Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 632060, 13 pages
http://dx.doi.org/10.1155/2014/632060

Research Article

Image Encryption Using the Chaotic Josephus Matrix

Gelan Yang,[1] Huixia Jin,[1] and Na Bai[2]

[1] *Department of Computer Science, Hunan City University, Yiyang, Hunan 413000, China*
[2] *School of Electronics and Information Engineering, Anhui University, Hefei, Anhui 230039, China*

Correspondence should be addressed to Na Bai; realbain@gmail.com

Received 3 October 2013; Accepted 14 January 2014; Published 6 March 2014

Academic Editor: Yue Wu

This paper presents a new image encryption solution using the chaotic Josephus matrix. It extends the conventional Josephus traversing to a matrix form and proposes a treatment to improve the randomness of this matrix by mixing chaotic maps. It also derives the corresponding encryption primitives controlled by the chaotic Josephus matrix. In this way, it builds up an image encryption system with very high sensitivities in both encryption key and input image. Our simulation results demonstrate that an encrypted image of using this method is very random-like, that is, a uniform-like pixel histogram and very low correlations in adjacent pixels. The design idea of this method is also applicable to data encryption of other types, like audio and video.

1. Introduction

Encryption is the process of transforming information (referred to as plaintext) using an algorithm (referred to as a cipher) to make the encrypted information (referred to as ciphertext) unreadable to anyone except those authorized users with special knowledge (referred to as a key) [1]. Encryption has been long used by militaries and governments to facilitate secret communications. Since the digital revolution in the 1980s, the demands of digital encryption in various applications have quickly increased because digital storage and communication are widely used. Encryption is now commonly used in protecting information within many types of civilian systems, like personal emails and patient documents.

Digital image is a major data type of two dimensions. Although a digital image can be extracted in order and becomes a one-dimensional data, its distinctive characteristics make conventional ciphers developed for one dimensional data unsuitable [2], for example, those based on Data Encryption Standard (DES) [3] and Advanced Encryption Standard (AES) [4]. As a result, digital image encryption has become an attractive research area in the past decade [5–11].

The chaotic map is considered a wise choice for data encryption because of its ergodicity, mixing property, high sensitivity to the initial conditions, high deterministic properties, high unpredictable random behaviors, and so forth [2, 5, 9, 12–15]. However, the chaotic encryption method is criticized for its vulnerability to attacks via certain basin structures [16], its low efficiency for encrypting the whole image [2], its deteriorated randomness property from its use of the finite precision with fixed-point arithmetic [17], and nonuniform distribution of the chaos sequence [18].

In computer science [19] and mathematics [20], the Josephus problem is a theoretical problem related to a certain counting-out game. If the counting-out order is recorded as a sequence, a Josephus traversing is obtained because all elements in the game are traversed without repetition. The Josephus traversing has already been used in data encryption field for years. The Josephus traversing is simple to realize and fast to compute, but previous attempts focused more or less on the scrambling purposes [21, 22]. However, scrambling based encryption is vulnerable to statistical attack, ciphertext-only attack, and known plaintext attack [23] because it never changes pixel values.

In order to achieve higher security level, many recent efforts adopt the hybrid idea to use one encryption system to suppress disadvantages of another system while keeping advantages unchanged. For example, [24] incorporates the chaotic map to DES; [25, 26] combine the chaotic map into

```
Input:  t, the initial total number of persons in a circle
        s, the starting position in the circle
        n, the counting period
Output: q_π, the Josephus permutation sequence according to parameter set (t, s, n)
count = 0; done = 0; pos = s; label = zeros(1, t); q_π = []; % initial settings
while (~done) % main loop
    todo = label(pos);
    if (todo == 0) % if this person has not been taken out
        count = count + 1;
        if (count == n) % if this is the nth person
            q_π(end + 1) = pos; count = 0; label(pos) = 1;
            if (length(q_π) == t)
                done = 1;
            end
        end
    end
    pos = pos + 1;
    if (pos > t)
        pos = 1;
    end
end
```

Algorithm 1: The generation of a Josephus permutation sequence.

conventional transform domain encryption; [27, 28] add chaotic map to Sudoku puzzles for encryption. In this paper, we develop a new image encryption method by combining the ideas of the chaotic map, the Josephus traversing sequence, and conventional substitution and transposition ciphers. The remainder of the paper is organized as follows: in Section 2, the Josephus permutation and the Logistic chaotic map are briefly reviewed; in Section 3, the CJPM and its generator are given; in Section 4, the proposed image encryption method based on CJPM is fully discussed including its flowchart and functions for each part; in Section 5, simulation results are shown and various security analyses are applied; in Section 6, the paper is concluded.

2. Preliminary

2.1. Josephus Permutation. The Josephus permutation or Josephus problem is well known in computer science and mathematics. It is named after Flavius Josephus, a Jewish historian lived in the 1st century. It is a theoretical problem related to a certain counting-out game that works by having t people standing in a circle, with consecutive tags from 1 to t. Starting at predetermined person, you count around the circle. Once you reach the nth person, take them out of the circle and have the members to close the circle. Then repeat the process, until only one person is left. That person wins the game. If we record the tags of people who have been taken out at each round as a sequence, then this sequence is a permutation of a natural number sequence and is called Josephus permutation sequence.

It is clear that three parameters are involved in the Josephus problem, namely, the initial total number of persons in a circle t, the starting position in the circle s, and the counting period n. Therefore, a Josephus permutation sequence q_π can be denoted as follows, where J denotes the Josephus permutation according to the set of parameters t, s, and n. A Josephus permutation sequence can be easily implemented by linked lists and dynamic arrays. Algorithm 1 describes a

$$q_\pi = J(t, s, n). \tag{1}$$

For example,

$$q_\pi = J(18, 1, 4) = [4, 8, 12, 16, 2, 7, 13, 18, 6, 14, 3, 11, 5, 17, 15, 1, 10, 9], \tag{2}$$

$$q_\pi = J(18, 1, 7) = [7, 14, 3, 11, 1, 10, 2, 13, 6, 18, 16, 15, 17, 5, 12, 4, 8, 9], \tag{3}$$

$$q_\pi = J(18, 4, 7) = [10, 17, 6, 14, 4, 13, 5, 16, 9, 3, 1, 18, 2, 8, 15, 7, 11, 12]. \tag{4}$$

It is clear that (1) compared with the length t natural number sequence, a Josephus permutation sequence q_π changes a lot, especially considering that none of the two neighbor numbers is consecutive; (2) for a fixed length t, different pairs of (s, n) give distinct Josephus permutation sequences. However, it is weak in that (1) q_π's very first several elements divulge the parameter of counting period n; (2) the difference between two q_πs may disclose the difference between their starting positions; for example, the difference between the two first elements of q_πs in (3) and (4) is 3, which is the difference of their parameters of starting positions. Therefore, it is not completely random for a Josephus permutation sequence.

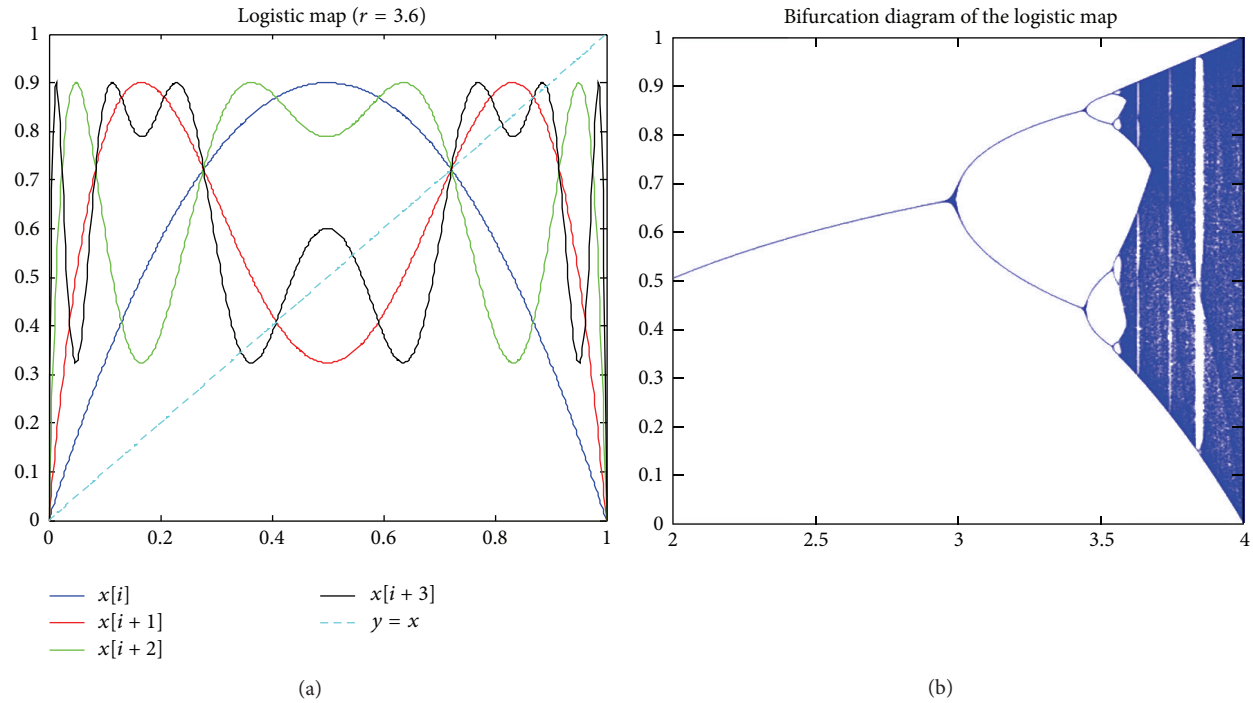


FIGURE 1: The Logistic map.

2.2. Chaotic Logistic Map. The Logistic map is a polynomial mapping of degree two. It was introduced by the biologist Robet May in 1976 [31]. The Logistic map is written as (5), where symbol $\mathscr{L}$ is used to denote the Logistic map; $X_i \in [0,1]$ and represents the population at year i and hence X_0 represents the initial population at year 0; r is a positive number and represents the combined rate for reproduction and starvation [32]. This map is often cited as an example of how complex chaotic behaviors can arise from a very simple nonlinear dynamic equation. Consider

$$X_{i+1} = \mathscr{L}(X_i) = rX_i(1 - X_i). \tag{5}$$

The Logistic map has been well studied. The plots of the first few iterations of the Logistic map and its bifurcation diagram are shown in Figure 1. It is well known that when $r \in [3.57, 4]$ (approximately), the Logistic map has chaotic behaviors for most values, but there are still certain isolated ranges of r that show nonchaotic behavior; for example, $r \approx 3.83$, which corresponds to a big gap in its bifurcation diagram.

In reality, the Logistic sequence is controlled by a set of parameters of (X_0, r, N, m), where (X_0, r) are parameters in the Logistic map, N is the length of sequence, and m denotes the number of thrown-away samples. Therefore, a Logistic sequence X can be denoted as (6). Based on this logistic sequence X of length N, sorted sequence X' can be obtained by sorting X in the ascending order. It is certain that X' is a permutation of the original X. Therefore, X and X' satisfy (7), where p_π is a permutation mapping sequence and i denotes sequence element index. Consider

$$X = \mathscr{L}(X_0, r, N, m), \tag{6}$$

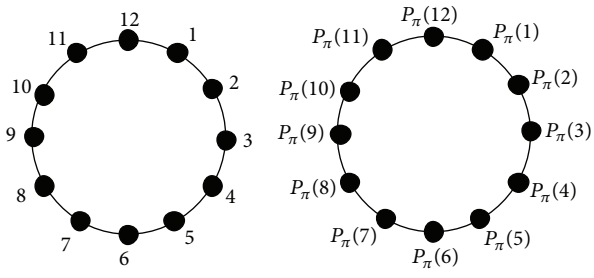
$$X'_i = X_{p_\pi(i)} \tag{7}$$

$$\begin{aligned} X = [&0.4000, 0.9120, 0.3050, 0.8055, 0.5954, 0.9154,\\ &0.2943, 0.7892, 0.6322, 0.8836, 0.3908,\\ &0.9047, 0.3277, 0.8372, 0.5179, 0.9488,\\ &0.1846, 0.5721], \end{aligned} \tag{8}$$

$$\begin{aligned} X' = [&0.1846, 0.2943, 0.3050, 0.3277, 0.3908, 0.4000,\\ &0.5179, 0.5721, 0.5954, 0.6322, 0.7892,\\ &0.8055, 0.8372, 0.8836, 0.9047, 0.9120,\\ &0.9154, 0.9488], \end{aligned} \tag{9}$$

$$\begin{aligned} p_\pi &= L(0.4, 3.8, 18, 0)\\ &= [17, 7, 3, 13, 11, 1, 15, 18, 5, 9, 8, 4, 14, 10, 12, 2, 6, 16]. \end{aligned} \tag{10}$$

For example, if $(X_0, r, N, m) = (0.4, 3.8, 18, 0)$, then X and X' are shown in (8) and (9), respectively. Correspondingly, the permutation sequence p_π is determined as (10).



(a) Conventional Josephus problem (b) Chaotic Josephus problem

Figure 2: The chaotic Josephus problem.

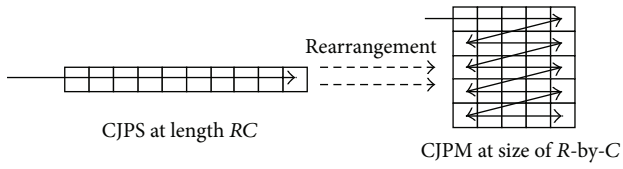


Figure 3: Rearranging a CJPS to a CJPM.

3. Chaotic Josephus Permutation Matrix (CJPM)

3.1. Chaotic Josephus Permutation Sequence (CJPS). From previous sections, it is clear that the Josephus permutation sequence q_π and the chaotic permutation sequence p_π are obtained from different mechanisms. The Josephus permutation sequence q_π is easy to obtain but not completely random-like, while the chaotic permutation sequence p_π is random-like but requires a large amount of computations for maintaining accuracy. It is desirable that a permutation sequence be random-like and only costs moderate computations.

In order to achieve the above objective, the new Josephus permutation sequence based on a chaotic permutation sequence p_π is defined, whose initial positions on the circle are not consecutive numbers like those in the conventional Josephus problem. This new problem can be restated as follows: (1) t people with numbered tags stand in a circle (these numbers together form a permutation sequence according to sorting a chaotic sequence); (2) starting at predetermined person, you count around the circle until you reach the nth person; take him out of the circle and have the members to close the circle and record his tag number; (3) repeat the process, until only one person is left. If we record the tags of people who have been taken out of the circle as a sequence, then a chaotic Josephus permutation sequence is obtained.

It is noticeable that the chaotic Josephus problem has parameters for both the chaotic Logistic map and for the conventional Josephus problem. In other words, a chaotic Josephus permutation sequence cq_π is determined by (11), where parameters t, s, n have the same meanings as in (1), X_0 and r are parameters in the Logistic map, and m determines the number of thrown-away samples in the chaotic Logistic sequence. Consider

$$cq_\pi = cJ\left(t, s, n, X_0, r, m\right). \tag{11}$$

In the conventional Josephus problem (controlled by parameters (t, s, n); see (1)), a natural number sequence 1 to t is used to denote the tags for people standing in the circle, while a permutated sequence p_π (controlled by parameters (X_0, r, N, m); see (6)) is used in the chaotic Josephus problem as the tags. In order to match the sequence lengths of p_π and q_π, $t = N$ is the condition that has to be satisfied. The conventional Josephus problem and the chaotic Josephus problem for $t = N = 12$ are illustrated in Figure 2.

Therefore, the new Josephus permutation sequence cq_π can be denoted as a composed function of a conventional Josephus permutation sequence q_π and a chaotic permutation sequence p_π:

$$cq_\pi = p_\pi \circ q_\pi = p_\pi\left(q_\pi\right). \tag{12}$$

For example, if $q_\pi = J(18, 1, 4)$ in (2) and $p_\pi = L(0.4, 3.8, 18, 0)$ in (10) are preknown, $cq_\pi = cJ(18, 1, 4, 0.4, 3.8, 0)$ is obtained as (13) shows by using (12). Similarly, (14)–(16) can be also obtained. Consider

$$\begin{aligned} cq_\pi &= cJ\,(18, 1, 4, 0.4, 3.8, 0) \\ &= [13, 18, 4, 2, 7, 15, 14, 16, 1, 10, 3, 8, 11, 6, 12, 17, 9, 5] \end{aligned} \tag{13}$$

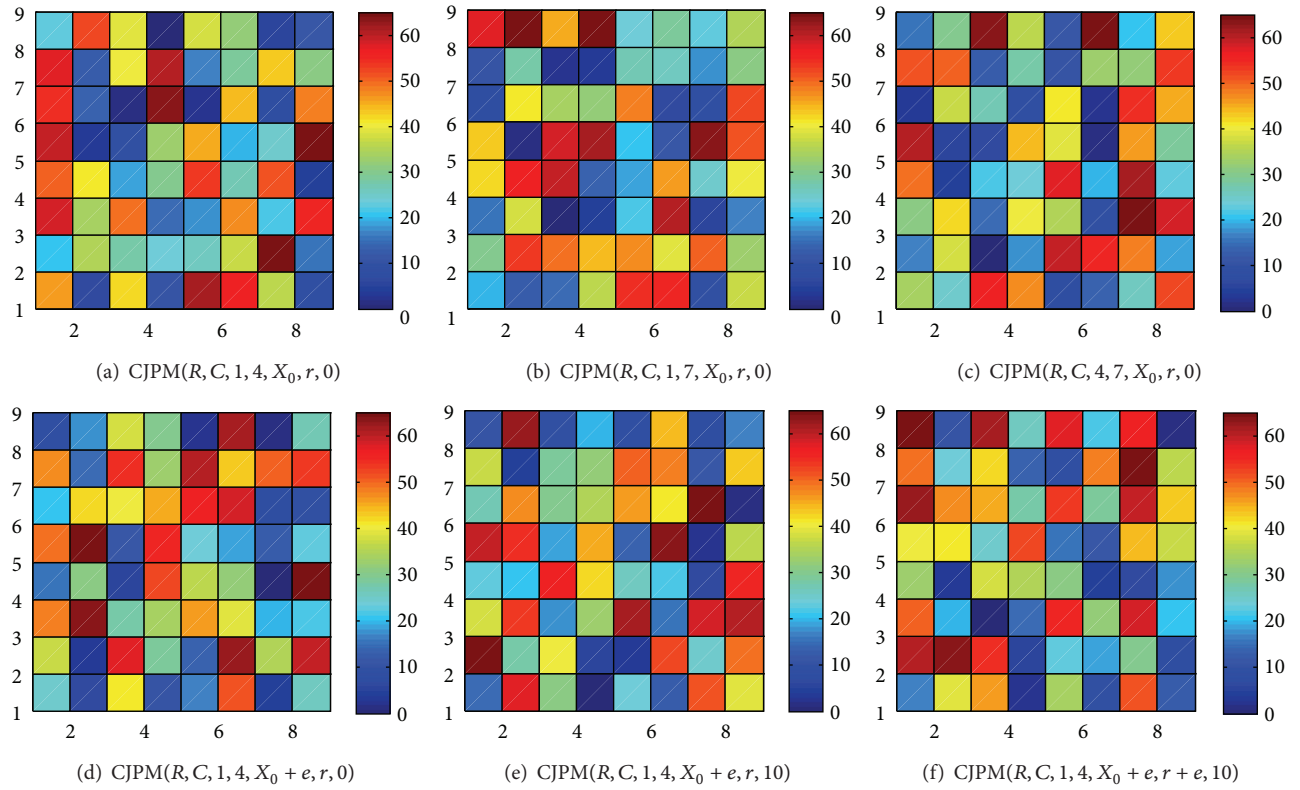
$$\begin{aligned} cq_\pi &= cJ\,(18, 1, 7, 0.4, 3.8, 0) \\ &= [15, 10, 3, 8, 17, 9, 7, 14, 1, 16, 2, 12, 6, 11, 4, 13, 18, 5] \end{aligned} \tag{14}$$

$$\begin{aligned} cq_\pi &= cJ\,(18, 4, 7, 0.4, 3.8, 0) \\ &= [9, 6, 1, 10, 13, 14, 11, 2, 5, 3, 17, 16, 7, 18, 12, 15, 8, 4] \end{aligned} \tag{15}$$

$$\begin{aligned} cq_\pi &= cJ\,(18, 4, 7, 0.40001, 3.8, 0) \\ &= [8, 6, 1, 10, 11, 12, 13, 2, 9, 15, 7, 18, 3, 5, 14, 17, 4, 16]. \end{aligned} \tag{16}$$

Compared to the previous conventional Josephus permutation sequence (see (2)–(4)), the chaotic Josephus permutation sequences (see (12)–(15)) are more random-like because (1) the difference between its very first elements is not related to its counting period anymore; (2) the difference between two cq_πs, which are only different in their starting positions, is no longer equal to the difference of their starting positions; (3) slight perturbations in chaotic map parameters lead to big changes in resulting cq_πs; (4) two neighbor elements in cq_π may or may not be consecutive. Therefore, the chaotic Josephus permutation sequence is more random-like than the conventional Josephus permutation sequence.

3.2. Chaotic Josephus Permutation Matrix (CJPM). Based on chaotic Josephus permutation sequence(s), a chaotic Josephus permutation matrix can be generated via various ways.



(a) CJPM$(R, C, 1, 4, X_0, r, 0)$ (b) CJPM$(R, C, 1, 7, X_0, r, 0)$ (c) CJPM$(R, C, 4, 7, X_0, r, 0)$

(d) CJPM$(R, C, 1, 4, X_0 + e, r, 0)$ (e) CJPM$(R, C, 1, 4, X_0 + e, r, 10)$ (f) CJPM$(R, C, 1, 4, X_0 + e, r + e, 10)$

Figure 4: Parametric CJPMs (Note: $R = 8$, $C = 8$, $X_0 = 0.4$, $r = 3.8$, and $e = 0.0001$.)

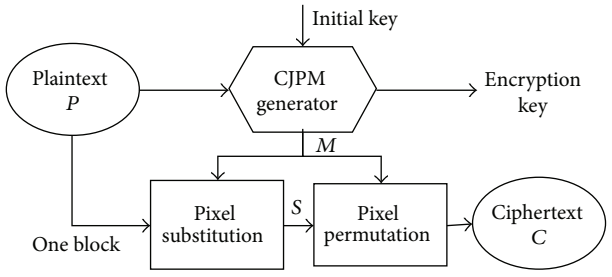


Figure 5: Image encryption method based on CJPM.

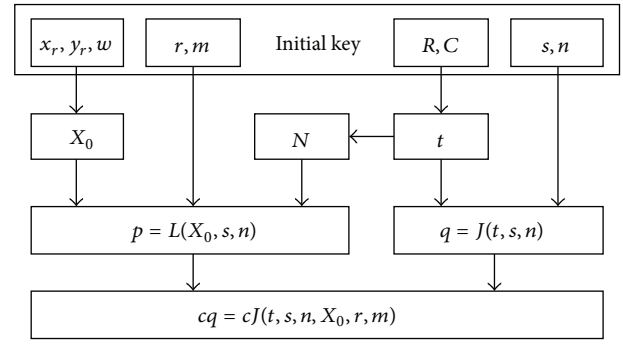


Figure 6: Key functions in CJPM.

Among these methods, Algorithm 2 illustrates a straightforward method to obtain a CJPM via a CJPS by rearranging CJPS elements to a matrix.

It is noticeable that Algorithm 2 is equivalent to rearranging a sequence of elements into a matrix following the order illustrated in Figure 3.

Therefore, a CJPM is determined by the same set of parameters controlling a CJPS. In order to emphasize the matrix property, the parameter t in CJPS is replaced by two parameters of height R and width C, where $t = RC$. Similarly, a CJPM is uniquely determined by a set of parameters (R, C, s, n, X_0, r, m) as for a CJPS. Mathematically, this claim can be denoted as

$$M = \mathrm{CJPM}\left(R, C, s, n, X_0, r, m\right). \tag{17}$$

Figure 4 illustrates various CJPMs via Algorithm 2 according to different parameter sets. It is clear that CJPM is very sensitive to its parameters and small changes in the parameter set lead to distinct CJPMs.

4. Image Encryption Algorithm Based on CJPM

In 1949, Claude Shannon, the father of "Information Theory," proposed that confusion and diffusion are two properties of the operation of a secure cipher, where the term confusion refers to making the relationship between the encryption key and the ciphertext a very complex and developed one

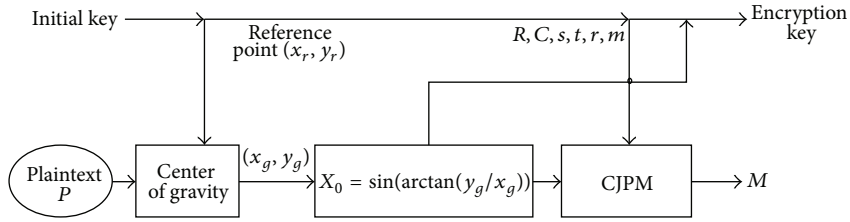


Figure 7: The internal structure of CJPM generator.

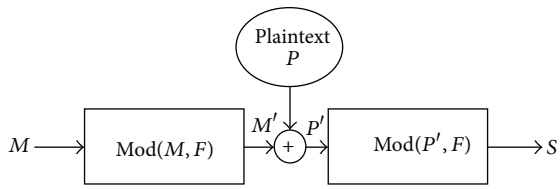


Figure 8: The internal structure of CJPM generator.

[33], and the term diffusion refers to the property that the redundancy in the statistics of the plaintext is "dissipated" in the statistics of the ciphertext [33]. In other words, for a secure cipher, it has to have good confusion and diffusion properties: (1) different ciphertexts are desired to have similar statistics; (2) any slight change in a plaintext is desired to lead to big difference in its ciphertext. The image encryption algorithm based on CJPM is proposed in this section to meet these two criteria.

4.1. Flowchart of Image Encryption Algorithm Based on CJPM. Since the CJPM is parametric and random-like, it can be used for image encryption directly. However, considering the requirements from confusion and diffusion properties, the encryption procedure can be described as Figure 5 shows. The plaintext image is first sent to the CJPM generator, which is a preparation stage for generating a CJPM M for future use. Later, this CJPM M is used as a reference matrix to permute and substitute image pixels for each image block in the stages of pixel permutation and pixel substitution, respectively. The decryption procedure is simply to reverse the encryption procedure.

4.2. Key Schedule. It is clear that the CJPM is the core of the cipher and thus key is related to the used CJPM reference matrix M. Initial key is composed of parameters $(x_r, y_r, w, r, m, R, C, s, n)$, where (x_r, y_r, w) is used in CJPM generator for obtaining plaintext-dependent parameter X_0 used in the Logistic map; (R, C) are used as the parameter t in (1) and the parameter N in (6). The functions of each part of the initial key are shown in Figure 6.

The output encryption key is composed of (R, C, s, n, X_0, r, m), all of which are directly required for determining a CJPM according to (17). Among these parameters, R, C, s, n, and m are restricted to integers; x_r, y_r, r, and X_0 are decimals. More specifically, R and C should be positive integers smaller than the plaintext image size; s and n should be positive integers below the product of RC; r should be a number in between $[3.6, 4]$; (x_r, y_r) is an arbitrary point on xy plain with weight w and m is a nonnegative integer.

4.3. CJPM Generator. In order to enhance the resistance to differential attacks, the CJPM generator used in Figure 5 is designed to be plaintext dependent. Recall that a CJPM is determined by a set of parameters (R, C, s, n, X_0, r, m) shown in (17). In the CJPM generator for image encryption, only the parameter X_0 is not directly given by the initial key but by the plaintext and a reference point (x_r, y_r) controlling the weight in calculating the center of gravity. Once X_0 is generated, it is stored in the encryption key. The whole procedure of translating the initial key to a plaintext-dependent CJPM matrix M and encryption key is shown in Figure 7.

A plaintext is considered as an object of pixels where its upper-left corner pixel is the reference point located at $(1, 1)$. Correspondingly, pixels next to it along x and y directions are $(2, 1)$ and $(1, 2)$, respectively. The center of gravity of this plaintext is calculated via (18), where P_i denotes the ith pixel intensity value and x_i and y_i denote the location of the ith pixel in the image with respect to the upper-left corner. Once the center of gravity (x_g, y_g) is obtained, the initial value of Logistic map X_0 is also determined via (19), where $\arctan(\cdot)$ is the arc tangent function and $\sin(\cdot)$ is the sine function. It is easy to verify that the range of (19) is $[0, 1]$, which satisfies restrictions for the initial value X_0 in the Logistic map. Finally, all required parameters for a CJPM, that is, (R, C, s, n, X_0, r, m), are obtained and thus a CJPM M is generated. Meanwhile, the used parameters are stored as the encryption key, which can be used in the decryption process. Consider

$$x_g = \frac{x_r \cdot w + \sum x_i P_i}{w + \sum P_i},$$
$$y_g = \frac{y_r \cdot w + \sum y_i P_i}{w + \sum P_i}, \tag{18}$$

$$X_0 = 0.5 \left[\sin\left(\arctan\left(\frac{y_g}{x_g} \right) \right) + 1 \right]. \tag{19}$$

It is worth noting that the plaintext-dependent CJPM generator guarantees that the proposed cipher has good diffusion property: any slight changes in plaintext lead to big difference in ciphertext. This is because the resulting

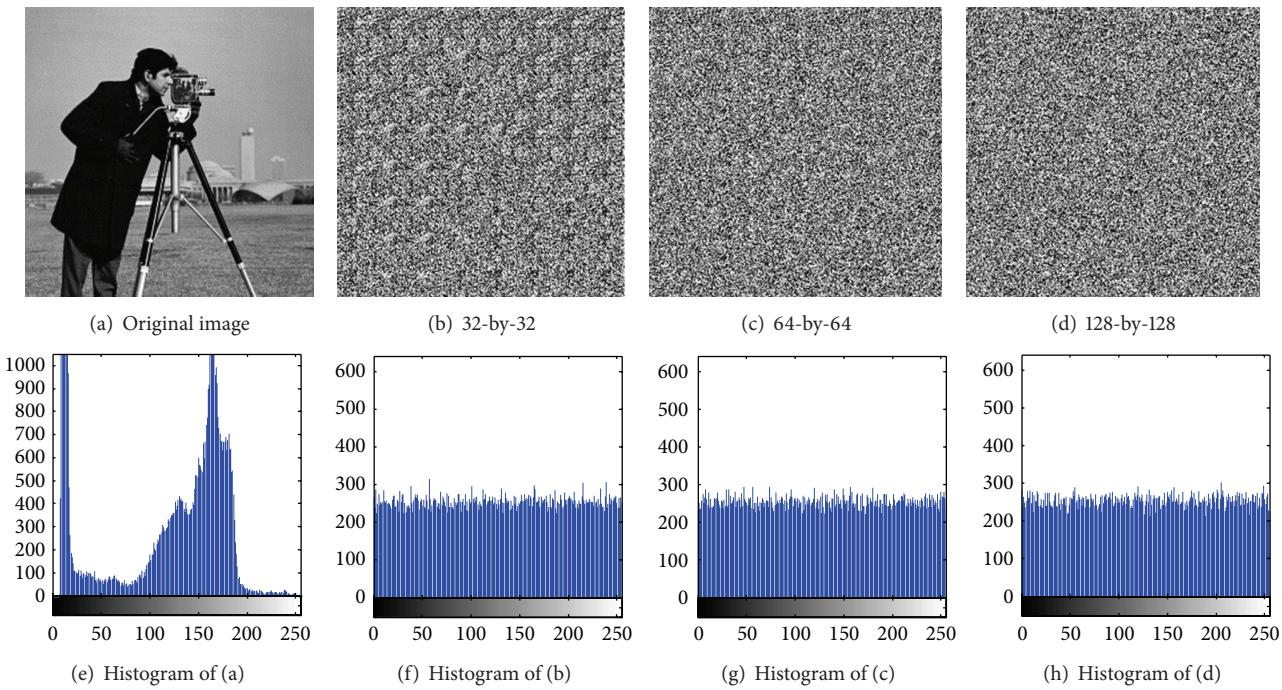


Figure 9: Pixel substitution results for CJPM at various sizes.

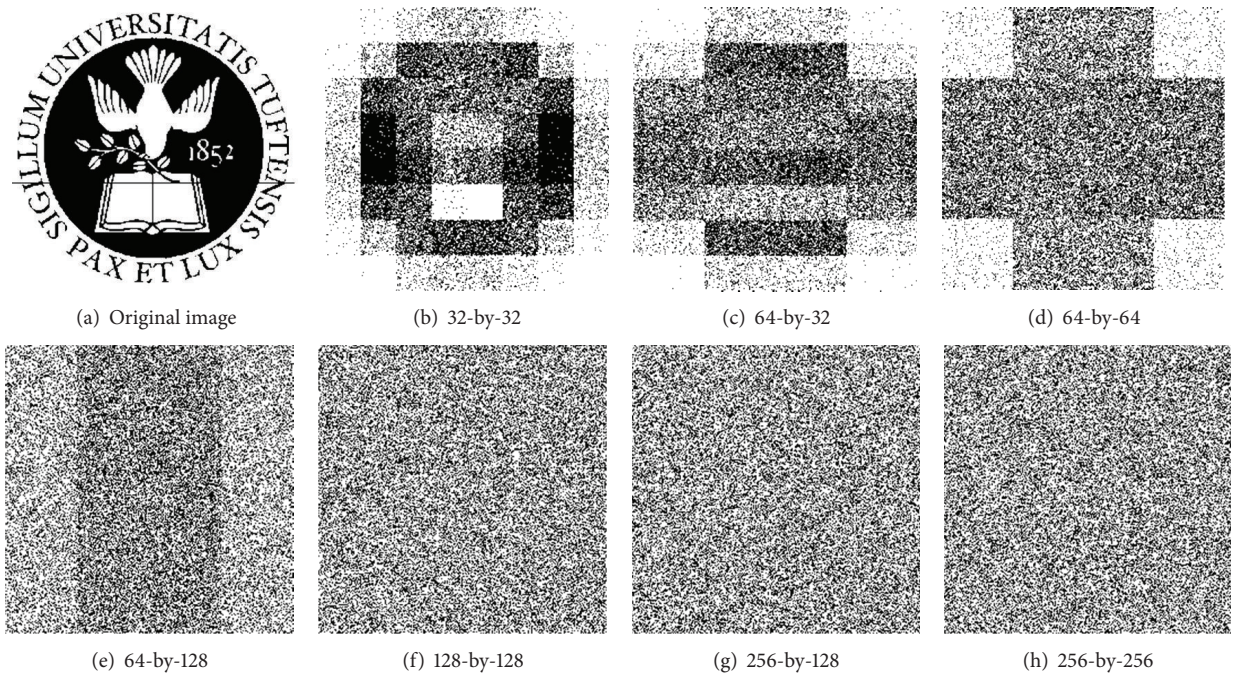


Figure 10: Pixel permutation results for CJPM at various sizes.

CJPM matrix M is dependent on the parameter X_0 and the parameter X_0 is dependent on the center of gravity for the plaintext, while the center of the plaintext gravity alters for any slight change in plaintext. Furthermore, this X_0 is the parameter in the chaotic map, and thus any slight change in initial value leads to a completely different trajectory as the bifurcation diagram in Figure 1 shows. Consequently, a completely different CJPM is obtained as the reference matrix. Eventually, this new reference matrix leads to a distinct ciphertext. It can be demonstrated that without

```
Input: cq_π, a CJPS at length t = R × C
Output: M, a CJPM of size R-by-C
for i = 1: R
        for j = 1: C
                M(i, j) = cq_π((i − 1) R + j);
        end
end
```

Algorithm 2: Reshape a CJPS into a CJPM.

preknowing the reference point's coordinate (x_r, y_r) and its weight w, there is no way to slightly change plaintext image pixels so that its gravity center is unchanged.

4.4. Pixel Substitution. Pixel substitution refers to the process of changing pixel values. From the point of view of statistics, this process is to change the statistics of a plaintext image, so that the statistics of resulting ciphertext image is completely different. Moreover, it is desired that different ciphertext images have similar statistics, which implies that ciphertext images tell little information about keys and plaintext images. As a result, the confusion property is achieved.

The proposed pixel substitution block is shown in Figure 8, where symbol $F = 2^b$ and b denotes the number of bits supported by the format of the plaintext image. For example, if plaintext P is an 8-bit gray image, then $F = 2^8 = 256$; if plaintext P is a binary image, then $F = 2^1 = 2$. It can be noticed that the reference CJPM M is first to convert to M', whose format is compatible with the format of the plaintext image; later M' and P are added over the space of F; finally the encrypted image S is obtained.

Because elements in a CJPM matrix M are uniformly distributed on integer set $[1, t]$, after "Mod" operation, M' still has a uniform-like distribution for its elements on $[0, F]$. As a result, when this M' is used to randomly shift the pixel value in plaintext, the resulting pixel value in ciphertext has an equal opportunity to be any value on $[0, F]$. As a result, a uniform-like histogram is achieved in the ciphertext.

Pixel substitution results based on CJPMs are shown in Figure 9. It is clear that histograms before and after pixel substitution are very different and ciphertext histograms are very flat compared to plaintext ones. It is also noticeable that as the size of the reference CJPM increases the ciphertext has a better encryption quality from the point view of human visual inspection.

4.5. Pixel Permutation. Pixel permutation refers to the process scrambling the positions of pixels in plaintext to disguise information contained in an image. Denote an image before and after pixel permutation as B and A, respectively. Assume the way of indexing image pixels is the same as the order to rearrange elements in a CJPM as Figure 3 shows. Then the pixel permutation process can be mathematically defined as a permutation f_π between domain B and range A: $\forall i, j \in \{1, 2, \ldots, t\}$, $\exists A_i = B_j = B_{f_\pi(i)}$, where t is the total number of pixels in the image. As we mentioned in previous sections, a CJPM is generated from a CJPS, which is a permutation sequence. Therefore, a CJPM can be directly used for pixel permutation that is given a CJPM M, and its pixel permutation can be defined as $f_\pi(i) = M(i)$.

For example, Figure 10 illustrates pixel permutation results of the "Tufts" logo image for different CJPMs. It is clear that pixels are well shuffled within the image block. As long as the size of CJPM/processing image block increases, the resulting shuffled image looks better and better. When the block size reaches to or over 128-by-128, pixels in the plaintext are almost evenly shuffled.

It is clear that images after pixel permutation look very different from the plaintext image. It is also worth noting that a CJPM also depends on a set of parameters besides the size and that any change in other parameters will lead to a completely different permutated image.

5. Simulation Results and Security Analysis

An excellent encryption method should be both robust and effective. Robustness means that the cipher should be applicable to any plaintext image written in a supported format. Effectiveness implies that the cipher is able to generate eligible ciphertext images, which hide information from possible intruders.

In this section, we focus on discussing the performance of the CJPM based image cipher described in Section 4. It is worth noting that all following computer simulations are run under MATLAB 2010a and Windows XP environment with Core 2 Quad 2.6 GHz processors.

5.1. Histogram Analysis. Histogram analysis is one of the most straightforward evaluations for ciphertext quality for it directly analyzes the pixel distribution of a ciphertext image.

Figure 11 illustrates image encryption results for various plaintext images: image "113" is a binary handwriting scanned image selected from ICDAR 2009 database; image "Lena" is a commonly used image of gray type; image "5.1.13" and "testpat.1k" are used to mimic the possible complex patterns in plaintext and they are both selected from USC-SIPI database. It is worth noting that the used CJPM is at size of 256-by-256.

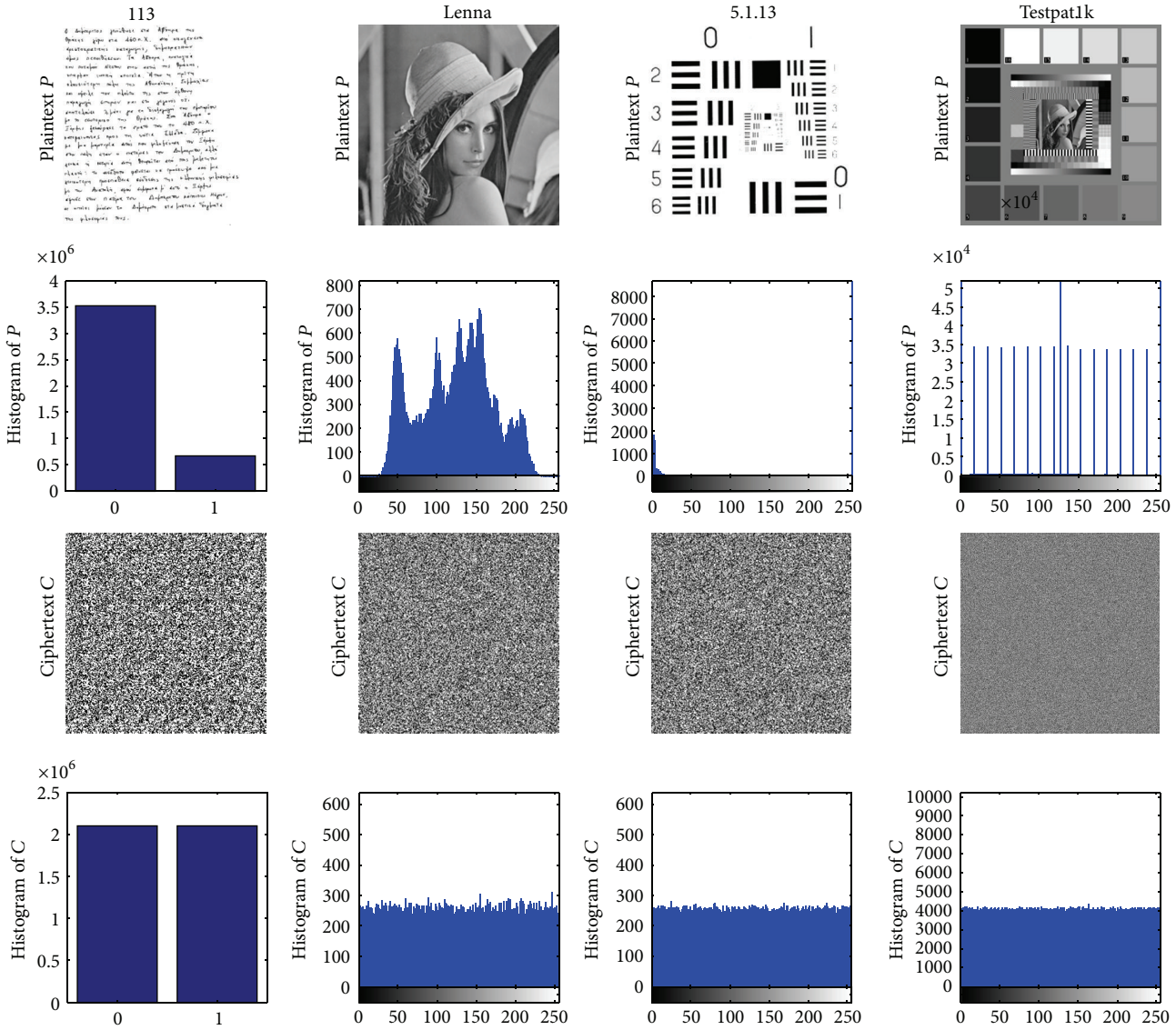


Figure 11: Histogram analysis for image encryption based on CJPM.

It is clear that no matter what histogram a plaintext image has, the histogram of its ciphertext image is flat, which implies that the pixel distribution is almost uniform. Complex patterns and large homogenous regions in plaintext images are completely unintelligible and become random-like in ciphertext images. These results imply that the proposed image encryption method based on CJPM is robust and effective for various image formats and contents.

5.2. Adjacent Pixel Autocorrelation (APAC) Analysis. High correlations of adjacent pixels can be utilized to carry out cryptanalysis. Therefore, a secure encryption algorithm should break the high correlation relationship between adjacent pixels.

In statistics, the autocorrelation R_a of a random process X describes the correlation between values of the process at different points in time, as a function of the two times or of the time difference. The autocorrelation coefficient R_a is defined in (20), where d is the time difference, μ is the mean value defined by (21), and σ is the standard deviation defined by (22); the definition of mathematical expectation is given in (23):

$$R_a(m) = \frac{E\left[\left(X_t - \mu\right)\left(X_{t+d} - \mu\right)\right]}{\sigma^2}, \tag{20}$$

$$\mu = E[X], \tag{21}$$

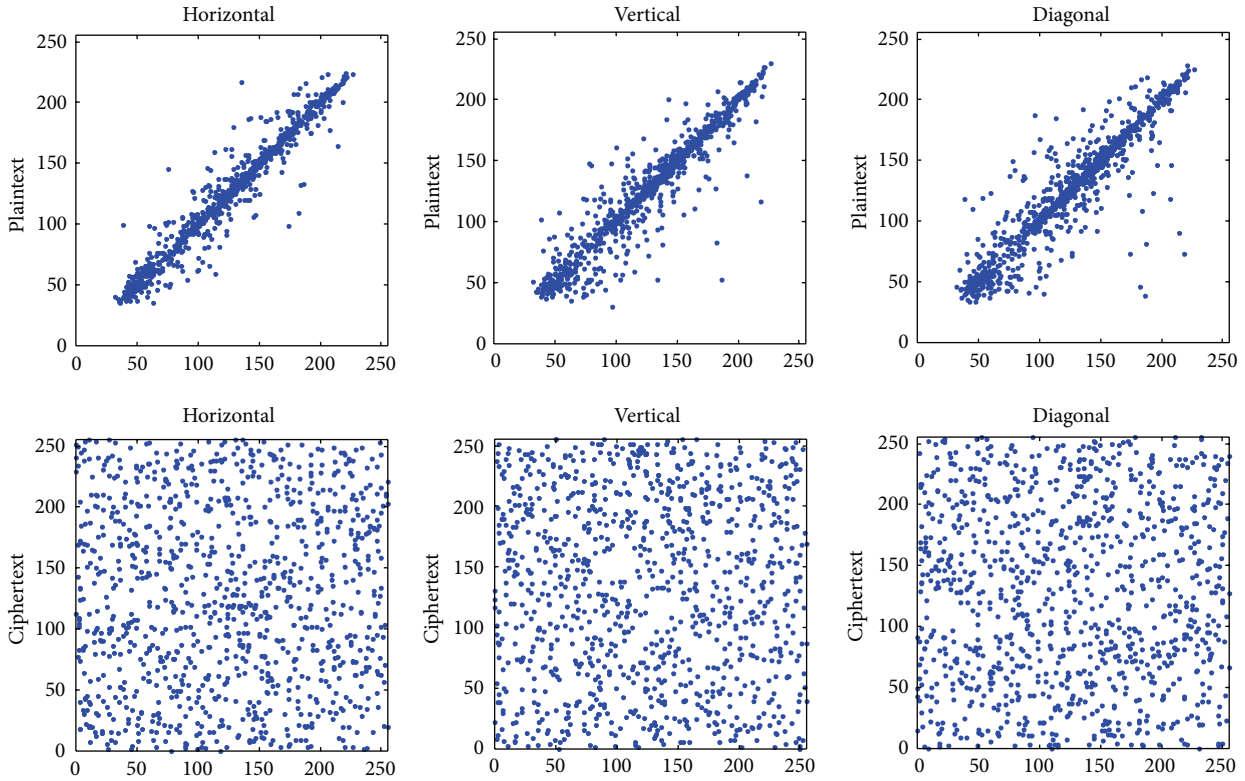


Figure 12: APAC analysis for image encryption based on CJPM.

Table 1: Adjacent pixel autocorrelation analysis.

Correlation coefficients (10^{-3})		Horizontal	Vertical	Diagonal
Plaintext	Lena	939.9652	970.9000	970.9894
Ciphertext	CJPM	−2.2281	**−0.3709**	**3.265**
	[6]	**2.0970**	16.1870	17.805
	[8]	−2.5000	−1.0000	−9.3000
	[5]	5.7765	28.434	20.662
	[29]	−12.7212	−60.2579	62.4427
	[9]	−13.4000	1.2000	39.8000
	[30]	−15.8900	−65.3800	−32.3100
	[16]	81.5860	−40.0530	−4.7150
	[10]	125.7000	58.1000	50.4000

$$\sigma = \sqrt{E\left[(X-\mu)^2\right]}, \tag{22}$$

$$E[x] = \sum_{i=1}^{N} \frac{x_i}{N}. \tag{23}$$

The closer to zero this coefficient is, the weaker the relationship two different time functions have. Specifically, in adjacent pixel correlation test, we let X be the image pixel sequence and let d be 1; that is, compare to the adjacent pixel sequence.

Based on the reference direction, there are three ways of extracting a two-dimensional image to a one-dimensional sequence and they are the horizontal adjacent correlation coefficient, the vertical adjacent correlation coefficient, and the diagonal adjacent correlation coefficient.

"Lena" image in the 2nd column of Figure 11 is used as the test plaintext image because its APAC is widely reported by other encryption methods. Peer comparison results of the proposed CJPM cipher and cited encryption methods on APAC are shown in Table 1 (best results are bolded).

In addition, Figure 12 shows the result of randomly selected 1024 pairs of two adjacent pixels from the plaintext and the ciphertext along horizontal, vertical, and diagonal directions, where x- and y-axes denote the intensity values of a randomly selected pixel and its adjacent pixel, respectively. It is clear that after applying the CJPM based image cipher, the high correlations between adjacent pixels in plaintext are broken.

5.3. Plaintext Sensitivity Analysis. In order to test the resistance of the cipher to differential attacks, plaintext sensitivity analysis is required for a secure cipher. In differential attacks, an adversary attempts to extract meaningful relationship between a plaintext image and its ciphertext image by making a slight change, usually only one pixel, in the plaintext image while encrypting the plaintext image with the same encryption key. By comparing the change in ciphertext images, the encryption key might be cracked and furthermore the information contained in ciphertext might be leaked.

Although there are other measures [34, 35] to evaluate the resistance of plaintext attacks, two classic measures are the Number of Pixel Change Rate (NPCR) and Unified Average

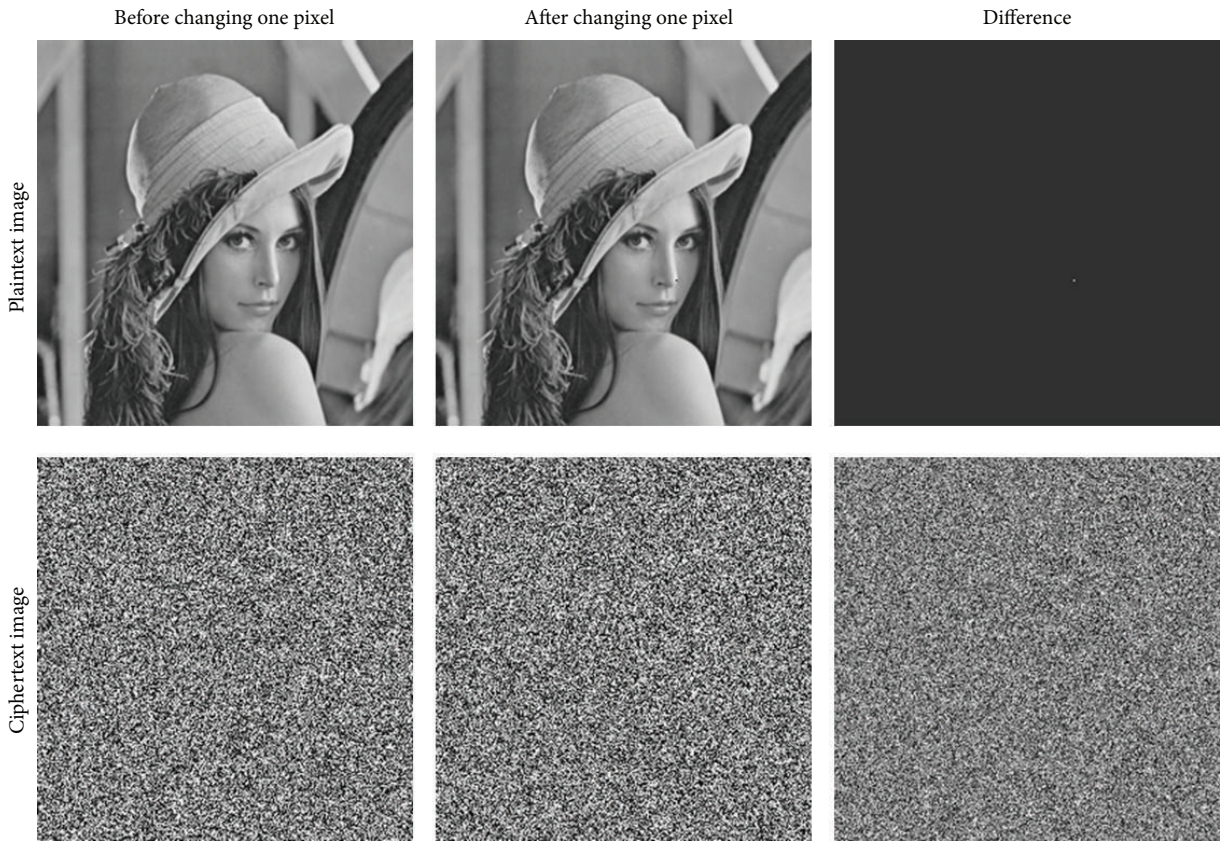


Figure 13: Plaintext sensitivity analysis (differential attacks) for image encryption based on CJPM.

(a) Encrypted image (1, 1, 128, 128, 1, 1, 4, 0)

(b) Decrypted image (1, 1, 128, 128, 1, 1, 4, 0)

(c) Decrypted image (1,1, 128, 128, 1, 2, 4, 0)

(d) Decrypted image (1, 1, 128, 128, 1, 1, 3.9999, 0)

Figure 14: Key sensitivity analysis.

Changing Intensity (UACI) [5]. The NPCR is used to measure the percentage of the number of pixels changed in ciphertext after making a slight change in plaintext. Therefore, the theoretical greatest upper-bound of the NPCR is 100%. The UACI is used to measure the averaged intensity change for pixels in ciphertext images after making a slight change in a plaintext image. It is demonstrable that the UACI of an ideal cipher for 8-bit gray images is about 0.3346 [36].

Suppose ciphertext images before and after one pixel change in a plaintext image are C^1 and C^2, respectively; the pixel values at grid (i, j) in C^1 and C^2 are denoted as $C^1(i, j)$ and $C^2(i, j)$; a bipolar array D is defined as (24), then the NPCR and UACI can be mathematically defined as (25) and (26), respectively, where symbol N denotes the total number of pixels in the ciphertext, symbol F denotes the largest supported pixel value compatible with the ciphertext image format, and $|\cdot|$ is the absolute value function. Consider

$$D(i,j) = \begin{cases} 1, & \text{if } C^1(i,j) = C^2(i,j) \\ 0, & \text{if } C^1(i,j) \neq C^2(i,j) \end{cases} \tag{24}$$

Table 2: NPCR and UACI analyses on "Lena" image.

CPJM size	16-by-16	32-by-32	64-by-64	128-by-128	256-by-256
NPCR %	99.6170	99.6246	99.5895	99.5850	99.6338
UACI %	33.8205	33.8379	33.4048	33.4076	33.4040

$$\text{NPCR} = \sum_{i,j} \frac{D(i,j)}{N} \times 100\% \tag{25}$$

$$\text{UACI} = \sum_{i,j} \frac{\left|C^1(i,j) - C^2(i,j)\right|}{(F \cdot N)} \times 100\%. \tag{26}$$

Table 2 shows the NPCR and UACI results for one pixel change in the "Lena" image. It is clear that, for CJPMs at various sizes, the proposed cipher has good performances in both the NPCR and UACI analyses. Simulation results fit the expectations of the ideal cipher very well.

Figure 13 shows a differential attack on "Lena" image while keeping the encryption key unchanged. It is noticeable that the difference between two plaintext images is the pixel on Lena's nose. However, the ciphertext images are so different that the image of their difference is still random-like. As a result, the one pixel change in the plaintext image is "dissipated" in ciphertext. From this point view, this plaintext sensitivity is closely related to the diffusion property of a cipher. In other words, it is reasonable to claim that a cipher has good NPCR and UACI results if it has good diffusion properties.

5.4. Key Space Analysis. The encryption key in the proposed image encryption method using CPJM is composed of a set of parameters $(x_r, y_r, w, R, C, s, n, r, m)$, where (x_r, y_r) is an arbitrary point on xy plain; w is a nonnegative decimal; R and C should be positive integers smaller than the plaintext image size; s and n should be positive integers below the product of RC; r should be a number in between $[3.6, 4]$; and m is a nonnegative integer. Therefore, theoretically, the key space of the proposed cipher is infinitely large.

Because the chaotic Logistic map is used as the trigger for pseudorandom sequences, the proposed cipher has high key sensitivities as well. The results of key sensitivity analysis are shown in Figure 14, where the set of parameters written in parenthesis is the used key. It is clear that unless the correct decryption key is applied, a ciphertext image cannot be restored.

6. Conclusion

In this paper, we discussed the generation of a chaotic Josephus permutation matrix by using the conventional Josephus permutation sequences and the logistic chaotic map. The proposed CJPM is parametric and is uniquely dependent on the set of parameters, which is sufficiently large to provide a secure size of key space. As another heritage from the chaotic Logistic map, the CJPM is highly sensitive to its initial values (parameters). Any slight change in parameters leads to significant differences in resulting CJPM. Simulation results show that (1) the ciphertext image is random-like from the perspective of human visual inspection; (2) the encryption quality is almost independent of the plaintext image; (3) the proposed encryption method is able to encrypt plaintext images with large homogeneous regions to secure ciphertext images; (4) histogram analysis also shows that different ciphertext images tend to have the uniform distribution on $[0, F]$; (5) adjacent pixel correlation analysis shows that neighbor pixels in ciphertext have lower correlations than many existing encryption methods; (6) the proposed cipher is highly sensitive to encryption key and plaintext; (7) experimental UACI results of the proposed cipher are very close to those of the ideal one.

The proposed cipher can be used for various image types, for example, binary images, 8-bit gray images, 16-bit gray images, RBG images, and so forth. The same encryption idea may also be applied to audio, video, or other types of digital formats.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the Scientific Research Fund of Hunan Provincial Education Department under Grant (no. 12B023). It is also supported by the National Natural Science Foundation of China (nos. 61204039 and 61106029) and Scientific Research Foundation for Returned Scholars, Ministry of Human Resources, and Social Security of the People's Republic of China.

References

[1] D. Stinson, *Cryptography: Theory and Practice*, CRC Press, 2006.

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

[3] FIPS PUB 46: Data Encryption Standard, National Bureau of Standards, 1977.

[4] FIPS PUB 197: Avanced Encryption Standard, Announcing the Advanced Encryption Standard (AES), 2001.

[5] Y. Mao and G. Chen, *Chaos-Based Image Encryption*, Springer, Berlin, Germany, 2005.

[6] H. Yang, K.-W. Wong, X. Liao, W. Zhang, and P. Wei, "A fast image encryption and authentication scheme based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 11, pp. 3507–3517, 2010.

[7] J. Hu and F. Han, "A pixel-based scrambling scheme for digital medical images protection," *Journal of Network and Computer Applications*, vol. 32, no. 4, pp. 788–794, 2009.

[8] Z. Shuo, C. Ruhua, J. Yingchun, and G. Shiping, "An image encryption algorithm based on multiple chaos and wavelet transform," in *Proceedings of the 2nd International Congress Image and Signal Processing (CISP '09)*, pp. 1–5, 2009.

[9] G. Ye, "Image scrambling encryption algorithm of pixel bit based on chaos map," *Pattern Recognition Letters*, vol. 31, no. 5, pp. 347–354, 2010.

[10] C. K. Huang and H. H. Nien, "Multi chaotic systems based pixel shuffle for image encryption," *Optics Communications*, vol. 282, no. 11, pp. 2123–2127, 2009.

[11] X. Liao, S. Lai, and Q. Zhou, "A novel image encryption algorithm based on self-adaptive wave transmission," *Signal Processing*, vol. 90, no. 9, pp. 2714–2722, 2010.

[12] S. Fu-Yan, L. Shu-Tang, and L. Zong-Wang, "Image encryption using high-dimension chaotic system," *Chinese Physics*, vol. 16, no. 12, pp. 3616–3623, 2007.

[13] V. Patidar, N. K. Pareek, and K. K. Sud, "A new substitution-diffusion based image cipher using chaotic standard and logistic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 7, pp. 3056–3075, 2009.

[14] Y. Wu, J. P. Noonan, G. Yang, and H. Jin, "Image encryption using the two-dimensional logistic chaotic map," *Journal of Electronic Imaging*, vol. 21, no. 1, Article ID 013014, 2012.

[15] Y. Wu, J. P. Noonan, and S. Agaian, "A wheel-switch chaotic system for image encryption," in *Proceedings of the International Conference on System Science and Engineering (ICSSE '11)*, pp. 23–27, June 2011.

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

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

[18] J. Jian, Y. Shi, C. Hu, Q. Ma, and J. Li, "Encryption of digital image based on chaos system," in *Computer and Computing Technologies in Agriculture II*, vol. 2, pp. 1145–1151, 2009.

[19] P. Van-Roy and S. Haridi, *Concepts, Techniques, and Models of Computer Programming*, MIT Press, 2004.

[20] P. Schumer, *Mathematical Journeys*, Wiley-Interscience, 2004.

[21] G. Ye, X. Huang, and C. Zhu, "Image encryption algorithm of double scrambling based on ASCII code of matrix element," in *Proceedings of the International Conference on Computational Intelligence and Security (CIS '07)*, pp. 843–847, December 2007.

[22] X. Desheng and X. Yueshan, "Digital image scrambling based on josephus traversing," *Computer Engineering and Applications*, no. 10, pp. 44–46, 2005.

[23] B. Furht and D. Kirovski, *Multimedia Security Handbook*, CRC Press, 2005.

[24] Y.-P. Zhang, Z.-J. Zhai, W. Liu, X. Nie, S.-P. Cao, and W.-D. Dai, "Digital image encryption algorithm based on chaos and improved DES," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '09)*, pp. 474–479, October 2009.

[25] J. Lang, R. Tao, and Y. Wang, "Image encryption based on the multiple-parameter discrete fractional Fourier transform and chaos function," *Optics Communications*, vol. 283, no. 10, pp. 2092–2096, 2010.

[26] S. Yang and S. Sun, "Video encryption method based on chaotic maps in DCT domain," *Progress in Natural Science*, vol. 18, no. 10, pp. 1299–1304, 2008.

[27] V. Wu, J. P. Noonan, and S. Agaian, "Binary data encryption using the Sudoku block cipher," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '10)*, pp. 3915–3921, October 2010.

[28] Y. Wu, Y. Zhou, J. P. Noonan, K. Panetta, and S. Agaian, "Image encryption using the Sudoku matrix," in *Proceedings of the Mobile Multimedia/Image Processing, Security, and Applications*, April 2010.

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

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

[31] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.

[32] J. Thompson and H. Stewart, *Nonlinear Dynamics and Chaos*, John Wiley & Sons, 2002.

[33] C. E. Shannon, "Communication Theory of Secrecy Systems,," *Bell Systems Technical Journal*, vol. 28, pp. 656–715, 1949.

[34] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J. P. Noonan, and P. Natarajan, "Local Shannon entropy measure with statistical tests for image randomness," *Information Sciences*, vol. 222, no. 323, 342 pages, 2013.

[35] Y. Wu, S. Agaian, and J. P. Noona, "A novel method of testing image randomness with applications to image shuffling and encryption," in *Defense, Security, and Sensing*, Proceedings of the SPIE, pp. 875507–875507, 2013.

[36] Y. Wu, J. P. Noonan, and S. Agaian, "NPCR and UACI randomness tests for image encryption," *Cyber Journals*, pp. 31–38, 2011.