Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2009, Article ID 762652, 22 pages
doi:10.1155/2009/762652

Research Article

Chaotic Image Encryption Design Using Tompkins-Paige Algorithm

Shahram Etemadi Borujeni[1, 2] and Mohammad Eshghi[1]

[1] *Computer Engineering Department, Faculty of Computer and Electrical Engineering, Shahid Beheshti University, Evin, Tehran 1983963113, Iran*
[2] *Computer Engineering Department, Faculty of Engineering, University of Isfahan, Isfahan 8174673441, Iran*

Correspondence should be addressed to Shahram Etemadi Borujeni, etemadi@eng.ui.ac.ir

Received 1 February 2009; Revised 29 April 2009; Accepted 14 July 2009

Recommended by Elbert E. Neher Macau

In this paper, we have presented a new permutation-substitution image encryption architecture using chaotic maps and Tompkins-Paige algorithm. The proposed encryption system includes two major parts, chaotic pixels permutation and chaotic pixels substitution. A logistic map is used to generate a bit sequence, which is used to generate pseudorandom numbers in Tompkins-Paige algorithm, in 2D permutation phase. Pixel substitution phase includes two process, the tent pseudorandom image (TPRI) generator and modulo addition operation. All parts of the proposed chaotic encryption system are simulated. Uniformity of the histogram of the proposed encrypted image is justified using the chi-square test, which is less than $\chi^2(255, 0.05)$. The vertical, horizontal, and diagonal correlation coefficients, as well as their average and RMS values for the proposed encrypted image are calculated that is about 13% less than previous researches. To quantify the difference between the encrypted image and the corresponding plain-image, three measures are used. These are MAE, NPCR, and UACI, which are improved in our proposed system considerably. NPCR of our proposed system is exactly the ideal value of this criterion. The key space of our proposed method is large enough to protect the system against any Brute-force and statistical attacks.

1. Introduction

In any communication system, including satellite and internet, it is almost impossible to prevent unauthorized people from eavesdropping. When information is broadcasted from a satellite or transmitted through the internet, there is a risk of information interception. Security of image and video data has become increasingly important for many applications including video conferencing, secure facsimile, medical and military applications. Two main groups of technologies have been developed for this purpose. The first group is content protection through encryption, for which a key is required for proper decryption of the data. The second group is digital watermarking, which aims to embed a message into the multimedia data. These two technologies could be used complementary to each other [1, 2].

In secured communications using encryption, which is the focus of the present work, the information under consideration is converted from the intelligible form to an unintelligible structure using certain operations at the transmitter. Data encryption is mainly scrambling the content of data, such as text, image, audio, and video to make the data unreadable, invisible or incomprehensible during transmission. The unintelligible or encrypted form of the information is then transmitted through the insecure channel, that is, internet, to the destination. At the intended recipient side, however, the information is again converted back to an understandable form using decryption operation and thus the information is conveyed securely. It should be noted that the same keys guide both these encryption and decryption operations. Such encryption system is grouped under private key cryptography [1, 3].

In particular, an image-scrambling scheme transforms an image into another unintelligible image, based on keys only known to the senders and the receivers. The fundamental techniques to encrypt a block of pixels are substitution and permutation. Substitution replaces a pixel with another one; permutation changes the sequence of the pixels in a block to make them unreadable.

In recent years, chaotic maps have been employed for image encryption. Most chaotic image encryptions (or encryption systems) use the permutation-substitution architecture. These two processes are repeated for several rounds, to obtain the final encrypted image. For example, in [4], Fridrich suggested a chaotic image encryption method composed of permutation and substitution. All the pixels are moved using a 2D chaotic map. The new pixels moved to the current position are taken as a permutation of the original pixels. In the substitution process, the pixel values are altered sequentially. Chen et al. employed a three-dimensional (3D) Arnold cat map [5] and a 3D Baker map [6] in the permutation stage. Guan et al. used a 2D cat map for pixel position permutation and the discretized Chen's chaotic system for pixel value masking [7]. Lian et al. [8] used a chaotic standard map in the permutation stage and a quantized logistic map in the substitution stage. The parameters of these two chaotic maps are determined by a key stream generated in each round. Mao et. al. construct a new image encryption scheme based on the extended chaotic Baker map [6]. Zhang et al. first permute the pixels of images with discrete exponential chaotic map, and then use "XOR plus mod" operation for substitution [9]. Gao et al. present the image encryption algorithm based on a new nonlinear chaotic algorithm using a power function and a tangent function instead of a linear function. It also uses a chaotic sequence generated by a nonlinear chaotic algorithm to encrypt image data using XOR operation [10]. Zhou et al. propose a parallel image encryption algorithm using discretized kolmogorov flow map. All the pixels are first permuted with a discretized chaotic map and then encrypted under the cipher block chain mode [11].

There are however some other chaotic image encryption systems with different structures. For example, Pisarchik and Zanin suggested an algorithm to convert image pixels to chaotic maps coupled to form a chaotic map lattice. The encrypted image is obtained by iterating the chaotic map lattice with secret system parameters and number of cycles [12]. Pareek et al. extended the concept of their text encryption to image encryption by using two logistic maps and a key [13].

In this paper, a new permutation-substitution architecture using chaotic maps and Tompkins-Paige algorithm is proposed. Our designed technique for speech scrambling [14] is extended to two-dimensional (2D) permutation and is applied to image permutation [15]. We have improved our work by using chaotic maps and adding a substitution part to an image encryption system. In the permutation phase, a logistic map is used to generate a bit sequence, which is used to generate Pseudorandom numbers in Tompkins-Paige algorithm. A tent map is also used in the substitution phase to product a Pseudorandom image that is used to mix it with the permuted image. The permutation and substitution operations need two different keys, *Key-P* and *Key-S*, respectively. Satisfactory security performance of the proposed system is achieved in only one round and therefore the total encryption time is short .

The paper is organized as follows. In Section 2, principles of chaotic cryptography including chaotic maps and chaotic encryption are introduced. The proposed chaotic encryption systems using logistic random bit sequence generator, Tompkins-Paige algorithm, and tent Pseudorandom image generator are described in Section 3. In Section 4, simulation results of the proposed image encryption systems are presented. Finally, the security analysis is explained in Section 5.

2. Principle of Chaotic Cryptography

The word cryptography refers to the science of keeping secrecy of information exchanged between a sender and a receiver over an insecure channel. The objective is achieved by data encryption so that only individuals who have the key can decrypt it. The key K, in a typical encryption system, determines the transformation from the set of all possible samples, to the set of all possible permuted samples. An encryption system is a finite set T of transformations from a finite sample space M onto a permuted sample space C. It means that each of the transformations in T must be reversible, so that if a sample m is transformed into the permuted sample c by transformation t, $c = t(m)$, then the sample m is $m = t^{-1}(c)$, where t^{-1} is the inverse transform of t [16, 17].

In practice, we need to transmit a reasonable amount of information, which requires a large sample space and that in turn implies a large number of keys. The distribution of a large number of keys is liable to cause horrendous management problems. In a practical system, a cryptanalyst will have to worry about time and facilities. Often, the time taken to solve a permuted sample will be of utmost importance. It is quite likely that the samples need to be secret for a limited period of time, referred to as required cover time. Thus, it is certainly possible for a theoretically insecure system to provide adequate practical security [1]. If we set the cryptanalyst a task requiring a large amount of storage, or sufficiently large number of operations, then we may regard our system as practically secure.

With the desirable properties of ergodicity and high sensitivity to initial conditions and control parameters, chaotic maps are suitable for various data encryption schemes. In particular, chaotic maps are easy to be implemented using microprocessors or personal computers. Therefore, chaotic encryption systems generally have high speed with low cost, which makes them better candidates than many traditional ciphers for multimedia data encryption. There are two types of chaotic encryption systems: chaotic stream encryption systems, and chaotic block encryption systems. In chaotic stream encryption systems, a key stream is produced by a chaotic map, which is used to encrypt a plain-text bit by bit. A chaotic block encryption system, on the other hand, transforms a plain-text block by block with some chaotic maps [8].

2.1. Chaotic Maps

In this subsection, we consider nonlinear and chaotic one-dimensional maps $f : S \longrightarrow S$, where $S \subset R$. The set S is S = $[0, 1]$. The one-dimensional dynamical system can be defined by a difference equation similar to

$$x_{k+1} = f(x_k), \quad k = 0, 1, 2, \ldots, x_k \in S, \tag{2.1}$$

where the variable k stands for time. A dynamical system consists of a set of possible states, together with a deterministic rule, which means that the present state can be determined uniquely from the past states. The orbit of x under f is the set of points $\{x, f(x), f^2(x), \ldots, f^n(x)\}$, where $f^2(x) = f(f(x))$ and $f^n(x)$ means n times iterating of the function $f(x)$. The starting point x for the orbit is called the initial value of the orbit. A chaotic orbit is one that forever continues to experience the unstable behavior that an orbit exhibits near a source, but that is not itself fixed or periodic [18]. Two well known one-dimensional chaotic maps are tent and logistic maps.

The iterative relation of the tent map is given by [18].

$$x_{k+1} = \begin{cases} \dfrac{x_k}{p}, & \text{if } 0 \le x_k \le p, \\ & \\ \dfrac{1 - x_k}{1 - p}, & \text{if } p \le x_k \le 1, \end{cases} \quad \text{where } x_k \in [0, 1], \tag{2.2}$$

where x_0 is the initial condition and p is the control parameter. The tent map is chaotic if p is in the range of $(0, 1)$ and $p \neq 0.5$. Figure 1(a) shows a sample return map of a tent map.

Logistic map is a one-dimensional quadratic map defined by

$$x_{k+1} = a x_k (1 - x_k), \quad x_k \in [0, 1], \tag{2.3}$$

where a is the control parameter, and x_0 is the initial condition. The control parameter a should be taken in the range of $[3.6, 4]$ to keep the logistic map chaotic. Figure 1(b) shows a sample return map of a logistic map.

The logistic equation involves two multiplications and one subtraction per iteration, while the tent equation includes one division and on average one subtraction. Meanwhile, the tent map has better chaotic behavior than the logistic map. As mentioned above, the range of control parameter (p) of the tent map is about twice the range of the logistic map. However, hardware implementation of the logistic map is simpler. We have used both these in our proposed system to improve security.

2.2. Chaotic Encryption Scheme

Due to the tight relationship between chaos and cryptography, the use of chaotic maps to construct an encryption system has been widely investigated [19]. There are three typical ways of using chaos in an image encryption.

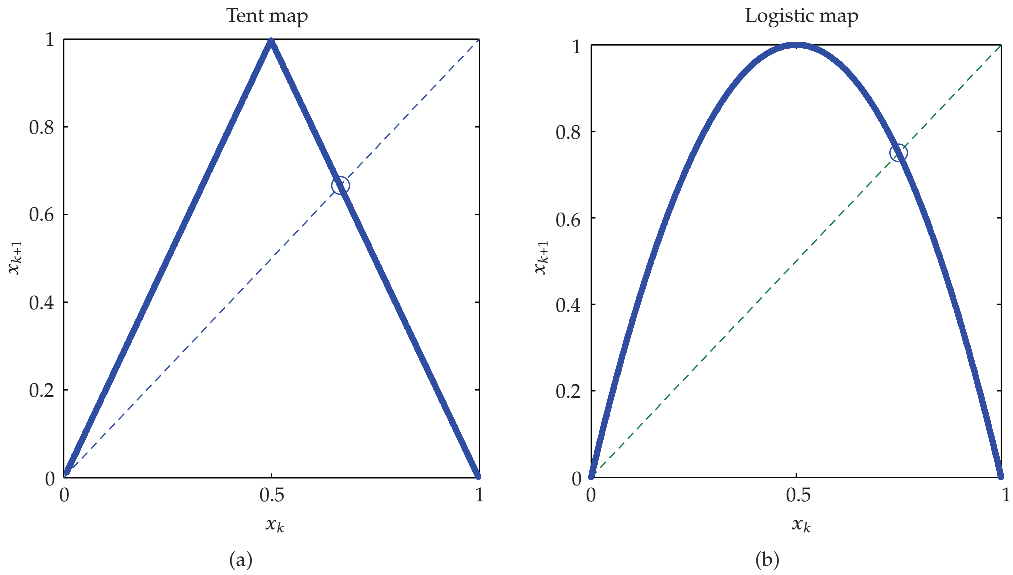


Figure 1: (a) Tent return map ($p = 0.5$). (b) Logistic return map ($a = 4$).

(1) Using chaos as a source to generate Pseudorandom bits with desired statistical properties to realize a secret permutation operation [6, 7, 20].

(2) Using chaos as a source to generate Pseudorandom pixels with desired statistical properties to realize a secret substitution operation [5, 21–23].

(3) Using two chaotic maps in both permutation and substitution [8, 11, 12].

The fundamental techniques to encrypt a block of symbols are confusion and diffusion. Confusion can make ambiguous the relationship between the plain-text and the cipher-text. Diffusion can spread the change throughout the whole cipher-text. Substitution, which replaces a symbol with another one, is the simplest type of confusion, and permutation that changes the sequence of the symbols in the block is the simplest method of diffusion. These techniques together are still the foundations of encryption [3].

2.2.1. Chaotic Permutation

In designing private key cryptographic techniques, permutation methods are considered as important building blocks in conjunction with Pseudorandom sequence generators for selecting a specific permutation key. First, a *Key-P* is entered as a binary number equivalent to the given key. Then, a 1Dimentional chaotic map generates a random bit-string. Subsequently, a permutation matrix for the system is calculated.

A permutation matrix is an identity matrix with the rows and columns interchanged. It has a single 1 in each row and column; all the other elements are 0. For example,

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{2.4}$$

Any vector A is multiplied by the permutation matrix in order to rearrange its elements. For example from 1-2-3-4 to 4-1-3-2 as shown in

$$P \cdot A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} = \begin{bmatrix} a_4 \\ a_1 \\ a_3 \\ a_2 \end{bmatrix}. \tag{2.5}$$

For simplicity, the 4 × 4 matrix p could be expressed as a 4-element vector Q as shown in

$$Q = \begin{bmatrix} 4 & 1 & 3 & 2 \end{bmatrix}. \tag{2.6}$$

Therefore, the $L \times L$ permutation matrix of the L elements could be expressed as an L-element vector Q for simplicity (2.7), where $1 \leq k_i \leq L$ [16].

$$Q = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 & \cdots & k_L \end{bmatrix}. \tag{2.7}$$

Each element in the vector Q shows the new position of each element in the permuted vector. The elements of the vector are rearranged in a new order according to K_i's in the simplified permutation matrix, Q. Practically, the element at the k_ith position is moved to the ith position, respectively. By permutation P on a set of L elements, A,

$$\begin{aligned} A &= \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & \cdots & a_L \end{bmatrix}^T, \\ P \cdot A &= P \cdot \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & \cdots & a_L \end{bmatrix}^T \\ &= \begin{bmatrix} d_1 & d_2 & d_3 & d_4 & \cdots & d_L \end{bmatrix}^T, \end{aligned} \tag{2.8}$$

where $d_i = a_{k_i}$, for $i = 1, 2 \ldots, L$ (as an example: if $k_1 = 4$ then $d_1 = a_4$).

2.2.2. Chaotic Substitution

In cryptography, a substitution cipher is a method of encryption by which blocks of plain text are replaced with cipher-text according to a regular system; the blocks may be single or several letters. The receiver deciphers the text by performing an inverse substitution. Substitution ciphers can be compared with permutation ciphers. In a permutation cipher, the blocks of the plain-text are rearranged in a different and usually quite complex order, but the blocks themselves are left unchanged. By contrast, in a substitution cipher, the blocks of the plain-text are retained in the same sequence as in the cipher-text, but the blocks themselves are altered.

A permutation-only encrypted system is insecure against attacks [24]. To improve the security, substitution process is added to the encryption system. The substitution could be one of simple operations such as XOR, XNOR, shift, Add, and/ or a combination of these simple operations. Chaotic map is used as generation of Pseudorandom image for substitution.

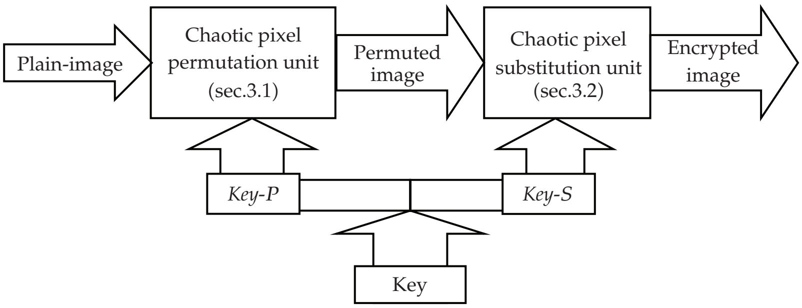

Plain-image
Chaotic pixel permutation unit (sec.3.1)
Permuted image
Chaotic pixel substitution unit (sec.3.2)
Encrypted image
Key-P
Key-S
Key

Figure 2: Block diagram of the chaotic image encryption system.

Actually, chaotic image with a size equal to plain-image is generated. All pixels of permuted image and new chaotic image are combined with modular addition. Substitute operation decreases the correlation between blocks or samples in text and makes its histogram uniform.

3. Design of a Chaotic Image Encryption System

The block diagram of the proposed chaotic image encryption system is illustrated in Figure 2. This system includes two major units, chaotic pixels permutation unit and chaotic pixels substitution unit. Two different dynamical systems, that is, logistic and tent maps are also considered to generate a more complicated key and consequently a highly secure encryption system. In this paper, logistic map is used as a Pseudorandom bit generator while tent map is utilized to generate a Pseudorandom image generator. Pixels of a plain-image are rearranged by the permutation unit. The permutation unit uses a chaotic bit generator and Tompkins-Paige algorithm, to implement a $1D$ and $2D$ image permutation. The pixels of the permutated image are then changed in the chaotic pixels substitution unit. The substitution unit is used for the modular addition of the permuted image with a Pseudorandom image. A key that is used for the encryption system includes *Key-P* and *Key-S*. *Key-P* is used as the initial value and control parameter of a logistic map, which is utilized to generate random bit sequences. *Key-S* is applied to the tent map, as an initial value and the control parameter. Tent map is used to generate a Pseudorandom image. More details of each unit are explained in the following sections.

3.1. Chaotic Pixel Permutation Unit

It is assumed that L pixels are expected to be permuted. The number of possible permutations for L pixels is $L!$, however not all permutations can be used. The Hamming distance is the number of elements moved by the permutation. The more number of elements moved by the permutation, the larger the Hamming distance. Meanwhile, permutation matrixes that are close to any circularly shifted versions of the identity order produce a permuted sample of high closeness to the original sample. Since typically, we might wish to have a choice of about N permutation matrixes, the number of key bit, M, should be selected such that $N < 2^M - 1$. In the encryption system, the number of pixels to be permuted L is assumed to be 128, since the image size 128. Therefore, the number of all possible permutations is $128!(3.85E + 215)$ [25, 26].

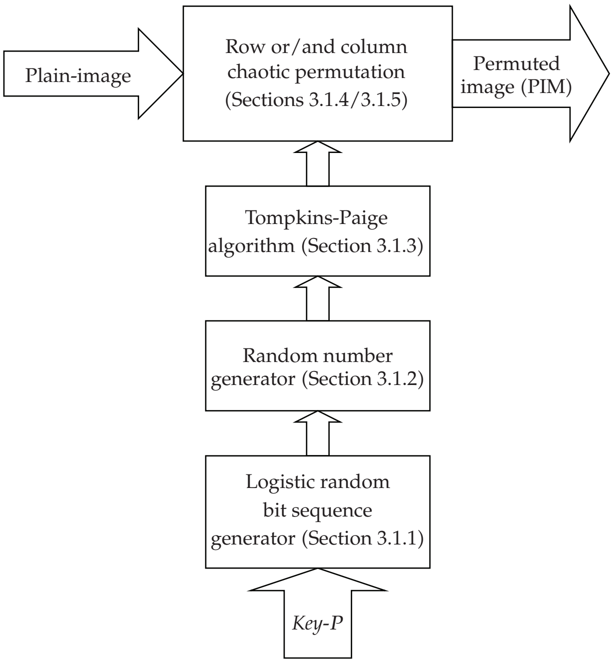

Plain-image
Row or/and column chaotic permutation (Sections 3.1.4/3.1.5)
Permuted image (PIM)
Tompkins-Paige algorithm (Section 3.1.3)
Random number generator (Section 3.1.2)
Logistic random bit sequence generator (Section 3.1.1)
Key-P

Figure 3: Block diagram of the chaotic pixel permutation unit.

There are three steps in the design of the permutation subsystem, which are explained as follows. First, a 90 *bit* key is entered as a binary number equivalent to the given sub key *Key-P*. It is used as the initial value (26 *bits*) and control parameter (64 *bits*) of a logistic random bit generator. Then, a random bit string is generated. Subsequently, integer Pseudorandom numbers are calculated according to its range and Tompkins-Paige algorithm. Finally, Tompkins-Paige algorithm is applied to provide a permutation matrix for permutation of the pixels.

The block diagram of the system is illustrated in Figure 3. Three subsections, logistic random bit sequence (LRBS) generator, Pseudorandom number calculator, and Tompkins-Paige algorithm perform pixel permutation. An LRBS generator is needed as a first stage of the permutation matrix generator. An initial key is used as an initial value and control parameter of the logistic map. A bit-string is generated by LRBS and the integer Pseudorandom numbers are calculated by (3.1). The Tompkins-Paige algorithm is used to generate the target permutation matrix, which is obtained from repetition of some simple permutations.

Chaotic pixel permutation is used as the target permutation matrix to implement 1D and 2D image permutation [27].

3.1.1. Logistic Random Bit Sequence (LRBS) Generator

Generation of the chaotic random bit sequence is done as follows [28]. An appropriate chaotic map is selected. The logistic map is general and simple as mentioned in Section 2. The probability density function of logistic is not uniform, but by introducing a proper threshold level, the output of the bit sequence becomes uniform. The control parameter and initial value of the map is determined. Then, a real value is generated by each iteration, which is converted into a bit by a single level threshold function. The threshold value is calculated using a computer simulation. Different values are considered and the occurrence of 0 and 1 is examined. The threshold is selected to 0.6 such that the probability of frequencies of 0 (f_0) and frequencies of 1 (f_1) is approximately equal. A sample histogram of a logistic map with $a = 3.9$ and $x_0 = 0.5$ is shown in Figure 4. The initial 90 *bit* Key- P contains a 26 *bit* initial value and a 64 bit control parameter. A string of 768*bits*, $\{b_0 \sim b_{767}\}$ is generated in 768 iterations of LRBS. If the real output of logistic map in the specific iteration is less than 0.6, the output bit of LRBS is 0, otherwise it is 1.

3.1.2. Pseudorandom Number Generator

Let b_i $(i = 0, 1, 2, \ldots)$ be the ith output bit of the LRBS, which is generated according to the initial key, *Key-P*. $L - 1$ integer Pseudorandom numbers, g_i's $(i = 1, 2, \ldots, L - 1)$ are calculated using these b_i's, as shown in [26]

$$\begin{aligned} g_1 &= 1, \\ g_2 &= \left[\frac{(2b_0 + b_1)(1)}{(2^2 - 1)}\right] + 1, \\ g_3 &= \left[\frac{(2b_2 + b_3)(2)}{(2^2 - 1)}\right] + 1, \\ g_i &= \left[\frac{\left(2^{j-1}b_k + 2^{j-2}b_{k+1} + \cdots + b_{k+j-1}\right)(i - 1)}{(2^j - 1)}\right] + 1, \end{aligned} \tag{3.1}$$

where

$$j = \lfloor \log_2 i \rfloor + 1, \qquad k = \sum_{s=2}^{i-1} \left(\lfloor \log_2 s \rfloor + 1\right). \tag{3.2}$$

$\lfloor x \rfloor$ denotes the floor of x. Since the number of permuted pixels ($L = 128$) is equal to the image size, 127 integers (g_i) and 768 *bits* (b_i) are required. It is obvious that the maximum value of every g_i is i, $(g_i < i + 1)$.

3.1.3. Tompkins-Paige Algorithm

Tompkins-Paige algorithm gives a one-to-one correspondence between the integers and the permutation. As an example, the simple permutation of nine elements of *order* 7 and *degree* 3 is shown in the second row. The last 7 elements are disturbed and an end-around shift of 3 elements to the left are performed

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 6 & 7 & 8 & 9 & 3 & 4 & 5 \end{bmatrix}. \tag{3.3}$$

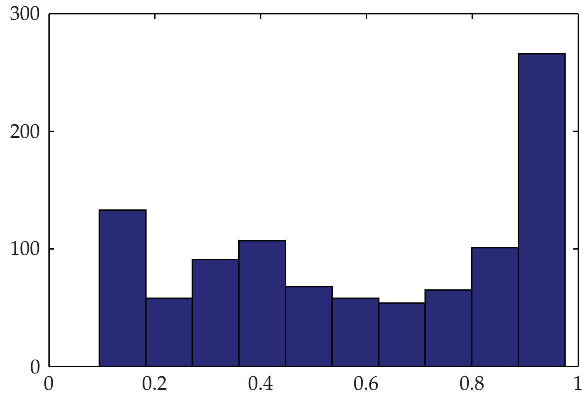


Figure 4: A sample histogram of a logistic map with $a = 3.9$ and $x_0 = 0.5$.

Therefore, the simple permutation of *order* $L - m_1 + 1$ and *degree* $m_2 - m_1$ is generally defined as shown in the matrix [26]

$$\begin{bmatrix} 1 & 2 & \cdots & m_1 - 1 & m_1 & m_1 + 1 & \cdots & m_2 - 1 & m_2 & m_2 + 1 & \cdots & L \\ 1 & 2 & \cdots & m_1 - 1 & m_2 & m_2 + 1 & \cdots & L & m_1 & m_1 + 1 & \cdots & m_2 + 1 \end{bmatrix}. \tag{3.4}$$

In the above example, $m_1 = 3$, $m_2 = 6$, and $L = 9$.

In this example, the target permutation of these 9 elements can be obtained through compounding 8 simple permutations of *orders* 9 to 2, and for each order an associated degree. In each simple permutation, degree should be less than its order. In General, the target permutation on L elements is the result of compounding $L - 1$ simple permutations with order of L to 2 and degree of g_i, where g_i is less than the corresponding order in each simple permutation.

In this paper, the Tompkins-Paige algorithm is applied to $L = 128$ elements with order of 128 to 2 and degree of g_i. Finally, the 128×128 permutation matrix of the 128 elements is expressed in a 128 elements vector [27].

As mentioned earlier, $L = 128$ is the number of pixels, to be permuted and g_i's are calculated using (3.1), where, $g_i < i + 1$.

3.1.4. 1*D Chaotic Pixel Permutation*

The main idea behind the present work is that an image can be viewed as an arrangement of 2D pixels [29]. The intelligible information present in an image is due to the correlations among the pixels in a given arrangement. This perceivable information can be reduced by decreasing the correlation among the pixels using certain random permutation techniques.

The image can be seen as a 2*D* array of pixels, each with 256 gray scales. In pixel permutation techniques the pixels taken from the image are permuted with the key chosen from the key space. There are two options in 1*D* permutation process: row permutation or column permutation.

In row permutation, according to Figure 3, the *Key-P* is used to generate the permutation matrix. The pixels of all rows are rearranged with respect to the permutation matrix/vector, as explained in Section 2.2.1. The result of these permutations is discussed in Section 4.

In column permutation, the pixels of all columns are rearranged with respect to the permutation matrix/vector according to the *Key-P*. The result of these permutations is also presented in Section 4.

3.1.5. 2D Chaotic Pixel Permutation

We extend the basic concept of $1D$ Chaotic pixel permutation in order to design a $2D$ permutation method. Here row and column permutation are applied simultaneously. In the $2D$ permutation of the image, the permutation matrix of the rows and columns could be either identical, using the identical *Key-P*, or different, using the dissimilar *Key-P*.

In identical permutation approach, the pixels of all rows are first rearranged, with respect to the permutation matrix/vector and the pixels of all columns are then rearranged with respect to the same permutation matrix/vector. The encrypted images that appear as a random noisy image are shown in Section 4.

In a different permutation approach, the pixels of all rows are first rearranged, with respect to the first permutation matrix/vector and the pixels of all columns are then rearranged, with respect to the second permutation matrix/vector. The encrypted images that appear as a random noisy image are also shown in Section 4.

3.2. Chaotic Pixel Substitution Unit

In the permutation part of the system, pixel positions are displaced without changing their gray level values. Hence, the histogram of the permuted image is similar to the histogram of the plain-image. The permuted image however cannot resist against ”statistical” and “known plain text” attacks [6]. To improve the security of the proposed encryption system, its histogram needs to approximate the uniform distribution. This improvement is done using a substitution scheme, shown in Figure 5.

There are two main subunits in the substitution unit, tent Pseudorandom image generator and modulo addition. A 128 *bits* key (*Key-S*) is entered to the tent map, 64 *bits* as an initial value of the map and the remaining bits as the control parameter. Tent map is used to generate a Pseudorandom image. In each iteration of the tent map, a Pseudorandom number between 0 to 1 is generated. For each 128 × 128 image, 16 384 iterations are required. Since a 256 gray-scale image is selected, the Pseudorandom number should be linearly transformed to the range of $[0, 255]$. Then, the permuted image is modularly added with random image pixel-wise in mod 256. The substitution procedure decreases a correlation between pixels and makes the histogram more uniform.

3.2.1. Tent Pseudorandom Image (TPRI) Generator

There are two options to generate a chaotic Pseudorandom image. A chaotic random generator along with a simple threshold detector similar to Section 3.1 could be utilized. Afterward, every 8 *bits* stream should be converted to a gray scale of a pixel. As a second option, a chaotic random generator along with a linear transform may be used. The transformer is employed to convert a real range of $[0, 1]$ to an integer range of $[0, 255]$

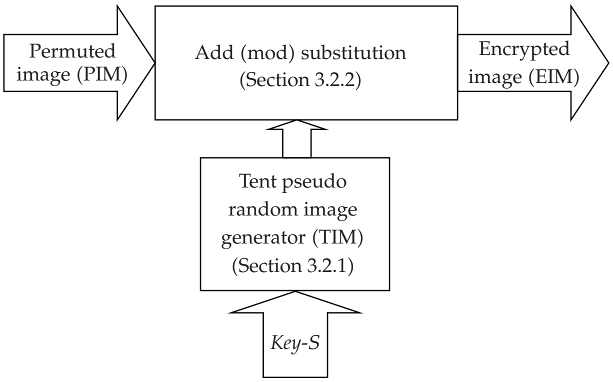

Permuted image (PIM)
Add (mod) substitution (Section 3.2.2)
Encrypted image (EIM)
Tent pseudo random image generator (TIM) (Section 3.2.1)
Key-S

Figure 5: Block diagram of the chaotic pixel substitution unit.

linearly. It is completed by introducing 255 threshold levels. As the latter seems to be faster, modification of chaotic random generator has been done as follows:

(1) Tent map is chosen as a chaotic system instead of a logistic map, since its probability density function (PDF) is uniform and implementation is almost simple.

(2) Control parameter and initial condition of the map is determined by *Key-S*. Each of them is defined with 64 *bits* and a simple linear transformation.

(3) Real values of chaotic sequences are generated by iterations of the map: $x_0, x_1, x_2, \ldots, x_{(nxn)}$ where n is the image size.

(4) 255 threshold levels in the range $[0, 1)$ are defined and a gray scale of pixels from 0 through 255 are attributed to them, respectively.

TPRI output seems to be a noisy image and its histogram is uniform.

3.2.2. Modulo Addition for Chaotic Pixel Substitution

It is desirable to decrease intelligibility of the encryption image. That is achievable with a substitute operation such that the final histogram becomes uniform and correlation between pixels is reduced. The permuted image could therefore be mixed with a noise image, TPRI. Modulo addition/subtraction is more suitable than XOR/XNOR operation. In this research, modulo 256 additions are performed. At the encryption side, the 2*D* permuted image, called *PIM*, is added modularly with the TRPI Image, called *TIM*, pixel wise to generate the encrypted image, called *EIM*. This is shown in

$$\text{EIM}_{i,j} = \text{PIM}_{i,j} + \text{TIM}_{i,j} \mod 256, \tag{3.5}$$

where i and j are the coordinates of the pixels in the range of $[0, 255]$.

In the decryption side, to recover the 2*D* encrypted image, the same TRPI Image should be modularly subtracted from the 2*D* encrypted image in mod 256. After that, the product image will be depermuted to retrieve a plain image. The simulation of the proposed encryption system is investigated in Section 4.

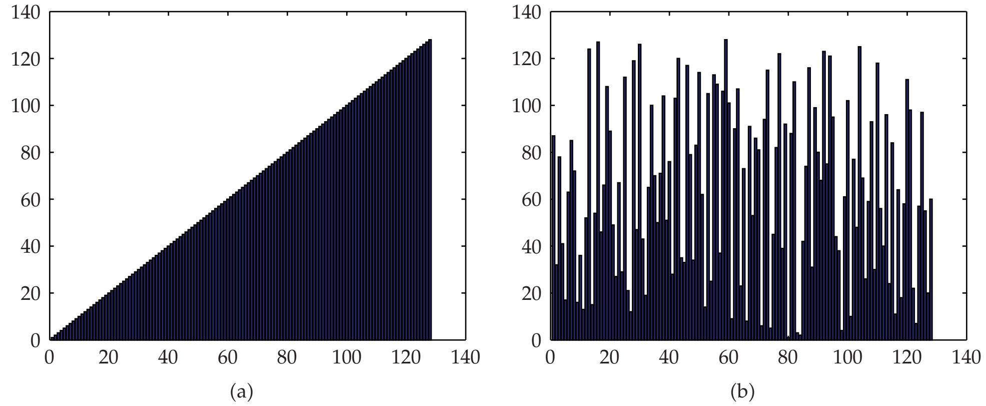


Figure 6: A sample of identity and permutation matrix.

4. Simulation of the Chaotic Image Encryption System

The proposed chaotic image encryption along with individual permutation and substitution has been simulated using MATLAB tools. In order to verify the exact operation of the proposed encryption system, and according to the process map of the system, that is, Figures 2, 3, and 5, the proposed chaotic image encryption has been coded and simulated. A 128 × 128 Lena image with 256 gray scales is used as a plain-image. The results obtained by the Lena gray scales image are demonstrated.

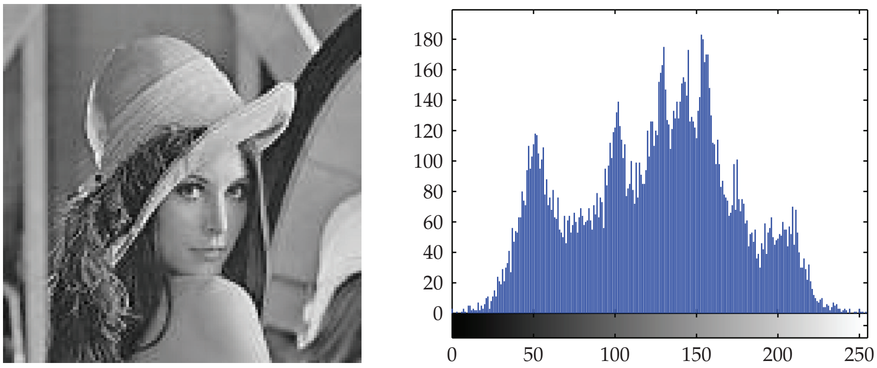
The proposed encryption system includes two major units, chaotic pixels permutation unit and chaotic pixels substitution unit. Three processes called logistic random bit sequence (LRBS) generator, Pseudorandom number calculator, and Tompkins-Paige algorithm are used to perform the pixel permutations.

First, the logistic map to generate a string of bits uses a *Key-P* with 90 *bits*. Since the chaotic range of the initial parameter is about 0.4, 26 *bits* of *Key-P* are used as an initial value. 64 *bits* are considered as a control parameter. Then 127 integer Pseudorandom numbers, degree, are calculated and used as the degree of permutation. The Tompkins-Paige algorithm is then used to find the target permutation by multiplication of 127 simple permutations. A sample of identity and permutation matrix of a sample key is presented in Figure 6.

Subsequently, as shown in Figure 5 and explained in Section 3.2, *Key-S* (128 *bits)* is entered as an input variable to the system, 64 *bits* as an initial value of the map and 64 *bits* as the control parameter. A Pseudorandom image of size 128 × 128 with 256 gray scales is generated using the tent map. The pixels of the Pseudorandom image are then modularly added with the pixels of permutated image of the previous phase to generate the final encrypted image.

Afterward, an 128 × 128 image with 256 gray scales (Figure 7) is used as a plain-image and applied to the proposed encryption system. The output of each stage is shown next. The results of row and column permutation unit are shown in Figures 8 and 9, respectively.

In Figure 10, the result of the 2D permutation of plain-image is illustrated, where the permutation matrixes of the rows and columns are identical using the identical *Key-P*'s. Figure 11 shows another 2D permutation with different permutation matrixes, using different *Key-P*'s. The encrypted images of Figures 10 and 11 approximate Pseudorandom noisy images.



(a)

(b)

Figure 7: 128 × 128 Lena image with 256 gray scales and its image histogram.

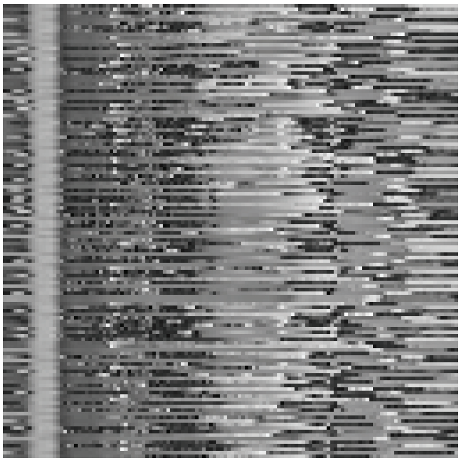


Figure 8: Row chaotic pixel permutation of Lena image.

A 128 *bits* key is then entered to the tent map, 64*bits* as an initial value of the map and 64*bits* as the control parameter. Tent map is used to generate a Pseudorandom image. In each iteration of the tent map, a Pseudorandom number between 0 to 1 is generated. The Pseudorandom number should be linearly transformed to a range of $[0, 255]$, since a 256 gray scale image is desired. Then, the permuted image is modularly added with Pseudorandom image pixel-wise in mod 256. Figure 12 illustrates an example of a 128 × 128 tent Pseudorandom image with 256 gray scales and its histogram. It is similar to a noisy image.

Finally, the 2D permuted image is modularly added with the TRPI Image pixel-wise in mod 256. The results of the modularly addition stage and its histogram are depicted in Figure 13. The final histogram clearly appears uniform.

5. Security Analysis

In this section, the performance of the proposed chaotic image encryption system is analyzed. The security analysis presented in this section is based on the performance of only one

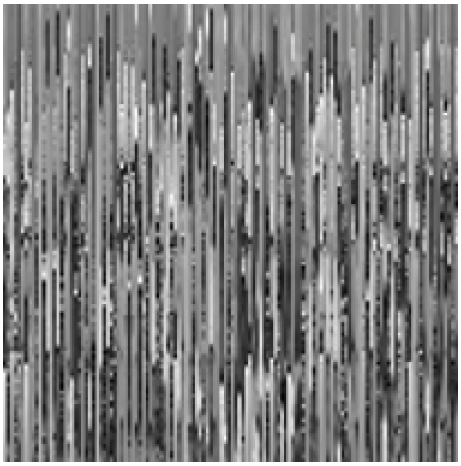


Figure 9: Column chaotic pixel permutation of Lena image.

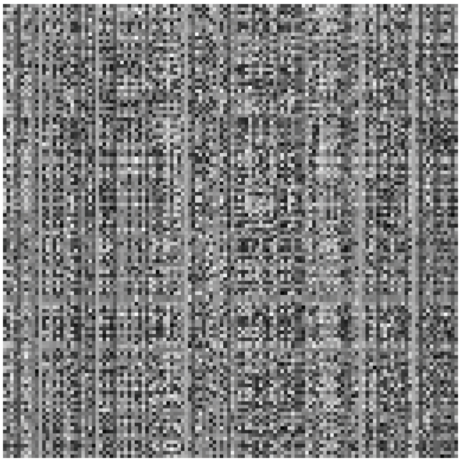


Figure 10: Row-Column chaotic 2D pixels permutation of Lena image (identical permutation matrixes for row and column).

round of operation of the proposed encryption system including a $2D$ permutation and a substitution. However, to improve the security of the proposed algorithm, more than one iteration can be applied with different keys. The first criterion for this security analysis is the chi-square test of histogram of each encrypted image. The second criterion is the correlation coefficients of pixels in the encrypted image in the vertical, horizontal, and diagonal directions. The third criterion is the difference between each encrypted and corresponding plain-image, which is measured by mean absolute difference, number of pixel change rate, and unified average changing intensity. The fourth criterion in this security analysis is key space.

5.1. Histogram

The histogram of the plain-image is illustrated in Figure 7. The histograms of all permuted images shown in Figures 8 to 11 are similar to the histogram of the plain-image. The

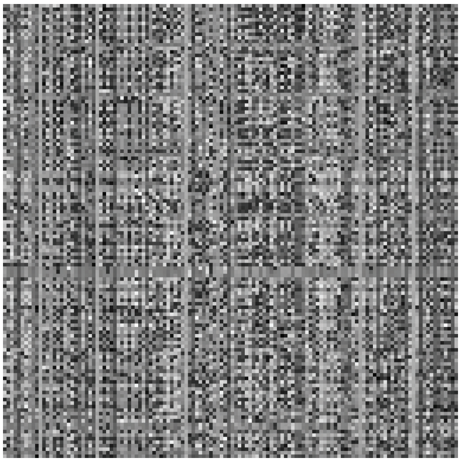
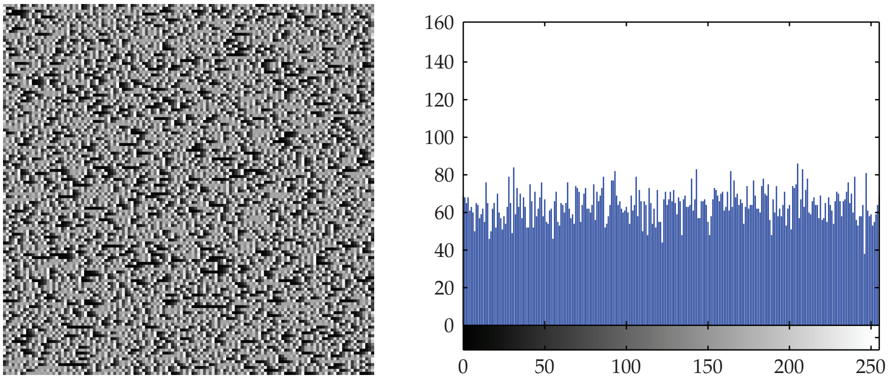


Figure 11: Row-Column chaotic 2D pixels permutation of Lena image (different permutation matrixes for row and column).



(a)

(b)

Figure 12: 128 × 128 tent Pseudorandom image with 256 gray scales and its image histogram.

histogram of the encryption system has to approximate the uniform distribution. The result of the encryption system and its histogram are illustrated in Figure 13. The histogram is approximated by a uniform distribution. The uniformity is justified by the chi-square test [30] in

$$\chi^2 = \sum_{k=1}^{256} \frac{(v_k - 64)}{64}, \tag{5.1}$$

where k is the number of gray levels (256), v_k is the observed occurrence frequencies of each gray level (0–255), and the expected occurrence frequency of each gray level is 64. Assuming a significant level of 0.05, $\chi^2(255, 0.05) = 293$. Chi-square value for the final encrypted image of the proposed system is 290, $\chi^2(\text{test}) = 290$. This implies that the null hypothesis is not rejected and the distribution of the encrypted histogram is uniform, $\chi^2(\text{test}) < \chi^2(255, 0.05)$ [30].

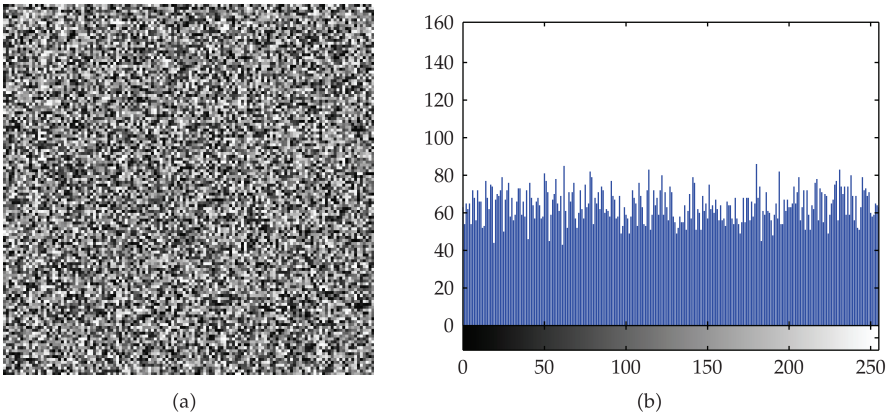


Figure 13: Proposed Chaotic Encrypted Image of Lena with 256 gray scales and its image histogram.

5.2. Correlation Coefficient

The proposed chaotic image encryption system should be resistant to statistical attacks. Correlation coefficients of pixels in the encrypted image should be as low as possible [12, 31]. Horizontal, vertical, and diagonal correlation coefficients (r_{xy}) of two adjacent pixels can be calculated using the following equations:

$$r_{xy} = \frac{\mathrm{COV}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}},$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \frac{1}{N}\sum_{i=1}^{N}x_i\right)^2, \tag{5.2}$$

$$\mathrm{COV}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)),$$

where x and y are gray-scale values of two adjacent pixels in the image and E denotes the expectation operator shown in

$$E(z) = \frac{1}{N}\sum_{i=1}^{N} z_i. \tag{5.3}$$

About a thousand pairs of two adjacent (in vertical, horizontal, and diagonal direction) pixels are randomly selected from the encrypted image, and the correlation coefficients are calculated, respectively. The results are shown in Table 1. It is clear that the correlation coefficients of the proposed encrypted image (Figure 13) in all three directions are smaller than the correlation coefficients of the proposed permuted image (Figure 7). Correlation coefficients of tent Pseudorandom image (Figure 12) are also small.

Meanwhile, the correlation coefficients of the proposed methods (Table 1: column 5, Table 2: column 6) are compared with results of four other papers [6, 9, 10, 32], which are

Table 1: Comparison of correlation coefficients of the proposed methods.

Correlation coefficient	Plain-image (Figure 7)	Proposed permuted image (Figure 10)	Proposed tent Pseudorandom image (Figure 12)	Proposed encrypted image (Figure 13)
Horizontal (H)	0.798	0.043	0.149	0.005
Vertical (V)	0.867	0.271	0.033	0.011
Diagonal (D)	0.769	0.054	0.051	0.023
$(\mathrm{H}^2 + \mathrm{V}^2 + \mathrm{D}^2)^{0.5}$	1.407	0.280	0.161	0.026
Average (H, V, D)	0.811	0.123	0.078	0.013

Table 2: Comparison of correlation coefficient of the proposed method and the other methods.

Correlation coefficient	Mao et al. [6]	Zhang et al. [9]	Gao et al. [10]	Zhou et al. [32]	Proposed encrypted image (Figure 13)
Horizontal (H)	0.045	0.082	0.016	0.012	0.005
Vertical (V)	0.028	0.040	0.065	0.027	0.011
Diagonal (D)	0.021	0.005	0.032	0.007	0.023
$(\mathrm{H}^2 + \mathrm{V}^2 + \mathrm{D}^2)^{0.5}$	0.057	0.091	0.074	0.030	0.026
Average (H, V, D)	0.031	0.042	0.038	0.015	0.013

shown in Table 2. As shown, the average correlation coefficient of the proposed system is less than all of the other methods.

5.3. Difference between Encrypted and Plain-Images

The encrypted image should be significantly different to the original one. To quantify this requirement, three measures are used: mean absolute error (MAE), the number of pixel change rate (NPCR), and unified average changing intensity (UACI) [1, 30].

The performance of each stage of the difference between permuted/encrypted and plain-images is measured by the mean absolute error (MAE) criterion in

$$\mathrm{MAE} = \frac{1}{L \times L} \sum_{j=1}^{L} \sum_{i=1}^{L} \left| a_{ij} - b_{ij} \right|, \tag{5.4}$$

where L, size of image, is equal to 128. The parameters a_{ij} and b_{ij} are gray-scale values of pixels in plain and encrypted images, respectively. The larger the MAE value, the better the encryption security. According to the selected key, the results are shown in Table 3. It is illustrated that MAE of the proposed column permutated image is about 23, while the MAE for row permuted image and $2D$ permuted image are about 27. The MAE of proposed encrypted image is about 35 that is 26 percent more than MAE of row and $2D$ permutation. It is obvious that substitution and permutation are more secure than only-permutation encryption systems.

Table 3: A comparison of MAE of different methods.

Proposed Methods	MAE
Row Permutation	27.48
Column Permutation	22.99
2*D* Permutation	27.84
Substitution and Permutation	35.13

Table 4: Comparison of NPCR and UACI criteria of proposed method and the others.

Criteria (expected value)	Mao et al. [6] first round	Zhang et al. [9] 2nd round	Gao et al. [10]	Zhou et al. [32] 2nd round	Proposed method first round
NPCR (99.61%)	37%	21.5%	NA	25.0%	99.7%
UACI (33.46%)	9%	2.5%	NA	8.5%	29.3%

The NPCR is the percentage of corresponding pixels with different gray levels in two images. Let $C_1(i, j)$ and $C_2(i, j)$ be the gray level of the pixels at the ith row and jth column of two W × H images. The NPCR of these two images is defined in

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i, j)}{W \times H} \times 100\%, \tag{5.5}$$

where $D(i, j)$ is defined as

$$D(i, j) = \begin{cases} 0, & \text{if } C_1(i, j) = C_2(i, j), \\ 1, & \text{if } C_1(i, j) \neq C_2(i, j). \end{cases} \tag{5.6}$$

Another measure, UACI, is defined as the average intensity difference in a gray level of corresponding pixels and is defined as

$$\mathrm{UACI} = \frac{1}{W \times H} \left[\sum_{i,j} \frac{C_1(i, j) - C_2(i, j)}{2^L - 1} \right]. \tag{5.7}$$

Considering two Pseudorandom images, the expected value of NPCR is found to be 99.61%. The proposed method is evaluated using this criterion and NPCR of Figure 13 is 99.7%. In the case of two Pseudorandom images, the expected value of UACI can be computed as 33.46%, assuming each gray level is coded with 8 *bits*. The proposed method is evaluated using this criterion too and UACI of image in Figure 13 is 29.3%.

As shown in Table 4 our proposed method with 0.09% difference to the expected value of NPCR is improved compared to the other reported methods. It also shows that, our method has advantage with respect to UACI criteria, with a difference of about 10% to its expected value.

Table 5: Comparison of Key length of proposed method and the others.

Key	Mao et al. [6]	Zhang et al. [9]	Gao et al. [10]	Zhou et al. [32]	Proposed Method
Length (Bin.)	2^{128}	NA	2^{150}	2^{112}	2^{218}
Length (Dec.)	10^{38}	NA	10^{45}	10^{33}	10^{65}

5.4. Key Space Analysis

Key space should be sufficiently large to make brute-force attack infeasible. Key space is the total number of different keys that can be used in the encryption system. The keys of the proposed system in this paper consist of permutation key, *Key-P* = 90 *bits*, and substitution key, *Key-S* = 128 *bits*. Each key includes initial value and control parameter of corresponding chaotic maps. Only 26 *bits* are used as parameter of logistic map, since the chaotic range of logistic map is about 40% of a chaotic range of tent map. The total key length is 218 *bits*, which contain three equal 64 *bits* plus 26 *bits*. Therefore, the key space is 2^{218}, that is, $4.12 \times 10^{+65}$. It is shown that, the key space is large enough to resist the proposed system against any brute-force attack. Comparison of the key length in our proposed method with the others is shown in Table 5.

It is possible to increase the number of bits for total key in hardware implementation. However, by increasing the key length, volume of hardware is increased and consequently speed of the system is decreased. With respect to the speed of the today's computers, the key space size should be more than $2^{100} = 10^{30}$ in order to avoid brute-force attacks [33].

6. Conclusion

In this paper, we presented a new permutation-substitution image encryption architecture using chaotic maps and Tompkins-Paige algorithm. The proposed encryption system included two major parts, chaotic pixels permutation and chaotic pixels substitution. A logistic map was used to generate a bit sequence, which was in turn used to generate Pseudorandom numbers in Tompkins-Paige algorithm, in pixel permutation phase. Pixel substitution phase, included two processes, the tent Pseudorandom image (TPRI) generator and modulo addition operation. A tent map was used to produce a Pseudorandom image that was mixed with the permuted image.

The permutation and substitution operations needed two different keys, *Key-P* and *Key-S*, respectively. The total key length was 218 *bits*. Therefore, the key space was 2^{218}, that is, $4.12 \times 10^{+65}$, which was large enough to protect the system against any brute-force attacks.

The image was a $2D$ array of pixels, each with 256 gray scales. The $2D$ permutation was designed by permutation of rows and columns simultaneously. To improve security of the proposed encryption system, the histogram needed to become uniform. This was achieved by pixel substitution. There were two main parts for pixel substitution here, tent Pseudorandom image generator and modulo addition operation. A *Key-S* was entered in to the tent map to generate a Pseudorandom image with uniform histogram. Subsequently, pixels of permuted image were modularly added to pixels of random image with uniform distortion.

All parts of the proposed chaotic encryption system were simulated using a computer code. The histogram of the encrypted image was approximated a uniform distribution. The uniformity was justified by the chi-square test. Chi-square value shows that the distribution of the histogram of the encrypted image is uniform. The vertical, horizontal, and diagonal correlation coefficients, as well as their average and RMS values for the proposed encrypted image were calculated. The individual values and their average and RMS values of correlation coefficients were lower than the corresponding values from previous research by a factor between 13% to 70%. Therefore, the proposed encryption system was resistant against any statistical attack.

To quantify the difference between encrypted image and corresponding plain-image, three measures were used: mean absolute error (MAE), number of pixel change rate (NPCR), and unified average changing intensity (UACI). It was concluded that the NPCR and UACI criteria of the proposed system were satisfactory when compared to other research results as was the security performance of the proposed system. All these results were obtained in only one round of encryption process.

Acknowledgment

The authors would like to thank Dr. Mehrnaz Shoushtarian, for her useful comments and suggestions.

References

[1] Y. V. Mitra, S. Rao, and S. R. M. Prasanna, "A new image encryption approach using combinational permutation techniques," *International Journal of Computer Science*, vol. 1, no. 2, pp. 127–131, 2006.

[2] D. Van de Ville, W. Philips, R. Van de Walle, and I. Lemahieu, "Image scrambling without bandwidth expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 892–897, 2004.

[3] M. Yang, N. Bourbakis, and S. Li, "Data-image-video encryption," *IEEE Potentials*, vol. 23, no. 3, pp. 28–34, 2004.

[4] J. Fridrich, "Image encryption based on chaotic maps," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1105–1110, 1997.

[5] G. Chen, Y. Mao, and C. K. Chui, "A symmetric image encryption scheme based on 3D chaotic cat maps," *Chaos, Solitons & Fractals*, vol. 21, no. 3, pp. 749–761, 2004.

[6] Y. Mao, G. Chen, and S. Lian, "A novel fast image encryption scheme based on 3D chaotic baker maps," *International Journal of Bifurcation and Chaos*, vol. 14, no. 10, pp. 3613–3624, 2004.

[7] Z.-H. Guan, F. Huang, and W. Guan, "Chaos-based image encryption algorithm," *Physics Letters A*, vol. 346, no. 1–3, pp. 153–157, 2005.

[8] S. Lian, J. Sun, and Z. Wang, "A block cipher based on a suitable use of the chaotic standard map," *Chaos, Solitons & Fractals*, vol. 26, no. 1, pp. 117–129, 2005.

[9] L. Zhang, X. Liao, and X. Wang, "An image encryption approach based on chaotic maps," *Chaos, Solitons & Fractals*, vol. 24, no. 3, pp. 759–765, 2005.

[10] H. Gao, Y. Zhang, S. Liang, and D. Li, "A new chaotic algorithm for image encryption," *Chaos, Solitons & Fractals*, vol. 29, no. 2, pp. 393–399, 2006.

[11] Q. Zhou, K.-W. Wong, X. Liao, T. Xiang, and Y. Hu, "Parallel image encryption algorithm based on discretized chaotic map," *Chaos, Solitons & Fractals*, vol. 38, no. 4, pp. 1081–1092, 2008.

[12] A. N. Pisarchik and M. Zanin, "Image encryption with chaotically coupled chaotic maps," *Physica D*, vol. 237, no. 20, pp. 2638–2648, 2008.

[13] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image and Vision Computing*, vol. 24, no. 9, pp. 926–934, 2006.

[14] M. S. Ehsani and S. E. Borujeni, "Fast Fourier transform speech scrambler," in *Proceedings of the 1st International IEEE Symposium on Intelligent Systems*, vol. 1, pp. 248–251, 2002.

[15] S. E. Borujeni and A. Zakerolhoseini, "Permutation based image encryption using pseudo random number generator and Tompkins-Paige algorithm," in *Proceedings of the International Conference on Robotics, Vision, Information and Signal Processing*, pp. 478–481, Penang, Malaysia, 2007.

[16] C. E. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.

[17] D. R. Stinson, *Cryptography: Theory and Practice*, CRC Press Series on Discrete Mathematics and Its Applications, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2nd edition, 2002.

[18] K. T. Alligood, T. D. Sauer, and J. A. Yorke, *Chaos: An Introduction to Dynamical Systems*, Textbooks in Mathematical Sciences, Springer, New York, NY, USA, 1997.

[19] B. Furht and D. Kirovski, *Multimedia Security Handbook*, CRC Press, Boca Raton, Fla, USA, 2005.

[20] J.-C. Yen and J.-I. Guo, "A new chaotic key-based design for image encryption and decryption," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '00)*, vol. 4, pp. 49–52, Geneva, Switzerland, May 2000.

[21] L. P. L. de Oliveira and M. Sobottka, "Cryptography with chaotic mixing," *Chaos, Solitons & Fractals*, vol. 35, no. 3, pp. 466–471, 2008.

[22] S. Lian, J. Sun, and Z. Wang, "Security analysis of a chaos-based image encryption algorithm," *Physica A*, vol. 351, no. 2–4, pp. 645–661, 2005.

[23] W. Yuanzhi, R. Guangyong, J. Julang, Z. Jian, and S. Lijuan, "Image encryption method based on chaotic map," in *Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications (ICIEA '07)*, pp. 2558–2560, 2007.

[24] S. Li, G. Chen, and X. Zheng, "Chaos-based encryption for digital images and videos," in *Multimedia Security Handbook*, pp. 133–167, CRC Press, Boca Raton, Fla, USA, 2004.

[25] S. E. Borujeni, "Speech encryption based on fast Fourier transform permutation," in *Proceedings of the 7th IEEE International Conference on Electronics, Circuits and Systems (ICECS '00)*, vol. 1, pp. 290–293, 2000.

[26] K. Sakurai, K. Koga, and T. Muratani, "A speech scrambler using the fast Fourier transform technique," *IEEE Journal on Selected Areas in Communications*, vol. 2, no. 3, pp. 434–442, 1984.

[27] G. Polya, *Applied Combinatorial Mathematics*, Krieger, 1981.

[28] H.-P. Xiao and G.-J. Zhang, "An image encryption scheme based on chaotic systems," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, pp. 2707–2711, Dalian, China, 2006.

[29] J. Zou, C. Xiong, D. Qi, and R. K. Waro, "The application of chaotic maps in image encryption," in *Proceedings of the 3rd International IEEE Northeast Workshop on Circuits and Systems Conference (NEWCAS '05)*, pp. 331–334, Quebec City, Canada, 2005.

[30] H. S. Kwok and W. K. S. Tang, "A fast image encryption system based on chaotic maps with finite precision representation," *Chaos, Solitons & Fractals*, vol. 32, no. 4, pp. 1518–1529, 2007.

[31] S. E. Borujeni and M. Eshghi, "Design and simulation of encryption system based on PRNG and Tompkins-Paige permutation algorithm using VHDL," in *Proceedings of the International Conference on Robotics, Vision, Information and Signal Processing*, pp. 63–67, Penang, Malaysia, 2007.

[32] F. Zhou, G. Cao, and B. Li, "Design of digital image encryption algorithm based on compound chaotic system," *Journal of Harbin Institute of Technology*, vol. 14, supplement 2, pp. 30–33, 2007.

[33] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.