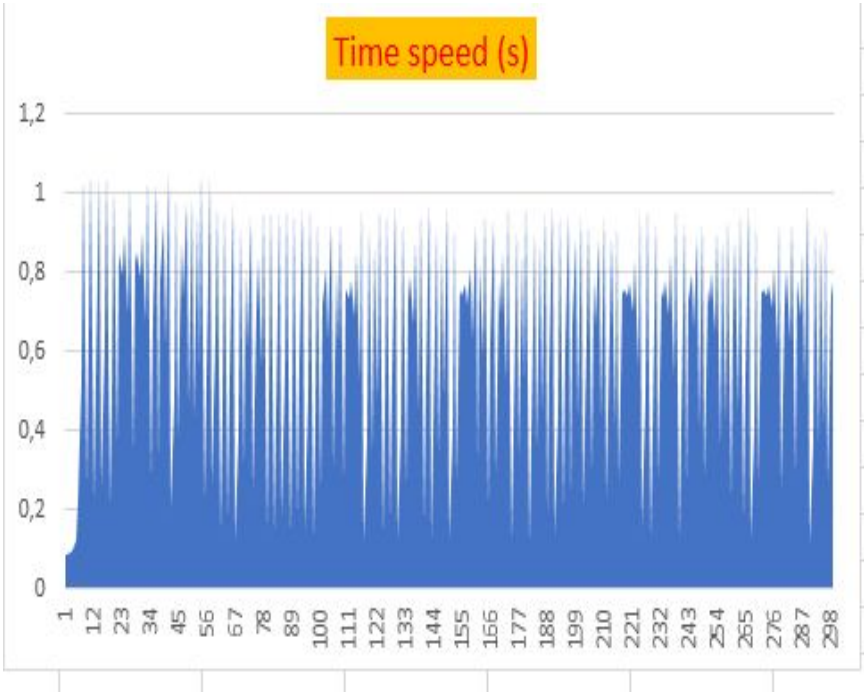


fIgure11: time of 300 images

II.Math Security

The size of the encryption key protects the system from brute force attacks. The construction of the chaotic permutations is difficult to reverse. Moreover, the established chaining protects the algorithm from differential attacks.

III.Conclusion

The beginning of this new algorithm is the implementation of the deeply improved Vigenere technology, which is ensured by two chaotic replacement matrices and additional new permutation functions, plus two Vigenere matrices to ensure diffusion, and then in the binary code as on the line Then apply several chaotic permutation applications at the bit level of the column, which provides our system with strong complexity and better protects the system from any known attacks.

REFERENCES

Al-Khalid, A. S., & Al-Khfagi, A. O. (2015, September). Cryptanalysis of a Hill cipher using genetic algorithm. In *2015 World Symposium on Computer Networks and Information Security (WSCNIS)* (pp. 1-4). IEEE.

Anwar, S., & Meghana, S. (2019). A pixel permutation based image encryption technique using chaotic map. *Multimedia tools and applications*, *78*(19), 27569-27590.

Bansal, R., Gupta, S., & Sharma, G. (2017). An innovative image encryption scheme based on chaotic map and Vigenère scheme. *Multimedia Tools and Applications*, *76*(15), 16529-16562.

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.

Essaid, M., Akharraz, I., & Saaidi, A. (2019). Image encryption scheme based on a new secure variant of Hill cipher and 1D chaotic maps. *Journal of Information Security and Applications*, *47*, 173-187.

IEEE Computer Society. (1985) IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std, New York.

Jarjar, A. (2017). Improvement of hill's classical method in image cryptography. *International Journal of Statistics and Applied Mathematics*, *2*(3 Part A).

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

Li, H., Wang, Y., & Zuo, Z. (2019). Chaos-based image encryption algorithm with orbit perturbation and dynamic state variable selection mechanisms. *Optics and Lasers in Engineering*, *115*, 197-207

Li, P., Li, Z., Halang, W. A., & Chen, G. (2006). A multiple pseudorandom-bit generator based on a spatiotemporal chaotic map. *Physics Letters A*, *349*(6), 467-473.

Lin, C. H., Lee, C. Y., & Lee, C. Y. (2004). Comments on Saeednia's improved scheme for the Hill cipher. *Journal of the Chinese institute of engineers*, *27*(5), 743-746.

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.

Overbey, J., Traves, W., & Wojdylo, J. (2005). On the keyspace of the Hill cipher. *Cryptologia*, 29(1), 59-72.

Pareek, N. K., Patidar, V., & Sud, K. K. (2006). Image encryption using chaotic logistic map. *Image and vision computing*, *24*(9), 926-934.

Rachmawanto, E. H., De Rosal, I. M. S., Sari, C. A., Santoso, H. A., Rafrastara, F. A., & Sugiarto, E. (2019, July). Block-based arnold chaotic map for image encryption. In *2019 International Conference on Information and Communications Technology (ICOIACT)* (pp. 174-178). IEEE.

Reddy, V. V. K., & Bhukya, S. (2018). Encrypt and decrypt image using vigenere cipher. *International Journal of Pure and Applied Mathematics*, *118*(24).

Saeednia, S. (2000). How to make the Hill cipher secure. *Cryptologia*, *24*(4), 353-360.

Saputra, I., Hasibuan, N. A., & Rahim, R. (2017). Vigenere cipher algorithm with grayscale image key generator for secure text file. *International Journal of Engineering Research & Technology (IJERT)*, *6*(1), 266-269

Wang, Y., Wong, K. W., Liao, X., Xiang, T., & Chen, G. (2009). A chaos-based image encryption algorithm with variable control parameters. *Chaos, Solitons & Fractals*, *41*(4), 1773-1783.

Ye, R. (2011). A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism. *Optics Communications*, *284*(22), 5290-5298.