

Pixel Permutation Using Chaotic Perfect Shuffle Technique for Image Encryption

Ernastuti Ernastuti¹, Ravi A Salim¹, Sulisty Puspitodjati¹

¹ Department of Computer Science and Engineering, Gunadarma University, Indonesia

Abstract – A proposed algorithm for digital image encryption is presented in this paper. It combines permutation and substitution techniques. Logistic map which is a chaos function is used in both techniques. Before encrypting, the keystream is generated first from the logistics map. Then, the plain image's pixel positions are shuffled with perfect shuffle permutation based on ascending keystream order, thereby executing the permutation process. Next, the pixel values are substituted using the XOR operation with a keystream which is also generated from the logistic map. Pixels are operated in cipher block chain mode. This research aim is to develop an image encryption algorithm, especially in the permutation process, that has high resistance to attacks by crackers. The attack types observed in this research include statistical attacks, permutation matrix attacks, differential attacks, as well as brute force attacks. From the experimental results and analysis of the proposed algorithm indicate that it has high resistance from all those attacks.

Keywords - Chaotic, image encryption, logistic map, permutation, perfect shuffle.

1. Introduction

The increasing urgency to guarantee that image information in communications is secure and confidential, image transmission is increasingly on demand.

DOI: 10.18421/TEM132-10

<https://doi.org/10.18421/TEM132-10>

Corresponding author: Ernastuti Ernastuti,
Department of Computer Science and Engineering,
Gunadarma University, Indonesia


Email: ernas@staff.gunadarma.ac.id

Received: 16 January 2024.

Revised: 14 April 2024.

Accepted: 06 May 2024.

Published: 28 May 2024.

 © 2024 Ernastuti Ernastuti, Ravi A Salim & Sulisty Puspitodjati; published by UIKTEN. This work is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License.

The article is published with Open Access at <https://www.temjournal.com/>

Information is visually presented by images, and the image-presented information enriches the textually-presented information. An example is satellite imagery that depicts the wealth of a country.

Image encryption methods that are common like the Arnold and Magic transformations, protect the image information through distribution changes in the pixels, but their achieved security is not high [1].

Researchers have developed many cryptographic algorithms for encryption, but mostly text-message-encrypting in nature. It should be noted, however, that some conventional encryption algorithms do encrypt images, although they are not efficient to implement. Such algorithms are DES, AES, RSA, Rabin, among others. The reason behind this is that textual data and images have distinct characteristics. Very large data capacity is involved in storing an image; hence a large computing volume should be served for its encryption. Meanwhile, some applications need to be real-time, thus requiring lightning speeds in computing, making conventional algorithms out of mode in image encryption [2]. Examples of such applications are teleconferences, live video streaming, etc. Apart from volume reasons, the characteristic that differentiates images from text is the correlation of data between neighbors. The data in the text only neighbor the data before (predecessor) and after (successor), whereas in the image the pixels are neighbors with other pixels in eight cardinal directions, making the high correlation with the pixels in all eight directions. Therefore, after an image is encrypted, what must be paid attention to is that the cipher-image pixels and adjacent pixels should have zero correlation [2].

Recent years witness a lot of focus to chaos topics such as [3], [4], [5], [6], [7]. Initial conditions are influential to chaotic systems, in fact the latter is extremely sensitive to the former. Thus, excellent encryption properties resulted from that fact. Stronger security and lower predictability are achieved through the use of chaotic encryption systems [8].

Among others in the algorithm, chaos-theory based image encryption techniques has as their main methods, pixel values confusion and diffusion.

These aim at securing the image and resisting many brute attacks [9]. It is well-known that image has some specific natures like having adjacent pixels which correlate strongly, having data that require large space and others. These natures make the conventional encryption cryptosystems like DES and AES face great challenges in research of image encryption. It is urgent to design more effective encryption methods for protecting image information [2], [10].

Typical scheme for image encryption is that it contains substitution and permutation stages aiming at fulfilling confusion and diffusion characteristics which are due to Shannon [11]. The latter or permutation stage alters the places by employing chaotic or non-chaotic generators [12], [13], [14]. The former or substitution stage alters the pixels values employing a generator of pseudo random number or two other techniques [15], [16], [17].

A perfect shuffle is defined as an n element permutation or order so that each of its application produces a new order or returns to a previous order. Thus, the applications at some point give the original order. It is noticeable that there are $n!$ permutations of $n!$ elements [18]. There is a recently proposed algorithm which is a technique based on perfect shuffle, hence its name: Perfect Shuffle Crypto Algorithm (PSCA). In the crypto system, the PSCA is categorized as a permutation or transposition technique [19]. There is a video encryption algorithm that is computationally efficient and secure that prompts the feasibility for encryption designed for real time applications avoiding complicated computational requirements and shortening key management through employing block shuffling technique. The algorithm is called Faro perfect shuffle [20].

To attempt the fulfilment of the Shannon's characteristics, it is indispensable for a technique of image encryption to embody permutations and substitutions. Two of three permutation techniques employing discrete chaotic maps are discussed in [21], these are: permutation vector, and discrete chaos. They display various scrambling properties therefore promise encryption breakthroughs. This paper deals with an image encryption via perfect shuffle permutation for confusion and XOR substitution operator for diffusion. The two phases are logistic map based, by integrating it to form a symmetric key cryptosystem, in which the decryption and encryption processes are inverses to each other.

This paper describes and compares two perfect shuffle permutation techniques, namely with non-chaotic generator and with chaotic generator. Section 2 explains the materials as well as methods forming the foundation of the research.

In Section 3, a comparison is presented between chaotic and non-chaotic based permutation shuffle exchange techniques. Section 4 discusses the analysis of the image encryption security system performance. Then, Section 5 concludes the paper.

2. Material and Methods

This section discusses the plain, scrambled, and cipher images, as well as the techniques, the model, and the measures applied to them along with their reasons.

2.1. Permutation Technique

The process where all pixels move from their original location is called the permutation phase. Obviously, it must be bijective since the number of pixels are finite. By doing so, the resulting image can be restored into the original. The bijection is representable as a matrix T with entry $T(i, j)$ indicating that the pixel of position i is mapped to position j . Thus, a size $M \times N$ permutation matrix T is given by:

$$T = \begin{cases} T_{ij}, T_{ij} \in \{1, 2, \dots, M \times N\}; \\ T_{ij} \text{ are distinct, } i \in \{1, \dots, M\}, j \in \{1, \dots, N\} \end{cases} \quad (1)$$

Suppose then the pixels of the image are processed top-bottom and left-right, by the following calculation (2) for the new pixel location:

$$\begin{aligned} v_{new} &= \text{div}(T_{ij} - 1, M) + 1 \\ w_{new} &= \text{mod}(T_{ij} - 1, M) + 1 \end{aligned} \quad (2)$$

where v_{new} and w_{new} are the new column and row indices.

Here, div and mod give respectively the quotient and remainder of integer division $\frac{T_{ij}-1}{M}$, for which $i \in \{1, \dots, M\}$ and $j \in \{1, \dots, N\}$ signify the row and column indices of the matrix T respectively. Let the permutation matrix T as follow.

$$T = \begin{pmatrix} 11 & 10 & 14 & 1 \\ 13 & 15 & 5 & 2 \\ 7 & 4 & 9 & 3 \\ 12 & 8 & 16 & 6 \end{pmatrix}$$

Let the pixel processing indices in a block of size 4×4 be as in figure 1 (a). For each pixel, the new location which is calculated by equation (2) based on matrix T , is presented in Figure 1 (b). As an example, when $i = 1$ and $j = 1$, then the permutation matrix entry is $T_{11} = 11$ and by (2), $v_{new} = 3$ and $w_{new} = 3$.

This indicates that at location (1,1), in the original block the pixel is now moved to the new block's location (3, 3).

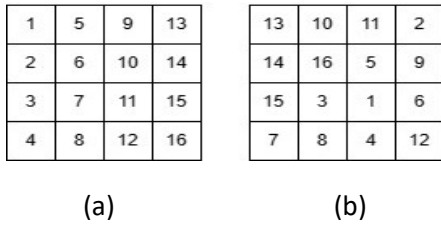


Figure 1. (a) original block's pixel indices, and (b) new pixel locations after applying (2a) and (2b) formulas

2.2. Perfect Shuffle Network Model

Shuffle and Exchange are two routing functions on which Perfect Shuffle network model is based. In perfect shuffle, link nodes *i* and *j* are connected as follow:

$$j = 2 * i \quad , 0 \leq i \leq 2^n / 2 - 1 \quad (3)$$

$$= 2 * i + 1 - 2^n \quad , other$$

where $n = \log_2 N$ (*N*: the number of processors)

Functions representing shuffle exchange are implementable as either a network that recirculates or a network with many stages.

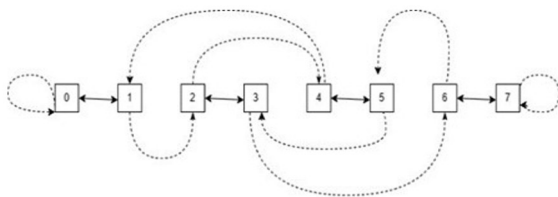


Figure 2. A network for *N* = 8 representing shuffle exchange

Shuffle connections are denoted by the dashed arrows, while the exchange connections are denoted by the solid arrows. Figure 2 depicts a one-stage network representing recirculating shuffle exchange for *N* = 8. The name of perfect shuffle originates from the following. Suppose an eight-card deck with cards numbered by is shuffled. Then the shuffle result is 0,4,1,5,2,6,3,7. After that the exchange is obtained by pairwise interchanging from left to right, resulting in 4,0,5,1,6,2,7,3. Figure 3 illustrates the shuffle exchange permutation process.

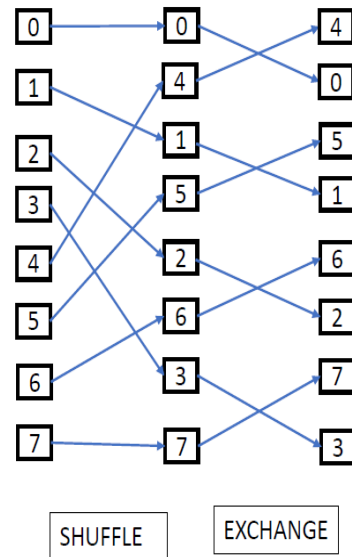


Figure 3. Perfect shuffle permutation

Figure 3 presents a perfect shuffle with *N* = 8 processors. When $N = 2^n$, a datum is obtained back to its original location after *n* times of shuffling operations [22]. Furthermore, when $N = 2^n$, a datum returns to its original location after 2*n* times of shuffle exchange operations [19], [23]. In other word the perfect shuffle permutation has a cycle period.

2.3. Logistic Map

Chaos-based cryptography is an interesting research topic nowadays. There are three reasons for Chaos to be employed in cryptography: (1) Chaos' sensitive nature to the system's initial conditions, (2) chaos behaves randomly, and (3) the lack of periods for chaotic values. There are many chaos functions. An example of a chaos function is the logistic map, which is conventional. It is defined as follows.

$$x_{n+1} = r * x_n (1 - x_n) \quad (4)$$

where $0 < x_n < 1$ and $0 < r \leq 4$.

The density in the orbital period of a chaotic system can be seen using a bifurcation diagram, which is a diagram to illustrate the possible values for each parameter, such as the initial value parameter. The bifurcation diagram is reconstructed by drawing a plot of the system based on its parameters. Figure 4 displays the logistic map's bifurcation diagram function. From Figure 4, when $3.56995 < r \leq 4$ the map shows chaotic behavior.

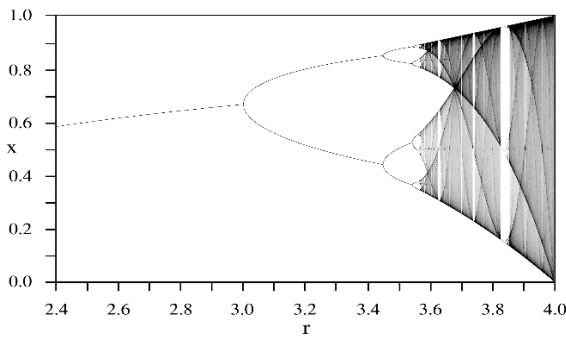


Figure 4. Bifurcation diagram of the logistic map

2.4. Image Cryptosystem Security Measures

This subsection discusses measures in some types of attacks on cryptosystems, which are used to analyze the performance of resistance from encryption algorithms.

2.4.1. Security Analyses from Statistics

2.4.1.1. Histogram Analysis

A histogram is an important image feature, because it exhibits the intensity of the distribution of the image pixel. In performing attacks using techniques from statistical analysis, the attacker uses a histogram in order to analyze occurrence frequency of the intensity of the pixels to deduce the plain image's key or pixels, so that attacks with statistical analysis are not possible. In image encryption, it is important to produce a cipher image histogram which is not similar statistically to the histogram of the plain-image. Therefore, the cipher image pixels need to have distribution as near as possible to being uniform so that the histogram appears flat. Figure 5(a) displays the 'Lena' image's histogram, while Figure 5(b) is the cipher image histogram.

The histogram for the cipher image looks flat unlike the histogram for the plain image.

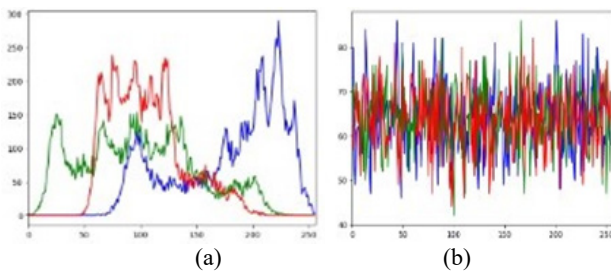


Figure 5. (a) Plain Lena histogram, and (b) cipher Lena histogram

2.4.1.2. Adjacent Pixels Correlation Analysis

Given two random variables x and y , one way of measuring their linear relationship's direction and strength is through their correlation coefficient cor_{xy} .

If x and y are discrete stochastic variables of size n , (sometimes called random variables), then cor_{xy} is given by

$$cor_{xy} = \frac{kov(x, y)}{\sqrt{dev(x) * dev(y)}} \quad (5)$$

where

$$kov(x, y) = \frac{1}{n} \sum_{i=1}^n [x_i - E(x)][y_i - E(y)]$$

$$dev(x) = \frac{1}{n} \sum_{i=1}^n [x_i - E(x)]^2$$

$$E(x) = \frac{1}{n} \sum_{i=1}^n x_i$$

The correlation coefficient value cannot exceed 1 in absolute value. A correlation coefficient value of +1 shows a linear perfect unidirectional relationship (correlation) so that an increase in one variable is perfectly mimicked by the other and vice versa, a correlation coefficient value of -1 shows a linear perfect opposite relationship (correlation) so that increase in one variable is followed by a decrease of the same amount and vice versa, while between -1 and +1 shows the degree of perfectness of linear dependence among the two variables. Positive value indicates unidirectionality, while negative value, opposite direction. The magnitude itself indicates the proportion of the effect of change of one variable towards the other. A correlation coefficient value approaching to -1 or +1 shows a near perfect linear unidirectional or opposite directional relationship between the two variables, while a coefficient value approaching 0 shows the weak version.

Most plain images, have correlation coefficient between their adjacent pixels usually high (namely approaching 1 or -1). Image encryption aims to reduce the absolute value of the correlation coefficient between pixels to as near as possible to zero. For an ideal cryptosystem, the adjacent pixels correlation coefficients for cipher-image should be close to zero to effectively resist statistical analysis attacks.

2.4.2. Security Analysis of Permutation Matrix

Two mean distance tests are evaluated in this paper for comparison, one of them is the mean distance one pixel moved, since the measurements involve only the images after one pixel is moved, and the other being is the mean distance between the new places of two adjacent pixels, since the measurements involve only distances among new places in the scrambled image. Evaluation for each test employs the Euclidean distance formula.

The Euclidean distance formula between two pixels P_i dan P_j is

$$L_{ij} = \sqrt{(v_i - v_j)^2 + (w_i - w_j)^2} \quad (6)$$

where the column and row indices for pixel P_i inside the image are written as v_i and w_i respectively.

The greater the displacement distance of two pixels from the original image, the more difficult it is for the cracker to guess the original image.

2.4.3. Differential Analysis Attack

To evaluate the robustness of cryptosystems resisting differential analysis attack, there are two indices commonly used, namely NPCR and UACI, each stand for Number of Pixels Change Rate and Unified Average Changing Intensity.

NPCR measures the percentage of pixels that differ when the original I and the cipher K images are compared. The formula is:

$$NPCR = 100\% \times \frac{1}{M \times N} \sum_{i,j} dist(i,j) \quad (7)$$

$$where \ dist(i,j) = \begin{cases} 1, & when \ I(i,j) \neq K(i,j), \\ 0, & when \ I(i,j) = K(i,j), \end{cases}$$

for which, $I(i,j)$ and $K(i,j)$ are pixel values in the plain and cipher images respectively.

The NPCR index is applied to calculate the pixel percentage possessing distinct intensity values when two images are compared.

Pixel rate of change within one pixel from the plain to the cipher images in the modification process is measured by NPCR since its value reflects effectivity in performance. As an accepted norm, 0.99 is the practical value for 1-NPCR. In other words, the higher the NPCR value is, the more effective the performance is.

UACI index measures the difference in mean intensity from a plain image denoted by I to its cipher image denoted by K . The formula is

$$UACI = 100\% \times \frac{1}{M \times N} \sum_{i,j} \frac{|I(i,j) - K(i,j)|}{255} \quad (8)$$

UACI means to calculate the average change rate of each pixel value between two images. The theoretical expectation value of UACI is 33.46%.

In statistics, MAE which stands for mean absolute error, measures two continuous variables' difference. Thus, MAE can judge the change from the plain image I to the cipher image K . If MAE is large enough, then the encryption effect is more secure. The formula for MAE is

$$MAE = \frac{1}{M \times N} [\sum_{i,j} |I(i,j) - K(i,j)|] \quad (9)$$

The computation of MAE can be viewed as follows. First the value is computed only for the red pixels, then for the green ones, then for the blue ones. Afterwards, the average results are taken by summing them up and dividing the number by 3.

3. Perfect Shuffle Technique

There are two kinds of perfect shuffle technique. One is non chaotic in nature and the second is chaotic. This section explores each of them.

3.1. Non-Chaotic Shuffle Exchange Permutation

In this section, an experiment using perfect shuffle permutation is performed to pixel indices of a plain image without any relation to keystream of a chaos function. Afterwards the resulting image is observed, and then its histogram and adjacent pixels correlation aspects are analyzed. Apart from that its average distance when one pixel is moved and pairwise average distance among adjacent pixels are also analyzed. The non-chaotic shuffle exchange permutation algorithm is presented as Algorithm 1 as follows.

Algorithm 1: Non-chaotic Shuffle Exchange Permutation

1. Input the plain image of size $M \times M$.
 2. Represent image indices as a matrix of size $M \times M$.
 3. Transform the matrix into a vector of size $1 \times M^2$ by putting its columns into a single sequence.
 4. Perform the shuffle exchange process.
 5. Transform the result back into matrix form by forming a sequence of column vectors of size M .
 6. Present the scrambled image.
-

For example, let a plain pixel indices matrix T_1 of size $M \times M$:

$$T_1 = \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix}$$

where $M = 4$.

Transform the matrix T_1 into a vector X of size $1 \times M^2$ by column as follows. $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$. After that perform the shuffle exchange permutation on X . The output is a sequence of indices which equals Y , where $Y = \{2, 4, 6, 8, 10, 12, 14, 16, 1, 3, 5, 7, 9, 11, 13, 15\}$, and then transform Y into a matrix T_1 , so that T_1 is now a shuffle exchange permutation matrix.

$$T_1 = \begin{pmatrix} 2 & 10 & 1 & 9 \\ 4 & 12 & 3 & 11 \\ 6 & 14 & 5 & 13 \\ 8 & 16 & 7 & 15 \end{pmatrix}$$

Figure 6 (a) displays the pixel indices of plain image and Figure 6 (b) displays the new location of image pixel indices after shuffling with nonchaotic shuffle exchange permutation for $M=4$.

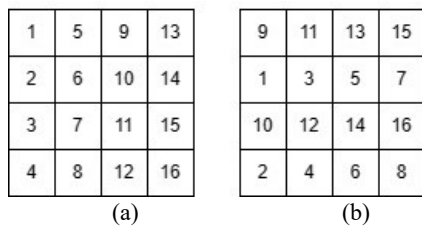


Figure 6. (a) pixel Indices of plain Image and (b) new pixel location after applying nonchaotic shuffle exchange permutation algorithm

Plain images and scrambled images are analyzed in histograms and adjacent pixel correlations. The histogram and adjacent pixel correlation for plain Lena and plain Baboon are explained in Figure 7 (a) and Figure 8 (a) respectively. Meanwhile, the histogram and correlation of adjacent pixels for scrambled Lena and scrambled Baboon are depicted in Figure 7 (b) and Figure 8 (b) respectively.

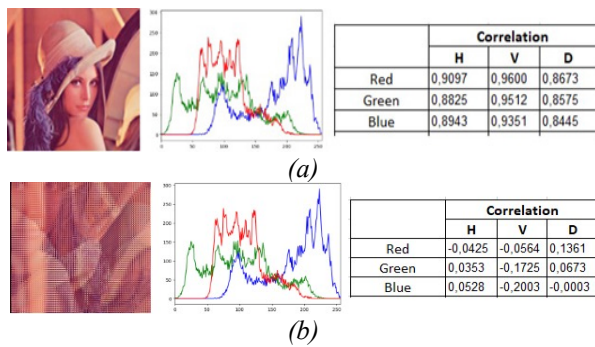


Figure 7. Lena Image, histogram and correlation of adjacent pixels for (a) Plain, (b) Scrambled images of nonchaotic shuffle exchange

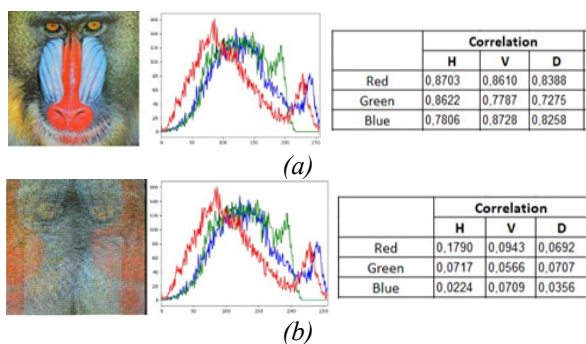


Figure 8. Baboon image, histogram, and correlation of adjacent pixels for (a) Plain, (b) Scrambled images

In Figure 7(a) and Figure 8(a) explain that the adjacent pixels' correlation respectively for the Horizontal (H), Vertical (V), and Diagonal (D) directions is close to 1. This indicates that in the plain image, between its pixels there is a strong correlation. Meanwhile, in scrambled images, the correlation coefficient is nearly zero. This indicates that adjacent pixels are no longer correlated.

In general, the adjacent pixels' average correlation in any direction, in scrambled Lena and scrambled Baboon is calculated by averaging the total correlation values for the 3 RGB colors namely red, green, and blue. The average results are listed in Table 1. The algorithm 1 effectively resists statistical analysis attacks.

Table 1. Adjacent pixels correlation of shuffle exchange for scrambled Lena and baboon (averaged over the 3 color channels)

| Adjacent Pixels Correlations | | | |
|------------------------------|------------|----------|----------|
| Scrambled | Horizontal | Vertical | Diagonal |
| Lena | 0.0435 | 0.1430 | 0.0679 |
| Baboon | 0.0910 | 0.0739 | 0.0585 |

The adjacent pixel parameters are good (nearly zero), but the scrambled Lena in Figure 7 (b) and the scrambled Baboon in Figure 8 (b), visually still look a little like their respective plain images. The scrambled image is still recognizable. This means that the nonchaotic random exchange permutation algorithm is insecure and can be hacked by attackers.

3.2. Chaotic Shuffle Exchange Permutation

The use of nonchaotic shuffle exchange permutations in Algorithm-1 has the potential to cause insecure image encryption because it has a cycle period. If the pixel positions in an image are continuously scrambled, then one day the resulting image will be the same as the original image. Additionally, Algorithm 1 is visually insecure for the scrambled images.

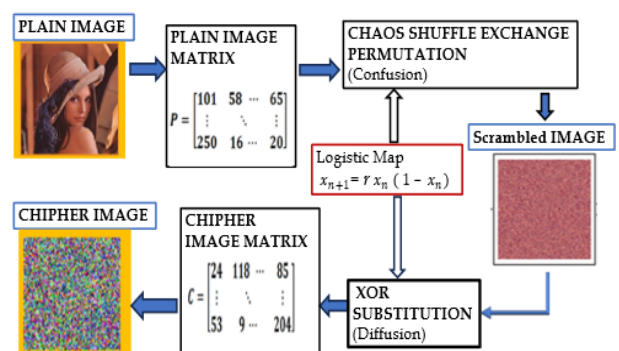


Figure 9. Block diagrams of the chaotic perfect shuffle encryption system

To overcome the insecurity problem of the permutation phase in Algorithm 1, this paper proposes an image encryption system (Fig. 2.8) which employs Perfect shuffle permutation in the confusion stage, and XOR substitution in the diffusion phase.

Both phases are based on a logistic map integrated in a symmetric key cryptosystem, in which the decryption and the encryption processes are inverses to each other.

3.2.1. Chaotic Perfect Shuffle Permutation

A plain image is encrypted by first scrambling it whole using the chaotic shuffle exchange permutation as in Algorithm 2.

Algorithm 2: Chaotic Shuffle Exchange Permutation

1. Input the plain image of size $(M \times M)$.
2. Represent image indices in the form of a matrix of size $(M \times M)$.
3. Transform the matrix into a vector of size $(1 \times M^2)$ by putting its columns into a single sequence.
4. Generate the M^2 keystream from the Logistic Map Function.
5. Put the keystream into one-to-one correspondence to the vector of plain image indices.
6. Arrange the keystream into ascending order.
7. Arrange the positions of the plain image indices according to the ascending keystream.
8. Perform the shuffle exchange.
9. Transform the result back into matrix form by forming a sequence of column vectors of size M .
10. Present the scrambled image.

For example, let a plain pixel indices matrix T_2 of size $M \times M$:

$$T_2 = \begin{pmatrix} 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \\ 4 & 8 & 12 & 16 \end{pmatrix}$$

where $M = 4$.

Transform the matrix T_2 into a vector X of size $1 \times M^2$ by column. It will be $X = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$. Then, generate 4×4 keys from the logistic map $x_{n+1} = r x_n (1 - x_n)$. The logistic map is repeated 16 times so that a 4×4 permutation matrix T_2 is produced. Suppose $r = 0.3999$ and $x_0 = 0.200$, then the logistic map output is $X' = \{0.6398, 0.9215, 0.2891, 0.8219, 0.5853, 0.9706, 0.1141, 0.4042, 0.9630, 0.1423, 0.4884, 0.9992, 0.3162, 0.1261, 0.4978, 0.1892\}$.

After arranging X' in ascending order, the plain image index positions are mapped to produce $Y = \{11, 13, 7, 12, 10, 15, 4, 8, 14, 5, 9, 16, 1, 2, 3, 6\}$, and then transform Y into a matrix T_2 , so that T_2 is now a discrete chaos permutation matrix.

$$T_2 = \begin{pmatrix} 11 & 10 & 14 & 1 \\ 13 & 15 & 5 & 2 \\ 7 & 4 & 9 & 3 \\ 12 & 8 & 16 & 6 \end{pmatrix}$$

Afterwards, shuffle exchange permutation is applied to Y , so that the plain image index positions are mapped to produce $Z = \{5, 9, 14, 7, 3, 13, 8, 16, 11, 10, 1, 15, 2, 4, 6, 12\}$, and then transform Z into a matrix T_2 , so that T_2 is now a chaos shuffle exchange permutation matrix.

$$T_2 = \begin{pmatrix} 5 & 3 & 11 & 2 \\ 9 & 13 & 10 & 4 \\ 14 & 8 & 1 & 6 \\ 7 & 16 & 15 & 12 \end{pmatrix}$$

Figure 10 (a) displays the pixel indices of the plain image and Figure 10 (b) shows new location of pixel indices after shuffling with chaotic shuffle exchange permutation for $M = 4$.

| | | | |
|---|---|----|----|
| 1 | 5 | 9 | 13 |
| 2 | 6 | 10 | 14 |
| 3 | 7 | 11 | 15 |
| 4 | 8 | 12 | 16 |

(a)

| | | | |
|----|----|----|----|
| 11 | 1 | 2 | 6 |
| 13 | 15 | 10 | 3 |
| 5 | 4 | 9 | 12 |
| 14 | 7 | 16 | 8 |

(b)

Figure 10. (a) pixel indices of plain image and (b) new pixel location after applying chaotic shuffle exchange permutation algorithm.

3.2.2. XOR Substitution

The encryption process is formulated as follows

$$a_i = b_i \mathbf{xor} k_i \tag{10}$$

where a_i : i -th pixel value of cipher image; b_i : i -th pixel value of scrambled image; k_i : i -th keystream.

The integer of keystream generated from the logistic map function with the initial value x_0 and parameter r is operated with the scrambled image pixel values via XOR operator as equation (10).

However, the random values generated from the logistic map are real numbers. Therefore, these values have to be expressed as integers. For this, a simple transformation that takes the decimal part of a real number and discard the insignificant zeros, then extract q whole number digits, can be utilized.

As an example, take $q = 4$ and $x_i = 0.003162812$, then taking the decimal part results in 003162812, removing the two zeros that are not significant in front of it, then extracting 4 digits, ends up in 3162.

This is the keystream that will be XORed with the i -th pixel. Since the pixel values are within the integer range of $[0, 255]$, then before being XORed the keystream is first modulated by 256. In this case, $k_i = 3162 \text{ mod } 256 = 90$.

4. Security Analysis and Experimental Results

The encryption system depicted in Figure 9 is simulated using Jupiter Lab with Python programming on CPU AMD Ryzen 3 3250U and RAM 8 GB. The experiment was run on two test images, namely the 'Lena' image and the 'Baboon' image, each of which had 3 different sizes (64×64), (128×128) and (256×256). The main parameters used to produce the keystream logistic map in the experiment are: $r = 0.3989$ and $x_0 = 0.6295$ for permutation process and $s = 0.3898$ and $y_0 = 0.5434$ for substitution process.

4.1. Pixel Average Distances in the Scrambled Image

The average distances of one pixel moved indicates the mean of how far a pixel has moved from its original location. Additionally, the average distance of two adjacent pixels in the scrambled image indicates the mean distance between each other resulting from the movements of adjacent pixels, which are presented horizontally, vertically, and diagonally respectively. The average distance values of Table 2 show that the pixels have been scattered.

Table 2 indicates that the chaotic shuffle exchange permutation algorithm provides better results on the average pixel displacement distance, because almost all indicators achieve a greater average distance than the nonchaotic shuffle exchange permutation algorithm.

Table 2. Average distances of three sizes Lena

| Scrambled LENA | Matrix (M) | One Pixel Moved | Between Two Adjacent Pixels | | |
|----------------|------------|-----------------|-----------------------------|----------|----------|
| | | | Horizontal | Vertical | Diagonal |
| Nonchaotic | 64 | 24,4953 | 32,0076 | 20,4643 | 45,1419 |
| | 128 | 48,9771 | 64,0038 | 64,0038 | 90,3934 |
| | Exchange | 256 | 97,9473 | 128,1566 | 93,0618 |
| Chaotic | 64 | 33,2838 | 33,4593 | 32,7364 | 33,5058 |
| | 128 | 66,5933 | 66,4522 | 66,4061 | 66,6672 |
| | Exchange | 256 | 133,5415 | 133,0872 | 133,1689 |

4.2. Histogram and Adjacent Pixels Correlation

4.2.1. Histogram

The pixels in Figure 11c, namely the Lena cipher and in Figure 12c, namely the Baboon cipher, each have a relatively uniform distribution. These cipher pixels are depicted with a histogram that appears flat.

Lena Image

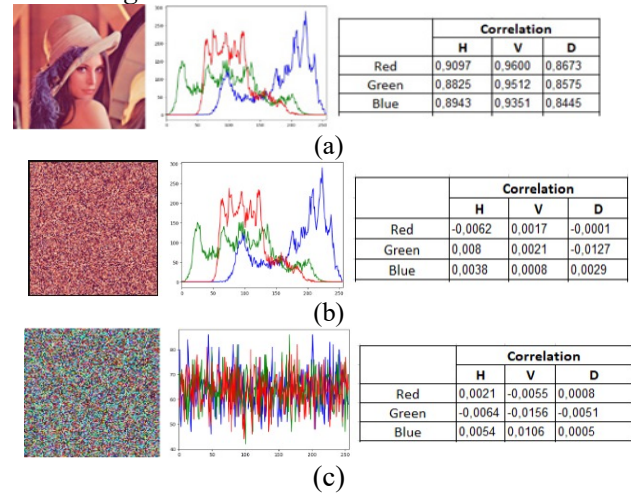


Figure 11. Lena image, histogram, and adjacent correlations for (a) Plain, (b) Scramble, (c) Encrypted/cipher

The histograms for the cipher images in Figures 11c and 12c each appears differently from the histograms for the plain images in Figures 11a and 12a. This indicates that attacks with statistical analysis will not be possible to work in the encryption system which produces a cipher image histogram that is not statistically similar to histogram of the plain image.

Baboon Image

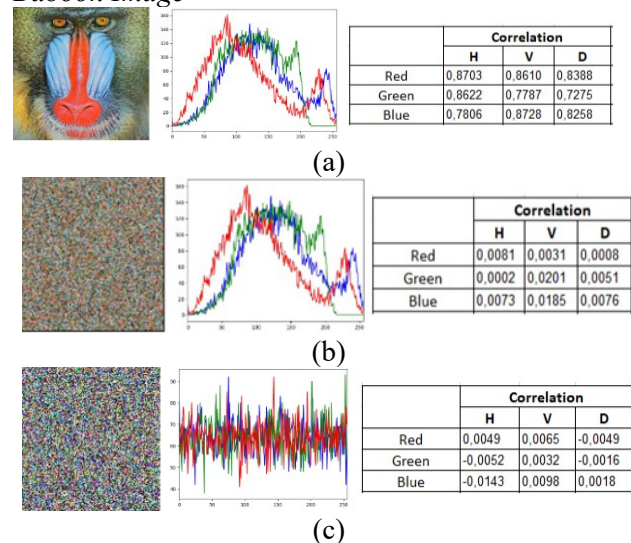


Figure 12. Baboon Image, Histogram and Adjacent Correlations for (a) Plain, (b) Scramble, (c) Encrypted/cipher

4.2.2. Adjacent Pixel Correlations

To investigate the correlation coefficient among adjacent pixels in plain-image and cipher-image, the chaotic shuffle exchange algorithm calculates from between two horizontally adjacent pixels, two vertically adjacent pixels, and two diagonally adjacent pixels in both images respectively.

This is applied to both the 'Lena' and 'Baboon' images for permutation and substitution processes where each produces the scrambled image and the cipher image. The correlation coefficient for adjacent pixels is calculated by employing Formula (5). The correlation values of the experiment result to examine the proposed algorithm are listed in Table 3.

Table 3. Adjacent pixels correlation of chaotic shuffle exchange

| Adjacent Pixels Correlations | | | |
|------------------------------|------------|----------|----------|
| Scrambled | Horizontal | Vertical | Diagonal |
| Lena | 0.0060 | 0.0015 | 0.0052 |
| Baboon | 0.0052 | 0.0139 | 0.0045 |
| Encrypted/chiper | Horizontal | Vertical | Diagonal |
| Lena | 0.0046 | 0.0106 | 0.0021 |
| Baboon | 0.0081 | 0.0065 | 0.0027 |

The adjacent pixels correlation values in Table 3 are obtained from the average of the sum of all adjacent pixels' correlation values for all the three color channels, namely red, green, and blue respectively in each three directions. The correlation values are presented in Figures 11 (b) and 12 (b) for the scrambled images, and in Figures 11 (c) and 12 (c) for the cipher images. Table 3 indicates that the adjacent pixel correlation coefficients in each direction in both the scrambled as well as the cipher images are nearly zero. This means that the neighboring pixels are virtually no longer correlated. In addition, it appears that their coefficient in the scrambled image resulting from permutation of chaotic shuffle exchange algorithm is 10 times weaker than the result from nonchaotic shuffle exchange algorithm (To compare, see Table 1). This indicates that the resistance of chaos shuffle exchange algorithm against the statistical analysis attack is very strong.

4.3. MAE, NPCR and UACI Measures

Table 4 lists the simulation results of MAE, NPCR, and UACI measures.

Table 4. Measures for differential attacks averaged from the three color channels

| Encrypted/Chiper Image | MAE | NPCR | UACI |
|------------------------|---------|---------|---------|
| Lena | 77,2255 | 99,5809 | 33,6179 |
| Baboon | 75,0869 | 99,6073 | 33,2845 |

The average MAE value is 76.1562. This is a fairly large MAE value. Hence, the proposed image encryption system effect is accordingly more secure.

The average values of UACI and NPCR of the two encrypted images in this experiment are 3.4512% and 99.5941% respectively. They approximate perfect values with differentiation. This proves that in resisting differential attacks, they are highly sensitive according to the standards of the proposed encryption. In other words, Table 4 demonstrates that the proposed image encryption system well resists differential analysis attacks.

4.4. Key Space

In the encryption/decryption process, the overall number of different keys involved is stated by the key space. To incapacitate the efficiency of a brute force attack, the key space must be large enough. There are four secret key parameters (initial value parameters) used in this encryption algorithm, namely r and x_0 in the logistic map for the permutation process, and s and y_0 in the logistic map for the substitution process. These are four real valued parameters where each can be computed in 10^{-15} order of 64-bit double precision in IEEE floating point standard. So, the number of possible initial values of logistic map is 10^{15} . Therefore, in this research the entire key space used in image encryption is $K(x_0, r, y_0, s) = 10^{15} * 10^{15} * 10^{15} * 10^{15} = 10^{60}$, large enough to survive brute-force attacks.

5. Conclusion

The experimental results of chaotic perfect shuffle permutation algorithm on color images shows that, firstly, the cipher image has flat histograms. This indicates that the pixels are distributed uniformly, so that attacks with statistical analysis are not possible. Besides that, the cipher image's adjacent pixels' correlation coefficient is low, namely around 0.00576. This indicates that these pixel values no longer have a linear relationship so that also incapacitating statistical analysis attacks. Secondly, permutation matrix analysis shows that the pixels are very scattered, so that the cipher image is not recognizable at all as being related to the plain image. Thirdly, the cipher image has average MAE= 75.20, PNCR= 99.60 and UACI= 33.40; it indicates the image encryption of chaotic perfect shuffle is very good and tough in facing differential analysis attacks. Then lastly, the key space has large number of possible keys; it shows that it is sufficiently tough in withstanding brute-force attacks. In other words, the chaotic perfect shuffle permutation algorithm which is proposed in this research is capable of encrypting digital images with high security in resisting the four attacks.

References:

- [1]. Dong, H., Lu, P., & Ma, X. (2013). Image Encryption Algorithm Based on CNN Hyper Chaotic System And Extend Zigzag Transformation. *Computer Applications and Software*, 30(5), 133-137.
- [2]. Ye, R., Ren, T., & Tan, B. (2022, September). A Chaotic Image Encryption Algorithm Using Pixel Permutation-diffusion and Bit Replacement. In *2022 8th Annual International Conference on Network and Information Systems for Computers (ICNISC)* 341-346. IEEE.
- [3]. Hu, G., Kou, W., & Peng, J. (2018). A Novel Image Encryption Algorithm Based on Cellular Neural Networks Hyper Chaotic System. *IEEE proceeding of 4th International Conference on Computer and Communications*, 1878-1882.
- [4]. El Assad, S., & Farajallah, M. (2016). A new chaos-based image encryption system. *Signal Processing: Image Communication*, 4(1), 144-157.
- [5]. Waghmare, A., Bhagat, A., Surve, A., & Kalgutkar, S. (2016). Chaos Based Image Encryption and Decryption. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(4), 64-68.
- [6]. Huang, L., Shi, D., & Gao, J. (2016). The Design and Its Application In Secure Communication and Image Encryption of a New Lorenz-Like System with Varying Parameter. *Mathematical Problems in Engineering*, MPIE, 1-11.
- [7]. Li, G. D., Zhao, G. M., Xu, W. X., & Yao, S. (2014). Research on application of Image Encryption Technology Based on Chaotic of Cellular Neural Network. *Journal of Digital Information Management*, 12(2), 151-158.
- [8]. Rashid A. A., & Hussein K. A (2023). Image encryption algorithm based on the density and 6D logistic map. *International Journal of Electrical and Computer Engineering*, 13(2), 1903-1913.
- [9]. Chaudhary N., Shahi T. B., & Neupane, A. (2022). Secure Image Encryption Using Chaotic, *Hybrid Chaotic and Block Cipher Approach*. *Journal of Imaging*, 8(6), 167
- [10]. Ye, G., Pan, C., Huang, X., & Mei, Q. (2018). An efficient pixel-level chaotic image encryption algorithm. *Nonlinear Dynamics*, 94, 745-756.
- [11]. Alvarez, G., & Li, S. (2006). Some basic cryptographic requirements for chaos-based cryptosystems. *International journal of bifurcation and chaos*, 16(08), 2129-2151.
- [12]. Zhang, A., Zhou, N., & Gong, L. (2013). Color Image Encryption Algorithm Combining Compressive Sensing with Arnold Transform. *J. Comput.*, 8(11), 2857-2863.
- [13]. Zhang, Y., & Xiao, D. (2014). Self-adaptive permutation and combined global diffusion for chaotic color image encryption. *AEU-International Journal of Electronics and Communications*, 68(4), 361-368.
- [14]. Abdeihaleem, S. H., Radwan, A. G., & Abd-El-Hafiz, S. K. (2014). A chess based chaotic block cipher. *The 12th IEEE International New Circuits and Systems Conference (NEWCAS)*, 405-408.
- [15]. Abd-El-Hafiz, S. K., Radwan, A. G., & AbdEl-Haleem, S. H. (2015). Encryption applications of a generalized chaotic map. *Applied Mathematics & Information Sciences*, 9(6), 3215.
- [16]. Barakat, M. L., Mansingka, A. S., Radwan, A. G., & Salama, K. N. (2013). Generalized Hardware Post-processing Technique for Chaos-Based Pseudorandom Number Generators. *ETRI journal*, 35(3), 448-458.
- [17]. Abd-El-Hafiz, S. K., Radwan, A. G., Abdel Haleem, S. H., & Barakat, M. L. (2014). A fractal-based image encryption system. *IET Image Processing*, 8(12), 742-752.
- [18]. Stone, H. S. (1971). Parallel processing with the perfect shuffle. *IEEE transactions on computers*, 100(2), 153-161.
- [19]. Margonda, J. (2014). Perfect Shuffle Algorithm for Cripthography. *ARPN Journal of Engineering and Applied Sciences*, 9(12), 2384-2386.
- [20]. Sultana, S. F., & Shubhangi, D. C. (2017). Video encryption algorithm and key management using perfect shuffle. *International Journal of Engineering Research and Applications*, 7(2), 1-5.
- [21]. Abd-El-Hafiz, S. K., AbdElHaleem, S. H., & Radwan, A. G. (2016). Permutation techniques based on discrete chaos and their utilization in image encryption. In *2016 13th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 1-6. IEEE.
- [22]. Quinn J.M. (1987). Designing Efficient Algorithms for Parallel Computers. *McGraw Hill International Editions*, 26-28.
- [23]. Das, N., Bhattacharya, B. B., Menon, R., & Bezrukov, S. L. (1998). Permutation admissibility in shuffle-exchange networks with arbitrary number of stages. In *Proceedings. Fifth International Conference on High Performance Computing (Cat. No. 98EX238)*, 270-276. IEEE.