



## Instructions New Technology of Color Image Encryption Based Two Improved Vigenere Laps Separated by a Genetic Mutation

Mohamed Jarjar<sup>1\*</sup>, Said Hraoui<sup>2</sup>, Said Najah<sup>1</sup>, Khalid Zenkour<sup>1</sup>

<sup>1</sup> Lab-SIA, Faculty of Sciences and Technologies, USMBA University, B.P. 2202 - Rte d'Imouzzer 30050, Fes, Morocco

<sup>2</sup> LIASSE, National School of Applied Sciences, USMBA University, B.P. 2202 - Rte d'Imouzzer 30050, Fes, Morocco

Corresponding Author Email: [mohamed.jarjar@usmba.ac.ma](mailto:mohamed.jarjar@usmba.ac.ma)

<https://doi.org/10.18280/ijssse.110512>

### ABSTRACT

**Received:** 13 June 2021

**Accepted:** 2 October 2021

#### Keywords:

*Vigenere grid, chaotic map, encryption function, S-Box, genetic mutation*

This document traces the development of a new cryptosystem using two circuits ensured by a deep Vigenere classical technique improvement. This new technique employs several dynamic substitutions matrices attached to chaotic replacement functions; whose construction will be detailed. The first round will start by modifying the seed pixels based on the initial values calculated from the original image, and will be infected through the chaotic map used to overcome the uniform image problem, followed by the injection of Vigenere technology improvements. The output vector will be subdivided into three sized blocks for future application of deeply improved genetic mutations to better adapt to medicine and color image encryption. The second round will increase the complexity of the attack and improve the installed systems. Simulations performed on a large number of images of different sizes and formats ensure that our approach is not exposed to known attacks.

## 1. 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 [1, 2]. Others are attempting to update their policies by improving some classical techniques, such as Hill [3, 4], Cesar, Vignere [5, 6], Feistel [7, 8].

### 1.1 Vigenere's classical technique

This technology is based on static ( $V$ ) matrix defined by the following algorithm. Despite the knowledge of the substitution matrix, this method has been able to withstand more than three centuries.

#### Algorithm 1. Classical Vigenere

```

Fist Row
For i = 1 to 26
  V(1, i) = i
Next i
folloing Rows
For i = 2 to 26
  For j = 1 to 26
    V(i, j) = V(i - 1, (j + 1), 26)
  Next j, i

```

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

$$\begin{cases} C_i = V(P_i, K_i) = (P_i + K_i) \mod 26 \\ P_i = V(C_i, K_i) = (P_i - K_i) \mod 26 \end{cases} \quad (1)$$

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 [9, 10]. In this work, the new structure of the substitution matrix and its attached replacement function will be described in detail.

### 1.2 Problematic

In the conventional Vigenere system, the recognition problem of the private size key, exposes the algorithm to statistical attacks. discovered and detailed by Babagh. The knowledge of the substitution matrix is an opportunity to expose the conventional system to brute force attacks. Moreover, in the absence of the broadcast operation and the chaining facility, all classical systems remain exposed to differential attacks. In addition, block ciphering independently facilitates the implementation of dictionary and statistical attacks.

### 1.3 Our contribution

Our contribution is to improve the encryption structure and function for replace the substitution matrix. To this end, two improved encryption functions will be constructed and two S-Boxes will be generated in different ways from the two most widely used chaotic graphs in the world of cryptography [11, 12]. In addition, the principle of double encryption will be applied to all pixels of the original image, and a chaotic broadcast will be installed in each tower, which will increase the impact of the avalanche effect and protect the system from differential attacks. Contrary to the classic method, our system will use different S-Boxes for decoding, and different





## 2.10 The first-round analysis

This first round is defined by the following algorithm,

$$\text{We note: } V_1(X(i)) = Y(i) \quad (9)$$

So

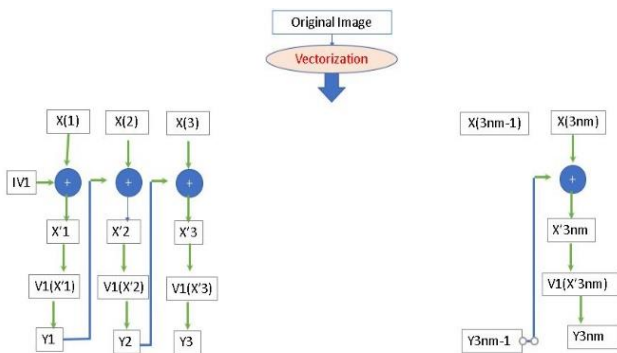
---

**Algorithm 7.** 2st initialization value computing

$$\left\{ \begin{array}{l} X'(1) = IV1 \oplus X(1) \\ Y(1) = V_1(X'(1)) \\ \text{For } i = 2 \text{ to } 3nm \\ \alpha = \Phi(X(i)) \\ Y(i) = V_1(\alpha) \oplus KR(i) \\ \text{Next } i \end{array} \right.$$


---

Figure 1 below shows the first round



**Figure 1.** First round

At the end of the first round, the output vector ( $Y$ ) will be treated as a clear image to be applied to the second round of encryption. The output vector is subdivided into three blocks of size  $(1, nm)$  for future gene mutations.

*Step5: Genetic Mutation*

The output vector is subdivided into  $(m)$  blocks of  $(3n)$  pixels each as well as the chaotic vector ( $CL$ ), for future chaotic mutation between the two vectors. This operation will be supervised by the ( $PH$ ) permutation vector obtained by a broad ascending sort on the binary vector ( $BC$ ) and generate by the following process:

---

**Algorithm 8.** ( $PH$ ) computing

$$\left\{ \begin{array}{l} h = 1 \\ \text{For } i = m \text{ to } 1 \\ \text{If } CR(i) = 1 \text{ then} \\ \text{If } CR(i) = 0 \text{ then} \\ PH(i) = h \\ h = h + 1 \\ \text{end if} \\ \text{end if} \\ \text{Next } i \end{array} \right. \quad \left\{ \begin{array}{l} \text{For } i = m \text{ to } 1 \\ \text{If } CR(i) = 1 \text{ then} \\ PH(i) = h \\ h = h + 1 \\ \text{end if} \\ \text{Next } i \end{array} \right.$$


---

*Example (Over 14 bits)*

0 0 1 1 0 1 1 1 0 0 1 1 0 1

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

$$PH = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 \\ 6 & 5 & 14 & 13 & 4 & 12 & 11 & 10 & 3 & 2 & 9 & 8 & 1 & 7 \end{pmatrix}$$

The mutation function is the confusion of the original sub-block with the chaotic sub-block only in the case where the bit of the vector ( $CR$ ) is not zero. This operation is defined by the following algorithm

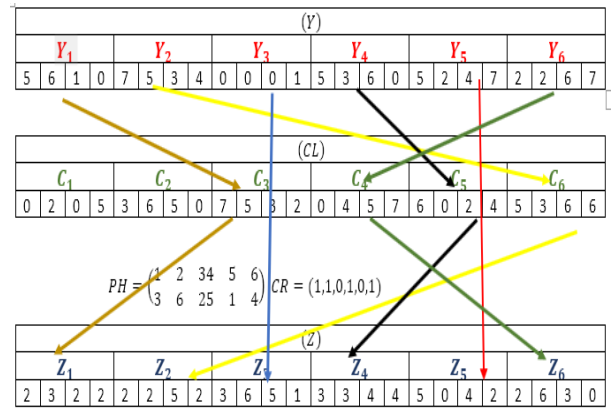
---

**Algorithm 9.** Mutation's new expression

$$\left\{ \begin{array}{l} Y = (Y_1 Y_2, Y_3, \dots Y_m,) \\ CL = (C_1, C_2, C_3 \dots C_m,) \\ Mt(Y, CL) = Z = (Z_1, Z_2, Z_3 \dots Z_m,) \\ \text{With } \forall i \in [1 m] \\ \text{If } CR(i) = 0 \text{ Then} \\ Z_i = Y_i \oplus C_{(PH(i))} \\ \text{Else } Z_i = Z_i \end{array} \right.$$


---

*Example:*



*Step6: Second Vigenere round*

At the end of the first round, the new ( $IV2$ ) initialization value will be calculated according to the following algorithm.

---

**Algorithm 10.** Second initialization value

$$\left\{ \begin{array}{l} \text{for } i = 2 \text{ to } 3nm \\ IV2 = IV2 \oplus Y(i) \\ \text{Next } i \end{array} \right.$$


---

In the second round, by simply replacing the position of the replacement matrix, the output vector will be treated as a new image to be encrypted by the same method as the first round.

## 2.11 Second round analysis

The second round can also be ensured by using a different same matrix in the first round.

---

**Algorithm 11.** Vigenere's second function

$$V_2(X(i)) = \left\{ \begin{array}{l} \text{if } VC(i) = 0 \text{ then} \\ Y(i) = VD1(GL(i), VG1(MR(i); X(i))) \oplus ML(i) \\ \text{else} \\ Y(i) = VG1(ML(i), VD1(GL(i), X(i))) \oplus MR(i) \end{array} \right.$$


---

The same mold will be used in the second round, but in a different way.

### 2.11.1 Second-round spread function expression

The second round will be equipped with the diffusion ( $\Omega$ ) ensured by the replacement matrix generated. The expression of this function is defined by the following notation:

**Algorithm 12.** Second function of diffusion

$$\forall i > 1 \Omega(X(i)) = VD1(ML(i), X'(i - 1) \oplus X(i))$$

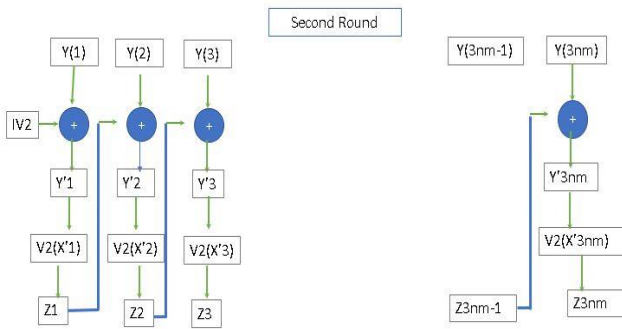
2.11.2 The second-round analysis

This second round is defined by the following algorithm:

**Algorithm 13.** 2° round function

$$\begin{cases} Y'(1) = IV2 \oplus Y(1) \\ Z(1) = V_2(Y'(1)) \\ \text{For } i = 2 \text{ to } 3nm \\ \alpha = \Omega(Y(i)) \\ Z(i) = V_2(\alpha) \\ \text{Next } i \end{cases}$$

Figure 1.1 below shows the first round.



**Figure 1.1.** Second round

The output vector ( $Z$ ) will be subjected to a permutation ( $RH$ ) obtained by sorting on the vector ( $CR$ ) by the same process). This permutation is applied to increase the complexity of our system It is defined by the following algorithm:

**Algorithm 14.** Permutation application

$$\begin{cases} \text{For } i = 1 \text{ to } 3nm \\ ZC(i) = PH(Z(i)) \\ \text{Next } i \end{cases}$$

The vector ( $ZC$ ) constitutes the image encrypted by our algorithm.

*Step7: Decryption of encrypted images*

In the literature, the classic Vigenere method uses the same matrix in both processes. Our contribution in this work is that the matrix used in encryption is different from the matrix used in decryption. Therefore, the calculation of the decryption matrix is necessary.

**2.12 Decryption matrix structure**

Each row of the encrypted S-box is a permutation in ( $G_{256}$ ), so the decryption matrix will consist of reverse permutations. For this reason, two decrypted  $S - Box$  generations are given by the following algorithm:

**Algorithm 15.** Vigenere inverse matrices

$$\begin{cases} \text{for } i = 1 \text{ to } 256 \\ \text{for } j = 1 \text{ to } 256 \\ VG2(i, VG1(i, j)) = j \\ VD2(i, VG2(i, j)) = j \\ \text{Next } j, i \end{cases}$$

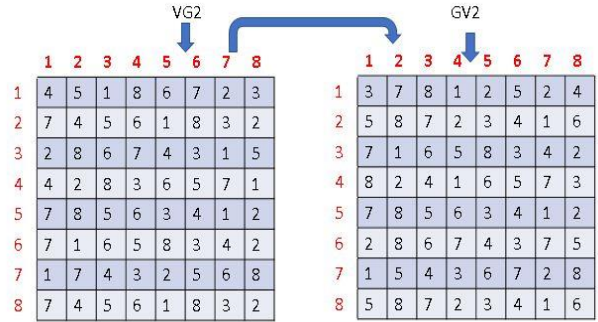
2.12.1 Decryption matrix structure

Each row of the encrypted S-box is a permutation in ( $G_{256}$ ), so the decryption matrix will consist of reverse permutations. For this reason, two decrypted  $S - Box$  generations are given by the following algorithm:

**Algorithm 16.** Inverse Matrices

$$\begin{cases} \text{for } i = 1 \text{ to } 256 \\ \text{for } j = 1 \text{ to } 256 \\ VG2(i, VG1(i, j)) = j \\ VD2(i, VG2(i, j)) = j \\ \text{Next } j, i \end{cases}$$

Example



The decryption process will follow the following reverse steps

- ✓ Application of the inverse permutation ( $HP$ ) of ( $PH$ )
- ✓ Application of the reverse of the second round
- ✓ Application of the inverse of the mutation
- ✓ Application of the reverse of the first round

2.12.2 Reverse permutation

The inverse permutation ( $HP$ ) of ( $PH$ ) is given by the following algorithm:

**Algorithm 17.** Inverse permutation

$$\begin{cases} \text{For } i = 1 \text{ to } 3nm \\ HP(PH(i)) = i \\ \text{Next } i \end{cases}$$

After vectorization of the image encrypted in vector ( $ZC$ ), an intervention of the permutation ( $HP$ ) to recover the vector ( $Z$ ).

This operation is determined by the following algorithm:

**Algorithm 18.** Inverse permutation application

$$\begin{cases} \text{For } i = 1 \text{ to } 3nm \\ Z(i) = HP(ZC(i)) \\ \text{Next } i \end{cases}$$

2.12.3 The reciprocal of the Second lap function

This step is given by the algorithm below:

**Algorithm 19.** Second lap Invers

$$\begin{cases} \text{For } i = 3nm \text{ to } 1 \\ Y(i) = V_2^{-1}(Z(i)) \oplus Z(i - 1) \\ \text{Next } i \end{cases}$$

A recalculation of the initialization value will make it possible to retrieve the exact value of pixel  $Y(1)$ .

### 2.12.4 The reverse mutation

In general, mutation is an involutive operation, therefore we have

$$\text{Algorithm 20. Reverse mutation}$$

$$(Mt)^{-1} = Mt$$

## 3. 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. In this part, all experiments are performed under the Matlab software running under Windows 7, on a basic i7 personal computer, 16 GB RAM, and 500 GB hard disk.

### 3.1 Key-space analysis

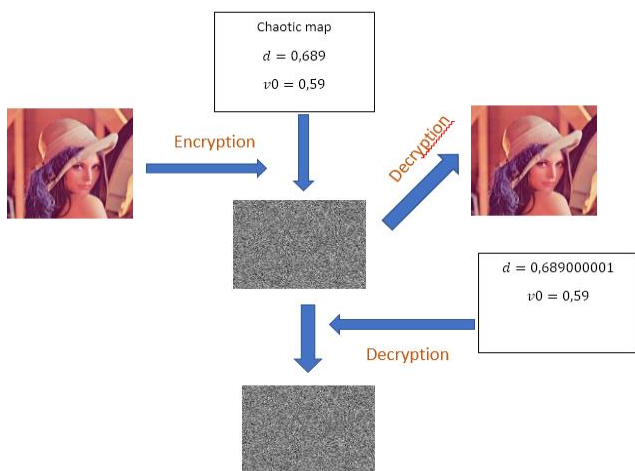
The chaotic sequence used in our method ensures strong sensitivity to initial conditions and can protect it from any brutal attacks. The secret key to our system consists of

Key size
Logistic map
$u_0 = 0,7655412001, \mu = 3.89231541$
PWLCM
$v_0 = 1,3561 p = 0.623$
SKTM
$w_0 = 1,3561 d = 0.752$

If we use single-precision real numbers  $10^{-10}$  to operate, the total size of the key will greatly exceed  $\approx 10^{-60} \gg 2^{180} \gg 2^{110}$ , which is enough to avoid any brutal attacks.

### 3.2 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. The image below illustrates this confirmation:



We notice that a tiny perturbation on a single element of the secret key, will generate a random decrypted image that is clearly different from the original image. This ensures a high sensitivity to the secret key, and therefore, in the absence of the real encryption key, the original image cannot be restored.

## 3.3 Statistics attack security

### 3.3.1 Entropy analysis

The entropy of an image of size  $(n, m)$  is given by the equation below

$$H(MC) = \frac{1}{t} \sum_{i=1}^t -p(i) \log_2(p(i)) \quad (10)$$

$p(i)$  is the probability of occurrence of level  $(i)$  in the original image attendance.

Table 1. Entropy of some tested images

Image	Size	Cypher	Entropy
	256x256		7,9993
	512x512		7,9998
	512x512		7,9997
	1024x1024		7,9999
	256x256		7,9991

We noticed that the entropy of all images tested by our algorithm is close to 8, which is the maximum value. These values ensure that our system is protected from entropy attacks (Table 1).

### 3.3.2 Correlation analysis

The correlation of an image of size  $(n, m)$  is given by the equation below

$$r = \frac{cov(x, y)}{\sqrt{V(x)}\sqrt{V(y)}} \quad (11)$$

Table 2. Correlation of some tested images

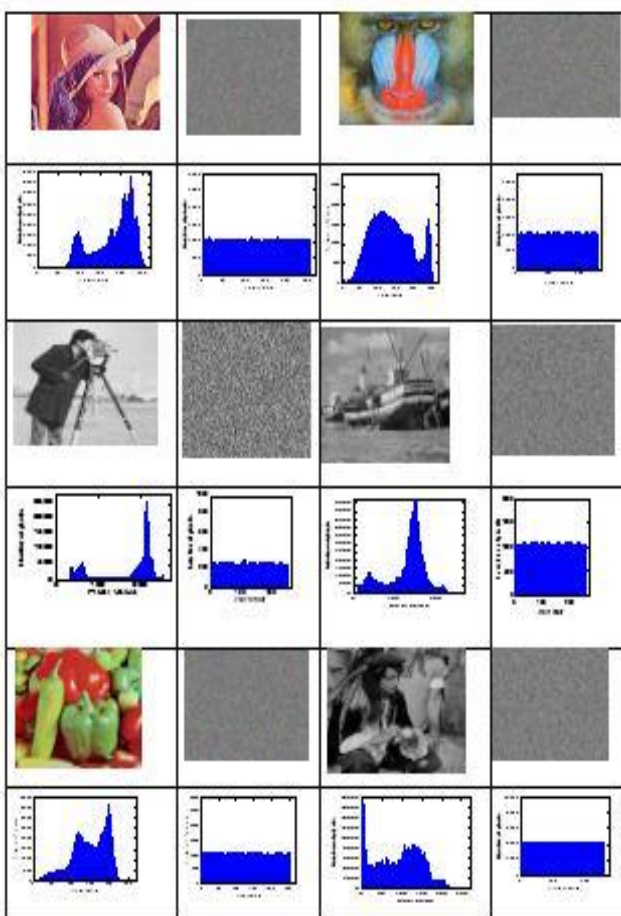
Image	Size	Original Image			Encrypted Image		
		H-C	V-C	D-C	H-C	V-C	D-C
	512x512	0.9047	0.8520	0.8238	-0.0007	-0.0004	0.0001
	1024x1024	0.9774	0.9813	0.9668	-0.0001	-0.0002	-0.0010
	512x512	0.9786	0.9820	0.9694	-0.0002	0.0006	0.0002
	512x512	0.9774	0.9881	0.9696	0.0023	-0.0001	-0.0003

Pixel correlation measures the independence of neighboring pixels. All the correlation measures of the images tested by our system are very close to zero. This can protect our methods from statistical attacks (Table 2).

### 3.3.3 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 (Table 3).

**Table 3.** Encrypted image histogram



### 3.4 Differential analysis

In cryptography, differential attacks are managed by the following constants.

#### 3.4.1 The NPCR constant

It is determined by the equation below

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

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

#### 3.4.2 The UACI constant

The *UACI* mathematical analysis of an image is given by the next equation

$$UACI = \left( \frac{1}{nm} \sum_{i,j=1}^{nm} Abs(C_1(i,j) - C_2(i,j)) \right) * 100 \quad (13)$$

### 3.4.3 Signal-To-Peak Noise Ratio (PSNR)

#### (1) MSE

The *MSE* mathematical analysis of an image is given by the next equation

$$MSE = \sum_{i,j} (P(i,j) - C(i,j))^2 \quad (14)$$

- ✓  $(P(i,j))$  ; pixel of the clear image
- ✓  $(C(i,j))$ : pixel of the cypher image

#### (2) PSNR

The *PSNR* mathematical analysis of an image (Table 4) is given by the next equation

$$PSNR = 20 \log_{10} \left( \frac{I_{max}}{\sqrt{MSE}} \right) \quad (15)$$

**Table 4.** Differential parameters

Image	Size	NPCR	UACI	PSNR
	256x256	99,92	33,35	8,36
	512x512	99,67	34,23	8,65
	1024x1024	99,96	33,37	8,10

### 3.4.4 Avalanche effect

**Table 5.** Avalanche effect

Original Image	Cypher Image	AE
		78,25
		77,04
		76,26






Our algorithm uses a strong link between encrypted pixels and subsequent clear pixels in the strategy. This leads to a gradual change in the value, which becomes more and more important as the data spreads through the structure of the algorithm. The avalanche effect is the number of bits that have been changed if a single bit of the original image is changed (Table 5). The mathematical expression of this avalanche effect is given by

$$AE = \left( \frac{\sum_i \text{bit change}}{\sum_i \text{bit total}} \right) * 100 \quad (16)$$

### 3.4.5 Performance time

In our technique, the encryption and decryption times (Table 6) are very similar and vary in the *interval* [0,05 0,1].

**Table 6.** Encryption time

Image	Size	Time	
		Encryption	Decryption
	256x256	0,04	0,06
	512x512	0,02	0,03
	256x256	0,03	0,02
	512x512	0,06	0,05
	256x256	0,02	0,03

## 4. MATH SECURITY

Our encryption keys are large, which can ensure that the new system is protected from brute force attacks. At the same time, the randomness of the operators described in the system makes it difficult to unlock any encrypted images, which increases the difficulty of statistical attacks. In addition, due to the high sensitivity to the initial parameters of our three chaotic cards, and the broadcast installed in each tower confirmed the robustness of our encryption system.

## 5. CONCLUSION

Due to their high sensitivity to initial conditions, chaotic systems are widely used in color image encryption. with an improvement of the Vigenere matrices, we have, in our strategy, developed two alternative matrices based on two

chaotic maps for the execution of two Vigenere towers. Two start-up settings were computed to initiate the process of diffusing confusion between the encrypted block and the next clear block, to significantly augment the avalanche action and to prevent the system from known differential attacks. All the statistical constant values derived from our analysis can ensure that our software is not exposed to known attacks.

## REFERENCES

- [1] Rachmawanto, E.H., De Rosal, I.M.S., Sari, C.A., Agus Santoso, H., Rafrastara, F.A., Sugiarto, E. (2019). Block-based Arnold chaotic map for image encryption. 2019 International Conference on Information and Communications Technology (ICOIACT), pp. 174-178. <https://doi.org/10.1109/ICOIACT46704.2019.8938443>
- [2] Bansal, R., Gupta, S., Sharma, G. (2017). An innovative image encryption scheme based on chaotic map and Vigenere scheme. Multimedia Tools and Applications, 76: 16529-16562. <https://doi.org/10.1007/s11042-016-3926-9>
- [3] Jarjar, A. (2017). Improvement of hill's classical method in image cryptography. International Journal of Statistics and Applied Mathematics, 2(3): 37-43.
- [4] 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.
- [5] Reddy, V.V.K., Bhukya, S. (2018). Encrypt and decrypt image using Vigenere cipher. International Journal of Pure and Applied Mathematics, 118(24): 1-8.
- [6] Kester, Q.A. (2012). A cryptosystem based on Vigenere cipher with varying key. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 1(10): 108-113.
- [7] Dewangga, I.G.A.P., Purboyo, T.W., Nugrahaeni, R.A. (2017). A new approach of data hiding in BMP image using LSB steganography and Caesar Vigenere cipher cryptography. International Journal of Applied Engineering Research, 12(21): 10626-10636.
- [8] Rahmani, K.I., Wadhwa, N., Malhotra, V. (2012). Alpha-Qwerty cipher: An extended Vigenere cipher. Advanced Computing, 3(3): 109. <https://doi.org/10.5121/acij.2012.3311>
- [9] Boussif, M., Aloui, N., Cherif, A. (2020). Securing DICOM images by a new encryption algorithm using Arnold transform and Vigenere cipher. IET Image Processing, 14(6): 1209-1216. <https://doi.org/10.1049/iet-ipr.2019.0042>
- [10] Peng, J., Liao, X., Wu, Z. (2002). Digital image secure communication using Chebyshev map chaotic sequences. In IEEE 2002 International Conference on Communications, Circuits and Systems and West Sino Expositions, pp. 492-496. <https://doi.org/10.1109/ICCCAS.2002.1180666>
- [11] Hraoui, S., Gmira, F., Jarar, A.O., Satori, K., Saaidi, A. (2013). Benchmarking AES and chaos based logistic map for image encryption. In 2013 ACS International Conference on Computer Systems and Applications (AICCSA), pp. 1-4. <https://doi.org/10.1109/AICCSA.2013.6616441>
- [12] François, M., Grosge, T., Barchiesi, D., Erra, R. (2012). A new image encryption scheme based on a chaotic



- function. *Signal Processing: Image Communication*, 27(3): 249-259. <https://doi.org/10.1016/j.image.2011.11.003>
- [13] Shah, A. (2016). Enhancing security of Vigenere cipher using modified RC4. *International Journal of Computer Applications*, 136(5): 38-41. <https://doi.org/10.5120/ijca2016908428>
- [14] 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. <https://doi.org/10.1016/j.optlaseng.2018.12.002>
- [15] Ge, R., Yang, G., Wu, J., Chen, Y., Coatrieux, G., Luo, L. (2019). A novel chaos-based symmetric image encryption using bit-pair level process. *IEEE Access*, 7: 99470-99480. <https://doi.org/10.1109/ACCESS.2019.2927415>
- [16] Jarjar, M., Najah, S., Zenkour, K., Hraoui, S. (2020). Further improvement of the HILL method applied in image encryption. In 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), pp. 1-6. <https://doi.org/10.1109/IRASET48871.2020.9092046>
- [17] Saputra, I., Mesran, 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.
- [18] Zhang, L., Zhang, X. (2020). Multiple-image encryption algorithm based on bit planes and chaos. *Multimedia Tools and Applications*, 79(29): 20753-20771. <https://doi.org/10.1007/s11042-020-08835-4>
- [19] Enayatifar, R., Guimarães, F.G., Siarry, P. (2019). Index-based permutation-diffusion in multiple-image encryption using DNA sequence. *Optics and Lasers in Engineering*, 115: 131-140. <https://doi.org/10.1016/j.optlaseng.2018.11.017>
- [20] Belazi, A., El-Latif, A.A.A., Belghith, S. (2016). A novel image encryption scheme based on substitution-permutation network and chaos. *Signal Processing*, 128: 155-170. <https://doi.org/10.1016/j.sigpro.2016.03.021>
- [21] Hua, Z., Zhou, Y., Pun, C.M., Chen, C.L.P. (2015). 2D Sine Logistic modulation map for image encryption. *Information Sciences*, 297: 80-94. <https://doi.org/10.1016/j.ins.2014.11.018>

## NOMENCLATURE

### Notation

$$\left\{ \begin{array}{l} G_t = \mathbb{Z}/_t\mathbb{Z} \text{ ring} \\ G_t^* = \text{Set of } G_t \text{ reversers} \\ \oplus \text{ Binary addition} \\ A(j): \text{Line number } j \text{ of matrix } A \\ A(:,j): \text{column number } j \text{ of matrix } A \end{array} \right.$$