doi:10.20944/preprints202104.0647.v1

VIGENERE FOLLOWED BY MULTIPLE BIT-LEVEL PERMUTATIONS TO ENCRYPT A COLOR IMAGE

Abdellatif JarJar

Moulay Rachid High School, Taza Morocco **IJCAT-326629**

abdoujjar@gmail.com

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 of medical and color image encryption, based on chaotic permutations acting at the bit level, and a diffusion confusion ensured by an application of the method of Vigenere largely improved and adapted to the subject. 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

- New sequence Chaotic design
- Implementation of Vigenere's advanced technique
- Switching to binary writing
    - Generation of chaotic permutations
    - New method setup

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

KEY WORD: **VIGENERE GRID; CHAOTIC MAP; BROADCAST FUNCTION; CHAOTIC PERMUTATION;**

I. INTRODUCTION

The rapid development of chaos theory in mathematics provides researchers with opportunities to further improve some classic encryption systems. In front of this great security focus, many techniques for color image encryption have flooded the digital world, mostly exploiting number theory and chaos By applying permutations at the pixel level $[1-2-3-4-5]$. Others are attempting to update their policies by improving some classical techniques, such as Hill $[6-7-8]$, Cesar, Vignere $[9-10]$, Feistel $[11-12]$.

1) Vigenere's classical technique

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

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

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

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

Even though Vigenere's matrix was known, the encryption was able to withstand several centuries. But, Babagh's cryptanalysis is not efficient in not knowing the size of the encryption key. Several attempts to improve Vigenere's technique have invaded the digital world we quote $[10-11]$. In this work, the new structure of the substitution matrix and its attached replacement function will be described in detail.

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 $[13-14-15]$, This new cryptosystem is based on the following axes

AXE 1: CHAOTIC SEQUENCES DEVELOPMENT

In order to build a new algorithm using a single-encryption key, we will use the $2d\ logistics\ map$ $[16-17-18]$.

This choice is due to the simplicity of its development and its high sensitivity to the initial parameters.

1) 2D Logistics Map Decryption

It is a two-dimensional chaotic sequence defined by second degree polynomials. This sequence is given by the equation below

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

All the conditions ensure the installation of the chaotic aspect.

2) Chaotic vector design

Our work requires the construction of four chaotic vectors(CL), (CR), $(KR)$and (KL) generated from the following algorithm

```
$Alg2$
    $Control\ and\ chaotic\ vector\ creation.$
    $for\ \ i = 1\ to24nm$
        $CL(i) = mod\left(E\left(\frac{x(i) + 2y(i)}{3} * 10^{11}, 254\right) + 1\right)$
        $KL(i) = mod\left(E\left(\frac{x(i) + y(i) * y(i)}{2} * 10^{11}, 253\right) + 2\right)$
        $KR(i) = E\left(\frac{KL(i) + CL(i)}{2}\right)$
        $if\ CL(i) \geq KL(i) then$
            $CR(i) = 0\ else\ CR(i) = 1$
        $end\ if$
    $Next\ i$
```

3) Binary control vector development.

we will construct two control vectors

```
$Alg3$
    $for\ \ i = 1\ to\ 12nm$
        $if\ x(i) \geq y(i)\ then$
            $BR(i) = 0\ Else$
            $BR(i) = 1$
        $End\ if$
    $Next\ i$
```

A passage of the (CL) vector in matrix (CM) of size $(8{,}3nm)$ is performed.

AXE 2: VIGENERE UPGRADE

In the first stage, Vigenere's $[19 - 20 - 21]$technology was greatly modified by integrating the new substitution matrix provided by the new powerful replacement function.

1) Original Image Vectorization

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

```
Alg4
    $For\ i = 1\ to\ \ nm$
    $X(3i-2) = V_r(i)$
    $X(3i-1) = V_g(i)$
    $X(3i) = V_b(i)$
    $Next\ i$
```

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.

The calculation of this value is described below

```
Alg 5
    $Initialization\ value\ computation.$
    $for\ i = 2\ to\ 3nm$
        $If\ CR(i) = 0\ Then$
            $V = IV \oplus X(i) \oplus CL(i)$
        $Else$
            $IV = IV \oplus X(i) \oplus KL(i)$
        $End\ if$
    $Next\ i$
```

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

- ✓ permutation (RP) obtained by descending ordering the first $256\ values$ of the sequence (U)
- ✓ permutation (RR) obtained by increasing the ordering the first $256\ values$ of the sequence (V),

with the following restrictions

$$Eq2\ \begin{cases} If\ PR(i) = 256\ Then\ PR(i) = 0 \\ If\ RR(i) = 256\ Then\ RR(i) = 0 \end{cases}$$

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

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

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

Example: in $(\boldsymbol{G_8})$

(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

$$Eq3 \qquad Y(i)=VG(CL(i),X(i))$$

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

$$Eq\ 4\quad\begin{cases}If\ BR(i)=0\ Then\\ V_1\big(X(i)\big)=Y(i)=VG\big(CL(i),VD(KL(i),X(i)\oplus KR(i))\big)\\ Else\\ V_2\big(X(i)\big)=Y(i)=VD\big(KL(i),VG(KR(i),X(i)\oplus CL(i))\big)\end{cases}$$

2) *First Encryption Process*

The figure below illustrates the first encryption process improved using only Vigenere

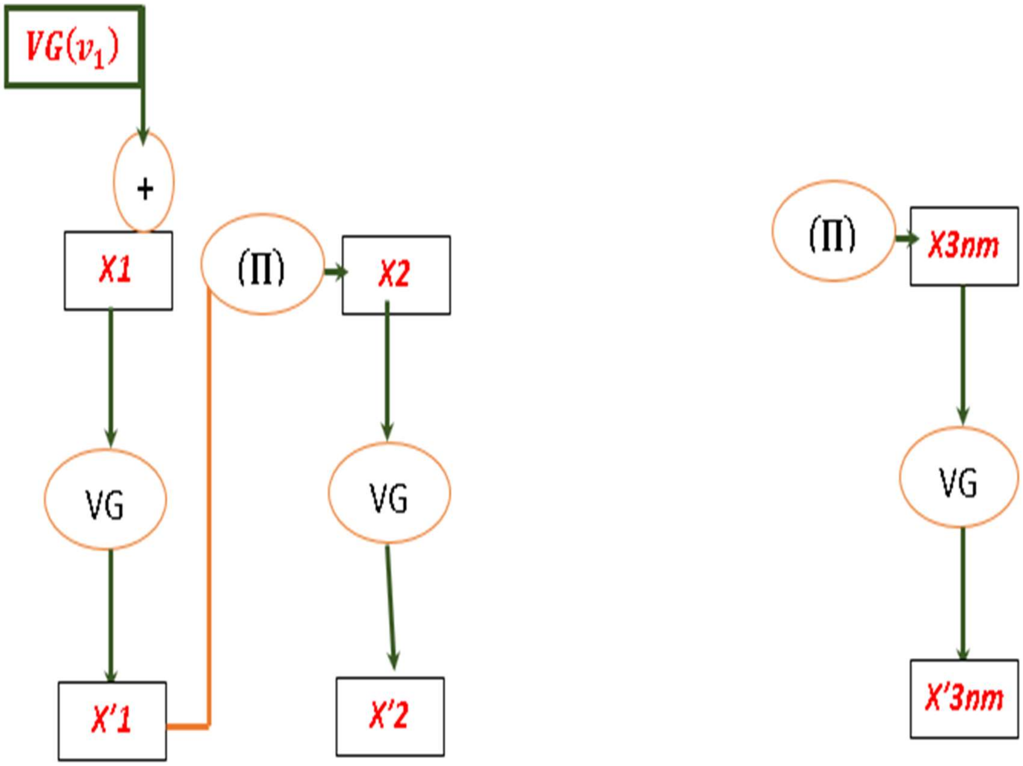


Figure2 : First Encryption

$$\begin{cases} \text{(V): advanced substitution matrix} \\ \text{(Π): New diffusion function} \end{cases}$$

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

$$\text{Eq5} \quad \Pi(X(i+1)) = VD\big(KR(i), VG(CL(i), X'(i) \oplus X(i+1))\big)$$

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

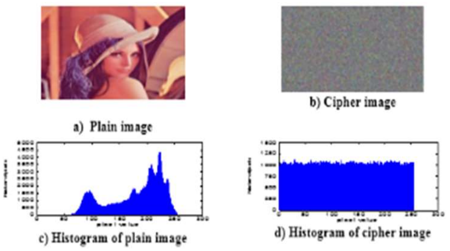
$$Alg\ 7 \begin{cases} X'(1) = VG\big(KR(1), VD(CL(1), IV \oplus X(1))\big) \\ For\ i)2\ to\ 3nm \\ \Pi\big(X(i)\big) = VD\left(KR(i), VG\big(CL(i), X'(i-1) \oplus X(i)\big)\right) \\ If\ BR(i) = 0\ Then \\ X'(i) = V_1\big(X(i)\big) \\ Else \\ X'(i) = V_2\big(X(i)\big) \\ End\ if \\ Next\ i \end{cases}$$

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

1) *First round analysis*

For a better follow-up of our algorithm, several reference images were tested by this first round, we quote

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

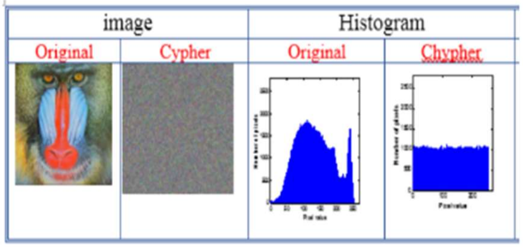


a) Plain image

b) Cipher image

c) Histogram of plain image

d) Histogram of cipher image



AXIS 4: PERMUTATION GENERATION

Eight permutations $(Q_i)_1^8$ are generated by the following process $[22-23-24]$

(Q_i) permutation in $(G_{3n}$)obtained by a descending sort in the broad sense on the row (i) of the matrix (CM)

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

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

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) is defined by the following algorithm

doi:10.20944/preprints202104.0647.v1

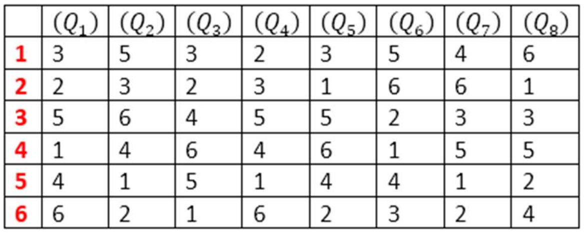
$$Alg8\left\{\begin{array}{c} For\ i = 1\ to\ 3nm \\ x = E\left(\frac{X'(i)}{16}\right) \\ y = X'(i) - 16 * x \\ \alpha = E\left(\frac{x}{4}\right) \\ \beta = x - 4 * \alpha \\ \gamma = E\left(\frac{y}{4}\right) \\ \delta = y - 4 * \gamma \\ XM(1,i) = T(\alpha,1) \\ XM(2,i) = T(\alpha,2) \\ XM(3,i) = T(\beta,1) \\ XM(4,i) = T(\beta,2) \\ XM(5,i) = T(\gamma,1) \\ XM(6,i) = T(\gamma,2) \\ XM(7,i) = T(\gamma,1) \\ XM(8,i) = T(\gamma,2) \\ Next\ i \end{array}\right.$$

Each row (i) of the matrix (XM) is subjected to the permutation $(Q_i)_1^8$, the resulting matrix (MX) is given by the algorithm below

$$Alg\ 9\ \left\{\begin{array}{c} For\ i = 1\ to\ 8 \\ For\ j = 1\ to\ 3nm \\ MX(i,j) = XM\big(i, Q_i(j)\big) \\ Next\ j, i \end{array}\right.$$

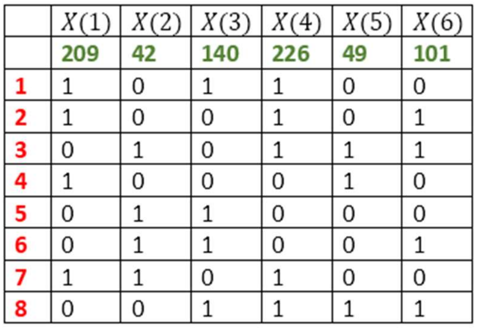
Example:

Permutations



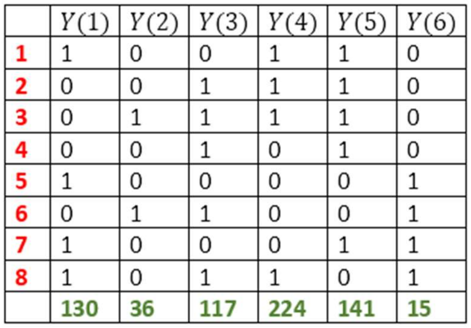
	(Q_1)	(Q_2)	(Q_3)	(Q_4)	(Q_5)	(Q_6)	(Q_7)	(Q_8)
1	3	5	3	2	3	5	4	6
2	2	3	2	3	1	6	6	1
3	5	6	4	5	5	2	3	3
4	1	4	6	4	6	1	5	5
5	4	1	5	1	4	4	1	2
6	6	2	1	6	2	3	2	4

Clear Pixel



	$X(1)$	$X(2)$	$X(3)$	$X(4)$	$X(5)$	$X(6)$
	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

Cypher Pixel



	$Y(1)$	$Y(2)$	$Y(3)$	$Y(4)$	$Y(5)$	$Y(6)$
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

STEP 5: DECRYPTION ENCRYPTED IMAGES

The solid chaining establishes between the encrypted block and the next clear block in the broadcast process, forcing us to start decryption from the last block using the opposite functions. So, the decryption process should follow these steps

- Binary conversion
- Reciprocal permutation generation
- Switching to grayscale
- Vigenere reciprocal matrix
- Vigenere's reciprocal application

1) Reciprocal permutation generation

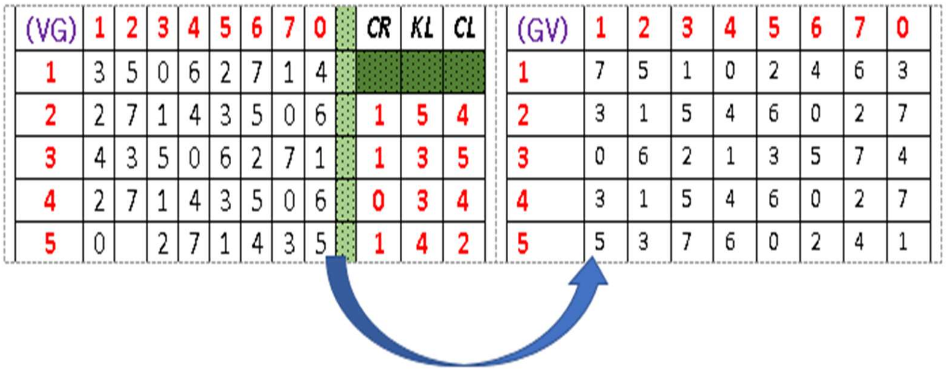
Consider $(T_i)_1^8$the reciprocal permutation of $(Q_i)_1^8$, it defines by the following algorithm

$$Alg10\begin{cases} For\ j = 1\ to\ 3nm \\ \quad T_i\big(Q_i(j)\big) = j \\ \qquad Next\ j \end{cases}$$

2) Vigenere Reciprocal matrix

$$Alg11\begin{cases} \ for\ \ i = 1\ \ to\ \ 256 \\ \ \ for\ j = 1\ to\ \ 256 \\ \ GV\big(i, VG(i,j)\big) = j \\ DV\big((i, VD(i,j)\big) = j \\ \qquad Next\ j, i \end{cases}$$

Example



(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

$$Equation6 \quad \begin{cases} if\ \ z = VG(y,x) \\ \quad Then \\ \ x = GV(y,z) \end{cases}$$

3) Vigenere's inverse expression

$$Algo12\begin{cases} if\ BR(k) = 0\ then \\ V_1^{-1}(X'_k) = DV\big(KL(k), GV(CL(k), X'_k)\big)\oplus KR(k) \\ else \\ V_2^{-1}(X'_k) == GV\big(KR(k), DV(KL(k), X'_k)\big)\oplus CL(k) \\ end\ if \end{cases}$$

4) Reverse diffusion

$$Equation\ 22 \qquad \Pi^{-1}(X'(k)) = \mathrm{GV}(CL(k), DV(KR(k), X'(\mathrm{k}))\oplus X(k-1)$$

AXIS 5: EXAMPLES AND SIMULATIONS

In order to measure the performance of our encryption system, we randomly select a large number of reference images, and then use our method to test them

1) Brutal assaults

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

a. Key-space analysis

The chaotic sequence used in our method ensures strong sensitivity to initial conditions, and can protect it from any brutal attacks.

The secret key to our system consists of

$$Equation\ 7 \qquad \begin{cases} x_0 = 0{,}7655412001\ , \mu_1 = 3.89541 \\ y_0 = 0.865421331,\ \ \mu_2 = 0{,}56120 \\ \mu_3 = 1{,}3561 \qquad\qquad \mu_4 = 0{,}56321 \end{cases}$$

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.

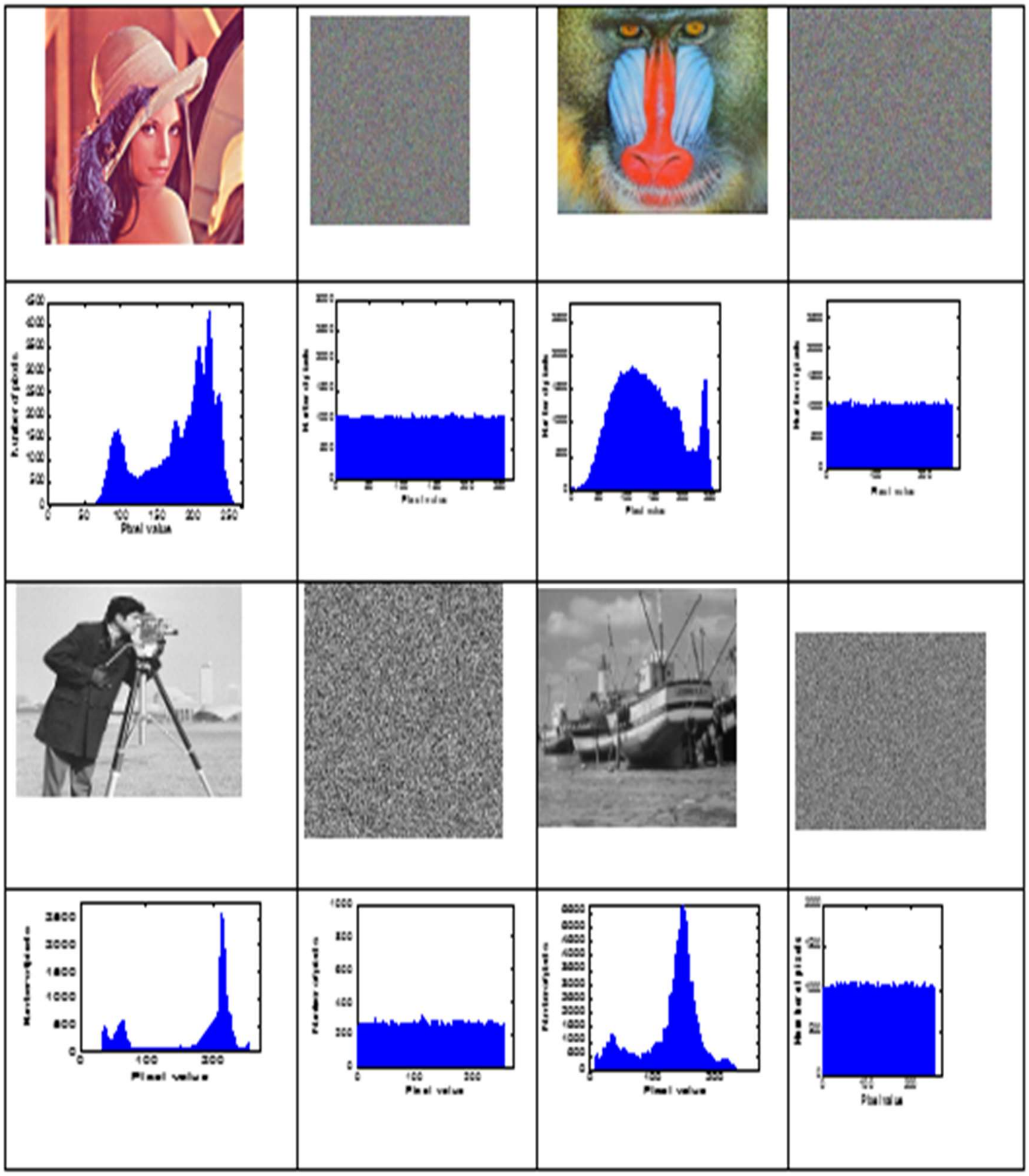
b. Secret key's sensitivity Analysis

Our encryption key has a high sensitivity, which means that a small degradation of a single parameter used will automatically cause a large difference from the original image. This ensures that in the absence of the real encryption key, the original image cannot be restored.

2) Histogram analysis

all images tested by our algorithm have a uniformly distributed histogram. This reflects that the entropy of the encrypted images is around 8, which makes the system immune to histogram attacks. The table1 shows that the horizontal correlation values of the encrypted images are close to zero. This ensures high security against correlation attacks.

Table 1: Encrypted image histogram



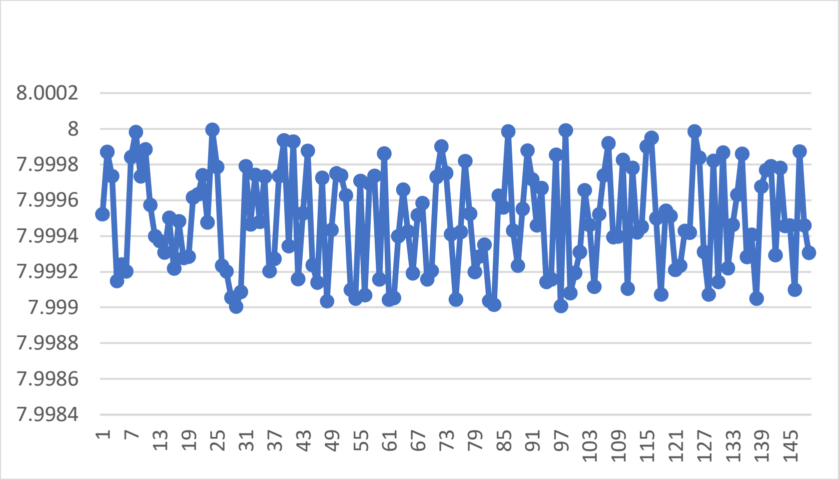
3) Statistics Attack Security

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

$$Equation\ 8 \qquad H(MC) = \frac{1}{t}\sum_{i=1}^{t} -p(i)\, log_2(p(i))$$

The entropy values on the $150\ images$ arbitrarily chosen from a large database of images of different sizes and formats, tested by our method are represented graphically by the following figure



Fugure3: Entropy of 150 images

All the entropy values of the images tested by our algorithm are close to 8, which confirms the uniformity of the histogram. This proves that the method is far from a statistical attack.

b. Entropy statistical analysis

We will study the uniformity of the distribution of entropy released by the test.

i. 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.

The purpose of this analysis is to show that the distribution follows a reduced central normal distribution. So

$$\text{Equation 9} \begin{cases} Q_1 = & First\ quartile \\ Q_2 = & Second\ quartile \\ Q_3 = & Third\ quartile \end{cases}$$

Position Parameters

Average	Max	Min	Q1	Q2	Q3
7,999120	7,99993210	7,999002	7,9990132	7,999310	7,999721

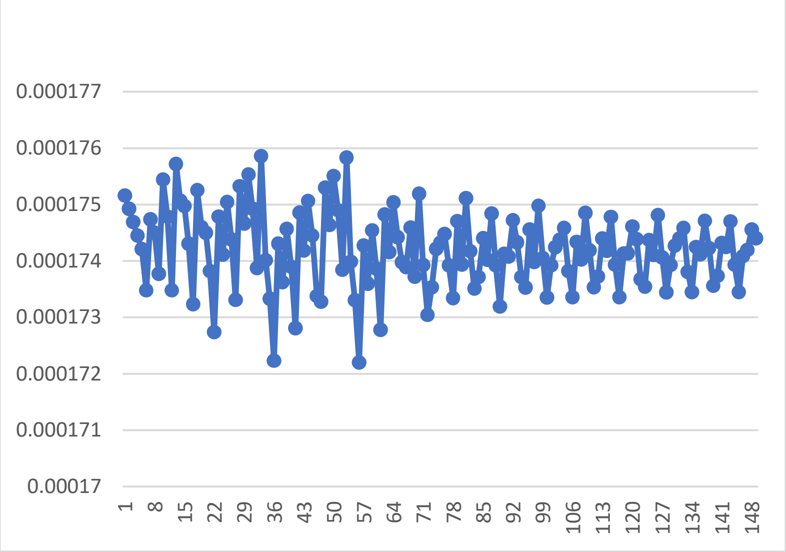
c. Correlation analysis

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

$$equation\ 29\ \ correlation \qquad r = \frac{cov(x,y)}{\sqrt{V(x)}\sqrt{V(y)}}$$

i. Horizontal Correlation

The following figure graphically represents the simulation of 150 color images of the same size, which are selected from an image database of various sizes, formats and related values



Fugure4: Entropy of 150 images

Average	Max	Min	Q1	Q2	Q3
7,999120	0,000179	0,000170	0,000173	0,000174	0,000176

ii. Vertical Correlation

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

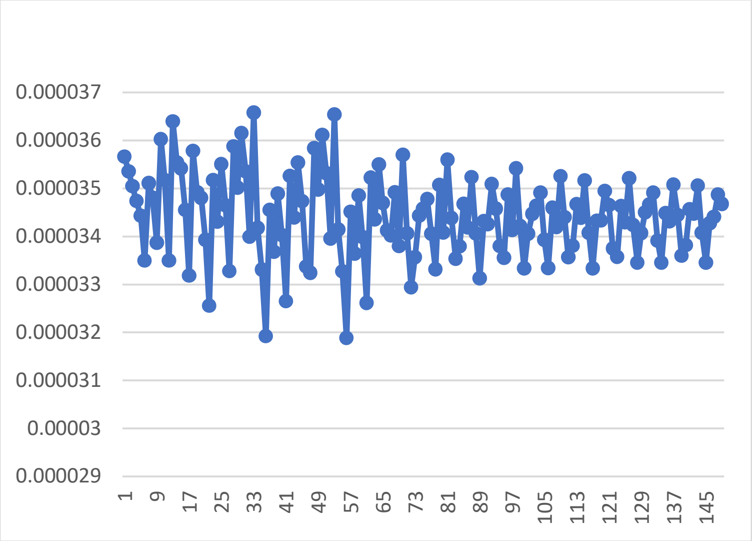


fIgure5: Vertical correlation of 150 images

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

iii. Diagonal Correlation

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

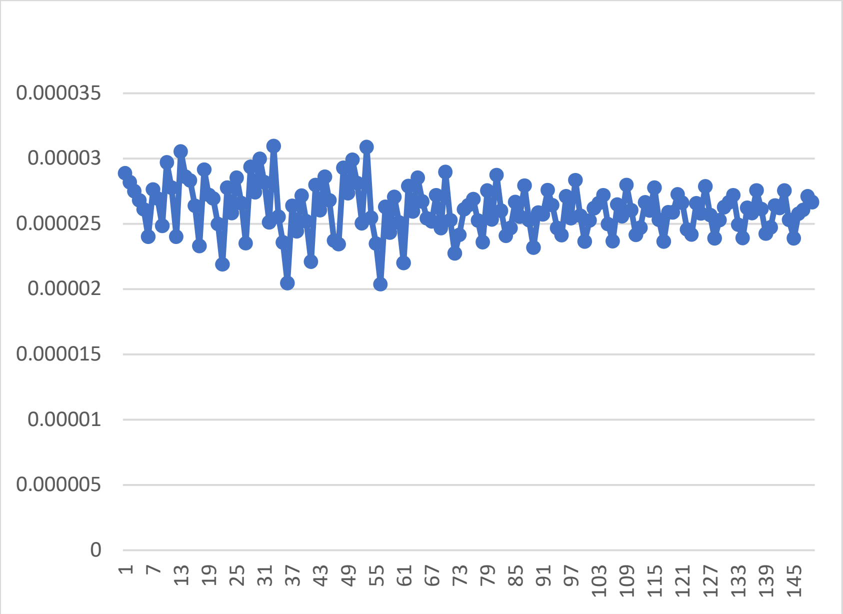


fIgure6: Diagonal correlation of 150 images

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

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

$$Equation\ 10 \qquad NPCR = \left(\frac{1}{nm}\sum_{i,j=1}^{nm} D(i,j)\right) * 100$$

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

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

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

The study of the $150\ selected\ images$ revealed the following diagram

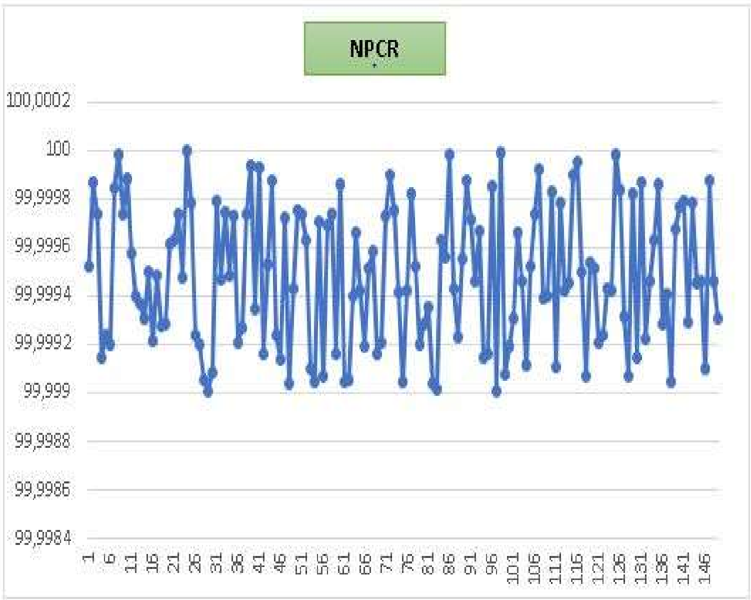


fIgure7: NPCR of 150 images

Average	Max	Min	Q1	Q2	Q3
99,995	99,999	99,991	99,993	99,994	99,997

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

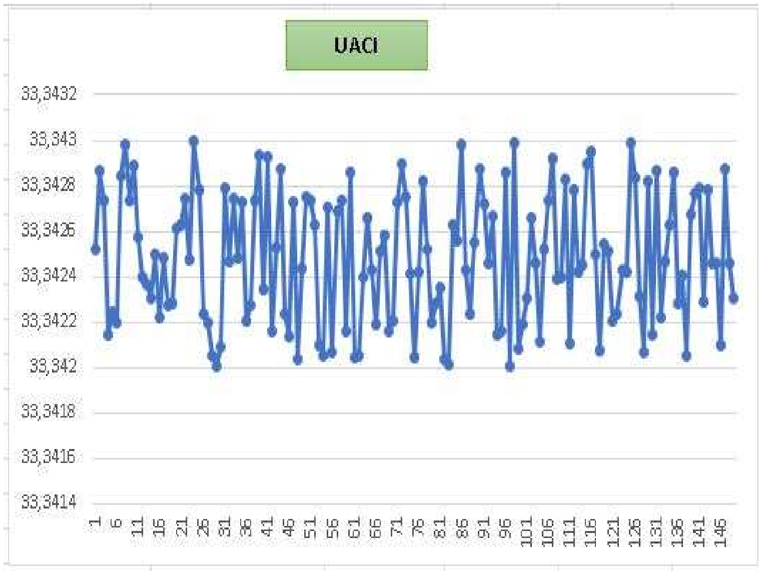


fIgure8: UACI of 150 images

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 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

$$Equation\ 12 \quad AE = \left(\frac{\sum_i bit\ change}{\sum_i bit\ total}\right) * 100$$

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

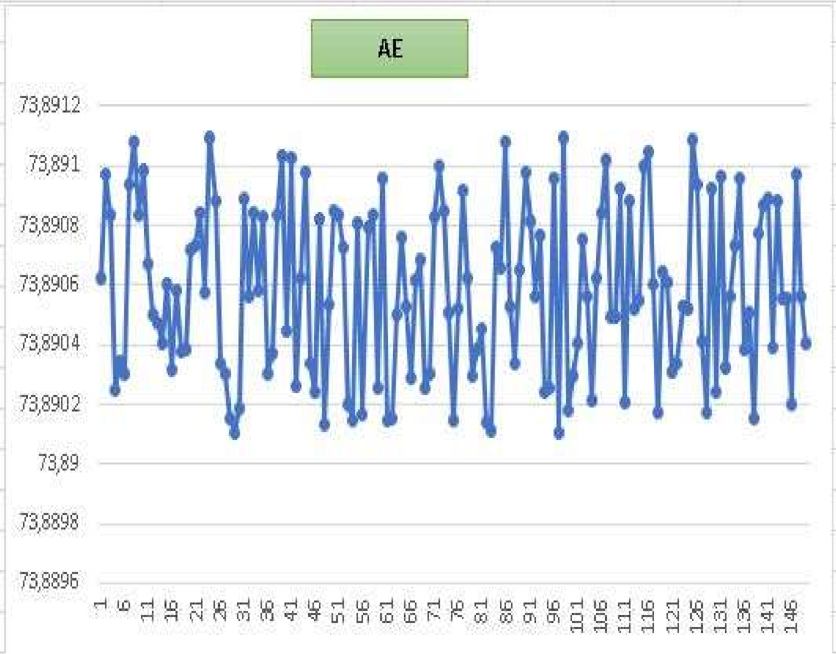


Figure9: Avalanche effect

Average	Max	Min	Q1	Q2	Q3
73,8906	73,892	73,890	73,8903	73,8905	73,891

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

$$Equation\ 13\quad MSE = \sum_{i,j}(P(i,j) - C(i,j))^2$$

$(P(i,j))$; 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

$$Equation\ 14\quad PSNR = 20Log_{10}\left(\frac{I_{max}}{\sqrt{MSE}}\right)$$

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.

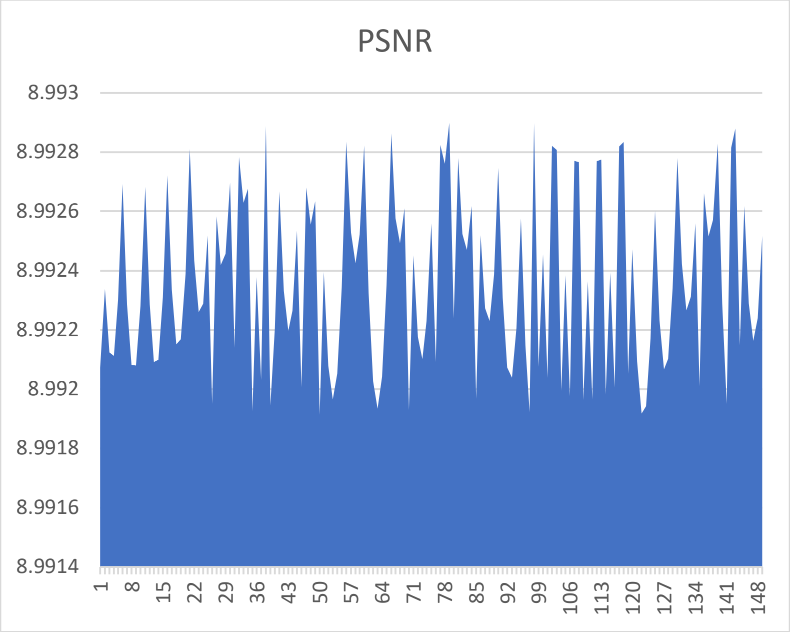
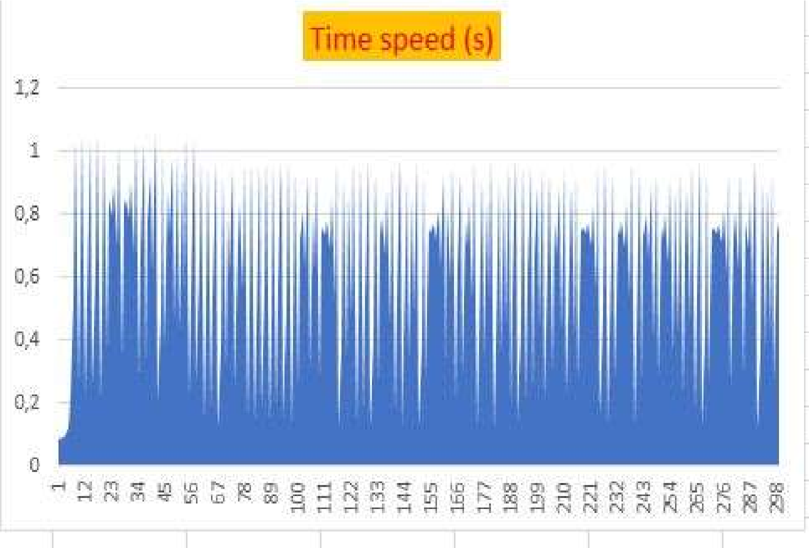


fIgure10: ¨PSNR of 150 images

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.



flgure11: 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

This new algorithm which starts with an implementation of the greatly improved Vigenere technique ensured by two chaotic substitution matrices and installation of a diffusion ensured also by the two Vigenere matrices, followed by a passage in binary writing for an application of eight chaotic permutations acting at the level of the bits for a better diffusion confusion can face any known attack .

References

[1]	Saiyma Fatima Raza « A novel bit permutation-based image encryption algorithm » Nonlinear Dynamic 95, pages859–873(2019)
[2]	Guangfeng Cheng, Chunhua Wang » A Novel Color Image Encryption Algorithm Based on Hyperchaotic System and Permutation-Diffusion Architecture "International Journal of Bifurcation and Chaos Vol. 29, No. 09, 1950115 (2019)
[3]	RasulEnayatifar "Index-based permutation-diffusion in multiple-image encryption using DNA sequence "Optics and Lasers in Engineering Volume 115, April 2019, Pages 131-140
[4]	LinqingHuang « On symmetric color image encryption system with permutation-diffusion simultaneous operation "Optics and Lasers in Engineering Volume 115, April 2019, Pages 7-20
[5]	Shamama Anwa "A pixel permutation based image encryption technique using chaotic map "Multimedia Tools and Applications volume 78, pages27569–27590(2019)

[6]	A.P.USiahaan "Genetic algorithm in Hill cipher encryption"international association of scientific innovation and research (IASR) vol 15,no,1 2016
[7]	A.S alkhalid « cryptanalyse of Hill cipher using genetic algorithm" dalam IEEE hanmument 2015
[8]	A.Jarjar« Improvement of hill' sclassical method in image cryptography » International Journal of Statistics and Applied Mathematics 2017, Volume 2 Issue 3, Part A
[9].	Imam Saputra, Mesran, Nelly Astuti Hasibuan3, "Vigenere Cipher Algorithm with Grayscale Image Key Generator for Secure Text File" International Journal of Engineering Research & Technology (IJERT), Vol. 6 Issue 01, January-2017
[10].	Vaka Vamshi Krishna Reddy, Sreedhar Bhukya 2, "ENCRYPT AND DECRYPT IMAGE USING VIGENERE CIPHER", International Journal of Pure and Applied Mathematics, Volume 118 No. 24 2018
[11]	A.P.U siahaan" three pass protocol in Hill cipher encryption technique" international journal of science and research (IJSR) vol 5 nà 7 2016 pp 1149-1152
[12]	I Gede Arya Putra Dewangga, Tito Waluyo Purboyo, Ratna Astuti Nugrahaeni," A New Approach of Data Hiding in BMP Image Using LSB Steganography and Caesar Vigenere Cipher Cryptography" International Journal of Applied Engineering Research, Volume 12, Number 21 (2017) pp. 10626-10636
[13]	Overbey.J.traversW and W ydylo J 2005 " On the key space of the Hill cipher" Cryptologia 29(1), 59-72
[14]	Saeednia,S 2000 "haow to make the Hill secure" Cryptologia 24(2), 353-360
[15]	Lin.C.H.Lee.C.Y and Lee.C.Yu 2004" comments on saeednia's improved scheme for the Hill cipher." Journal of the chineese institute of engineers 27/5, 743-746
[16]	A.Jarjar« Improvement of hill's classical method in image cryptography » International Journal of Statistics and Applied Mathematics 2017, Volume 2 Issue 3, Part A
[17]	lYongWang all » A chaos-based image encryption algorithm with variable control parameters" Chaos, Solitons & Fractals Volume 41, Issue 4, 30 August 2009, Pages 1773-1783"
[18]	Jan Sher Khan all, "Chaos based efficient selective image encryption" Multidimensional Systems and Signal Processing volume 30, pages943–961(2019)"
[19]	H Li, Y Wang, Z Zuo - Optics and Lasers in Engineering, 2019 "Chaos-based image encryption algorithm with orbit perturbation and dynamic state variable selection mechanisms" Volume 115, April 2019, Pages 197-207
[20]	Rongjun Ge all « A Novel Chaos-Based Symmetric Image Encryption Using Bit-Pair Level Process" July 8, 2019, date of current version August 7, 2019.
[21]	Mohamed JarJar "Further improvement of the HILL method applied in image encryption" Procedia computer sciences 00(2019)000-000
[22]	Shams Mahmoud Abd Ali » Novel Encryption Algorithm for Securing Sensitive Information Based on Feistel Cipher"Test engeenering managenment Page Number: 10 - 16 Publication Issue:19 Volume: 80 September-October 2019
[23]	Zhi-hua Gan » A chaotic image encryption algorithm based on 3-D bit-plane permutation » Neural Computing and Applications (2019) 31:7111–7130