Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 8124912, 11 pages
https://doi.org/10.1155/2017/8124912

Research Article

Cryptanalysis and Improvement of Chaos-Based Image Encryption Scheme with Circular Inter-Intra-Pixels Bit-Level Permutation

Haiju Fan[1,2] and Ming Li[2,3]

[1] *China National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002, China*
[2] *College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China*
[3] *School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Ming Li; liming@htu.edu.cn

Received 7 March 2017; Revised 25 April 2017; Accepted 24 May 2017; Published 5 July 2017

Academic Editor: Jonathan N. Blakely

A novel chaos-based image encryption scheme has been proposed recently. In this scheme, redundancies of the Fridrich's structure were reduced significantly via a new circular inter-intra-pixels bit-level permutation strategy. However, we proved that the original encryption scheme is vulnerable to the known/chosen-plaintext attacks. Both the permutation and diffusion phases have been improved to enhance the security of the original scheme. By shifting each row of the plain image randomly, known-plaintext attacks could be resisted. Furthermore, by appending double crossover diffusion to the end of the original scheme, chosen-plaintext attacks lost their efficacies. Simulation results demonstrated that the improved encryption scheme outperforms the original one.

1. Introduction

With the development of Internet technology, image data has been widely applied in many fields, such as military systems, government agencies, medical imaging systems, and online trade and business. However, digital images are also vulnerable to security risks; that is, sensitive information can be destroyed in the case of unauthorized access, such as interception, tampering, illegal copy, and dissemination. Encryption is a widely employed tool to minimize the potential security risks. However, in most cases, traditional encryption techniques, such as DES, RSA, AES, and IDEA, are specifically made for text data. For image data with different intrinsic characteristics, such as large data storage requirements, high redundancy, and strong correlation among adjacent pixels [1, 2], traditional encryption techniques demonstrate limited performance at best. Therefore, chaos-based image encryption schemes have been developed specifically for images. Benefiting from the inherent properties of chaotic sequences, such as ergodicity, pseudo-randomness, and sensitivity to initial conditions and control parameters [3], these chaos-based image encryption schemes have exhibited excellent properties, including complexity, security, and computational efficiency.

To date, a variety of chaos-based schemes have been proposed. However, most [4–6] are based on Fridrich's permutation-substitution model [7] and demonstrate statistical correlations among the plain image, the cipher image, and the key. Previous studies [8, 9] proved that 96.7% of the bit values were unchanged using Fridrich's encryption structure. To reduce redundancy and statistical links, many bit-level-based techniques have been proposed [10–16]. However, these techniques also have some limitations. For example, the permutation phase has repetitive patterns [14] and the operations require significant computation time [15, 16]. In 2016, Diaconu proposed a circular inter-intra-pixels bit-level permutation-confusion strategy [17]. This strategy takes advantage of Fridrich's structure; however, employing the confusion strategy reduces the redundancy significantly. Bit-level circular shifting of each row demonstrates three strong characteristics, that is, no repetitive patterns, uniform bits distribution, and reduced correlation between adjacent higher bit-planes.

To improve the security of existing cryptosystems, many cryptanalytic methods have been proposed [18, 19]. Such processes can render the existing cryptographic mechanisms insecure. In this study, we assessed Diaconu's strategy [17] and found a dangerous vulnerability in the circular shift procedure. In addition, the secret matrix, that is, the equivalent of the keys, can be revealed through known-plaintext attacks. To overcome these security problems, we propose two improvements/changes to the scheme; that is, we improve the permutation phase to resist known-plaintext attacks and improve the diffusion phase to resist chosen-plaintext and differential attacks [20], respectively.

The remainder of this paper is organized as follows. Section 2 presents an overview of Diaconu's encryption scheme [17]. Section 3 discusses cryptanalysis using a known-plaintext attack. Improvements to the original algorithm are described in Section 4. Simulation results are provided and they are evaluated in Section 5, and the conclusions are presented in Section 6.

2. Original Encryption Scheme

In this section, we describe the original scheme, which is composed of permutation and diffusion phases. The permutation phase scrambles pixels by bit-level circular shift, which updates the pixel positions and generates new pixel values. In the diffusion phase, the permutated image is encrypted using two ciphering matrices.

2.1. Permutation. Here, let $\mathbf{I}_0 = \{I_0(i,j)\}_{i=1,j=1}^{m,\ m}$ denote a gray-scale plain image, and let

$$\mathbf{I}_0(i,:) = [I_0(i,1), I_0(i,2), \ldots, I_0(i,m)] \tag{1}$$

be the ith row of $\mathbf{I}_0$.

The permutation phase is performed as follows.

Step 1. Decompose $\mathbf{I}_0(i,:)$ into a vector $\mathbf{I}_{0i} = \{I_{0i}(k)\}_{k=1}^{8m}$ as follows:

$$\mathbf{I}_{0i} = \text{dec2bin}(\mathbf{I}_0(i,:)), \tag{2}$$

where the function dec2bin converts the decimal number to a binary number.

Step 2. Compute the circular shift steps N_{shifts} and direction D_{shifts} of each row.

$$N_{\text{shifts}} = \sum_{k=1}^{8m} I_{0i}(k), \tag{3}$$

where N_{shifts} is the number of 1s of each row.

$$D_{\text{shifts}} = \text{mod}(N_{\text{shifts}}, 2). \tag{4}$$

Each vector $\mathbf{I}_{0i}$ is right circular-shifted with N_{shifts} steps if D_{shifts} equals 0 and left circular-shifted if D_{shifts} equals 1.

Step 3. Each group of 8 bits of vector $\mathbf{I}_{0i}$ is converted to a decimal number step by step; thus, the permuted image $\mathbf{I}_s = \{I_s(i,j)\}_{i=1,j=1}^{m,\ m}$ can be obtained:

$$\mathbf{I}_s(i,:) = \text{bin2dec}(\mathbf{I}_{0i}), \tag{5}$$

where $\mathbf{I}_s(i,:)$ denotes the ith row of the permuted image and the function bin2dec converts the binary number to a decimal number.

2.2. Diffusion. The final cipher image $\mathbf{I}_{\text{ENC}} = \{I_{\text{ENC}}(i,j)\}_{i=1,j=1}^{m,\ m}$ is obtained as follows.

For each row $\mathbf{I}_{\text{ENC}}(i,:)$,

$$\mathbf{I}_{\text{ENC}}(i,:) = \mathbf{I}_s(i,:) \oplus \mathbf{I}_{\text{cipher\_row}}(i,:), \tag{6}$$

and for each column $\mathbf{I}_{\text{ENC}}(:,j)$,

$$\mathbf{I}_{\text{ENC}}(:,j) = \mathbf{I}_{\text{ENC}}(:,j) \oplus \mathbf{I}_{\text{cipher\_col}}(j,:)', \tag{7}$$

where symbol $\oplus$ represents a bitwise XOR operation, and $\mathbf{I}_{\text{cipher\_row}} = \{I_{\text{cipher\_row}}(i,j)\}_{i=1,j=1}^{m,\ m}$ and $\mathbf{I}_{\text{cipher\_col}} = \{I_{\text{cipher\_col}}(i,j)\}_{i=1,j=1}^{m,\ m}$ are two ciphering matrices.

2.3. Computing Ciphering Matrices. Two ciphering matrices are computed by the pseudorandom number generator PRNG (8), which has been proven to have attractor's fractal structure, ergodicity, and an enormous key space [24].

$$y_i = f_{r_1}(x_i^1) * f_{r_2}(x_i^2) = \frac{f_{r_1}(x_i^1) + f_{r_2}(x_i^2)}{1 - f_{r_1}(x_i^1) f_{r_2}(x_i^2)}, \tag{8}$$

where r_1, r_2 are the control parameters, x_0^1, x_0^2 are the initial values, and x_i^1, x_i^2 can be generated by the following equation:

$$x_{i+1}^{\omega} = f_{r_\omega}(x_i^{\omega}) = \frac{2}{\pi}\,\text{arctg}(\text{ctg}(x_i^{\omega} \cdot r_\omega)), \quad \omega = 1, 2. \tag{9}$$

A chaotic sequence of size $8m^2$ can be generated using (8) and (9). Then, the sequence is discretized into dibits by using four thresholds. The high and low bits of all dibits are arranged in two vectors $\mathbf{h} = \{h(t)\}_{t=1}^{8m^2}$ and $\mathbf{l} = \{l(t)\}_{t=1}^{8m^2}$, respectively.

In [17], $x_0^1 = 0.68775492511773$, $x_0^2 = -0.0134623354671$, $r_1 = 5.938725025421$, and $r_2 = 1.257490188615$; thus, the key is denoted as $\mathbf{K} = [x_0^1, x_0^2, r_1, r_2]$.

Based on the definition above, two ciphering matrices are obtained as follows.

Step 1. Initialize

$$\begin{gathered} \mathbf{I}_{\text{cipher\_row}} = \mathbf{I}_{\text{cipher\_col}} = \mathbf{O}, \\ n = 0. \end{gathered} \tag{10}$$

Step 2. For $i = 1 : m$ and $j = 1 : m$

(a) take eight consecutive bits separately from **h** and **l** in turn,

$$\begin{gathered} \mathbf{B}_{\text{col}} = \text{strcat}(h(n+k)) \quad k = 1, \ldots, 8, \\ \mathbf{B}_{\text{row}} = \text{strcat}(l(n+k)) \quad k = 1, \ldots, 8, \end{gathered} \tag{11}$$

(b) get the element of the ith row and jth column,

$$I_{\text{cipher\_col}}(i,j) = \text{bin2dec}(\mathbf{B}_{\text{col}}),$$
$$I_{\text{cipher\_row}}(i,j) = \text{bin2dec}(\mathbf{B}_{\text{row}}), \tag{12}$$

(c) update the variable n,

$$n = n + 8. \tag{13}$$

3. Cryptanalysis of the Original Scheme

The original algorithm proposed circular inter-intra-pixels bit-level permutation. Within this algorithm, the chaotic map is highly sensitive to the four parameters and a large key space can be obtained. However, the circular shift process has a dangerous vulnerability.

Equations (6) and (7) can be transformed into the following expressions:

$$\mathbf{I}_{\text{ENC}} = \mathbf{I}_s \oplus \mathbf{I}_{\text{cipher\_row\_col}}, \tag{14}$$

$$\mathbf{I}_{\text{cipher\_row\_col}} = \mathbf{I}_{\text{cipher\_row}} \oplus \mathbf{I}'_{\text{cipher\_col}}. \tag{15}$$

According to Kerckhoff's principle, the security of a cryptosystem should only depend on the secrecy of the key; that is, everything else in the system can be public knowledge. In other words, the original scheme is assumed to be known, but the key is unknown. Obviously, it is nearly impossible to obtain the key used in Section 2.3. Fortunately, the chaotic secret matrix $\mathbf{I}_{\text{cipher\_row\_col}}$ in (14) is unchanged if the key is fixed. Thus, our goal is to obtain this matrix, which can be realized by a known-plaintext attack.

Known-plaintext attacks assume that the plaintext and corresponding cipher are known. If a known-plaintext attack is used to break the original scheme, we must first obtain the scrambled image of a known plain image. Fortunately, the scrambled image can be computed easily by circular shifting.

Here, consider a known plain image $\mathbf{I}_k$, whose corresponding scrambled image and cipher image are $\mathbf{S}_k$ and $\mathbf{C}_k$, respectively. The scrambled image $\mathbf{S}_k$ can be computed easily using the procedure described in Section 2.1. With the cipher image $\mathbf{C}_k$ and the computed $\mathbf{S}_k$, the secret matrix $\mathbf{I}_{\text{cipher\_row\_col}}$ can be obtained using

$$\mathbf{I}_{\text{cipher\_row\_col}} = \mathbf{C}_k \oplus \mathbf{S}_k, \tag{16}$$

because $\mathbf{C}_k \oplus \mathbf{S}_k = \mathbf{S}_k \oplus \mathbf{I}_{\text{cipher\_row}} \oplus \mathbf{I}'_{\text{cipher\_col}} \oplus \mathbf{S}_k = \mathbf{I}_{\text{cipher\_row}} \oplus \mathbf{I}'_{\text{cipher\_col}} = \mathbf{I}_{\text{cipher\_row\_col}}$.

If the secret matrix $\mathbf{I}_{\text{cipher\_row\_col}}$ is available, any given cipher image to be decrypted can be restored into the scrambled image $\mathbf{S}_k$. Since the number of 1s in each row of the scrambled image is the same as that of the plain image, the permutation can be reversed easily, and the deciphering image can be obtained.

The steps of a known-plaintext attack are as follows.

Step 1. Obtain the secret matrix.

Step 1.1. Calculate $\mathbf{S}_k$ (Section 2.1) with the known plain image $\mathbf{I}_k$.

Step 1.2. Obtain the secret matrix $\mathbf{I}_{\text{cipher\_row\_col}}$ using (16).

Step 2. Decrypt the given cipher image.

Step 2.1. With the given cipher image $\mathbf{I}_{\text{ENC}}$, calculate the scrambled image $\mathbf{I}_s$ using (17).

$$\begin{aligned}\mathbf{I}_{\text{ENC}} \oplus \mathbf{I}_{\text{cipher\_row\_col}} &= \mathbf{I}_s \oplus \mathbf{I}_{\text{cipher\_row}} \oplus \mathbf{I}'_{\text{cipher\_col}} \\ &\quad \oplus \mathbf{I}_{\text{cipher\_row\_col}} \\ &= \mathbf{I}_s \oplus \mathbf{I}_{\text{cipher\_row}} \oplus \mathbf{I}'_{\text{cipher\_col}} \\ &\quad \oplus \mathbf{I}_{\text{cipher\_row}} \oplus \mathbf{I}'_{\text{cipher\_col}} = \mathbf{I}_s.\end{aligned} \tag{17}$$

Step 2.2. Obtain the deciphering image $\mathbf{I}_0$ by applying the opposite circular shift to $\mathbf{I}_s$; that is, if $D_{\text{shifts}} = 0$, then each row of $\mathbf{I}_s$ should be left circular-shifted, and if $D_{\text{shifts}} = 1$, then each row of $\mathbf{I}_s$ should be right circular-shifted.

To further demonstrate this idea, experimental results are shown in Figures 1 and 2. Figure 1 illustrates how to obtain the secret matrix. Assume that a known plain image is Lena (Figure 1(a)). The cipher image corresponding to Figure 1(a) is shown in Figure 1(b). The circular-shifted image, that is, the permutated image $\mathbf{S}_k$, is shown in Figure 1(c). The secret matrix $\mathbf{I}_{\text{cipher\_row\_col}}$ computed by (16) is shown in Figure 1(d). The deciphering process of the given cipher image is shown in Figure 2. Here, Figure 2(a) is the plain image and its corresponding cipher image to be deciphered is shown in Figure 2(b). The scrambled image, which is obtained by an XOR operation between the secret matrix $\mathbf{I}_{\text{cipher\_row\_col}}$ (Figure 1(d)) and the cipher image (Figure 2(b)), is shown in Figure 2(c). Then, after inverse circular shifting, the deciphering image is obtained which is shown in Figure 2(d). Note that the deciphering image is the same as the plain image.

4. Analysis, Discussion, and Improvement

4.1. Improving Permutation

4.1.1. Vulnerability to Known-Plaintext Attack. In the original scheme, the shift steps and direction are decided by the number of 1s of each row. After permutation, the number of 1s of each row remains unchanged. Therefore, the scrambled image for any known plain image can be computed, which is the primary vulnerability of the original encryption scheme.

4.1.2. Improvement. Randomizing the shift steps and direction of each row can remove this vulnerability. Here, a secret sequence must be introduced to satisfy this requirement. For simplicity, the chaos map based on (8) and (9) is used. If the plain image has m rows, the number of pairs of steps and directions is m. The permutation phase is improved as follows.

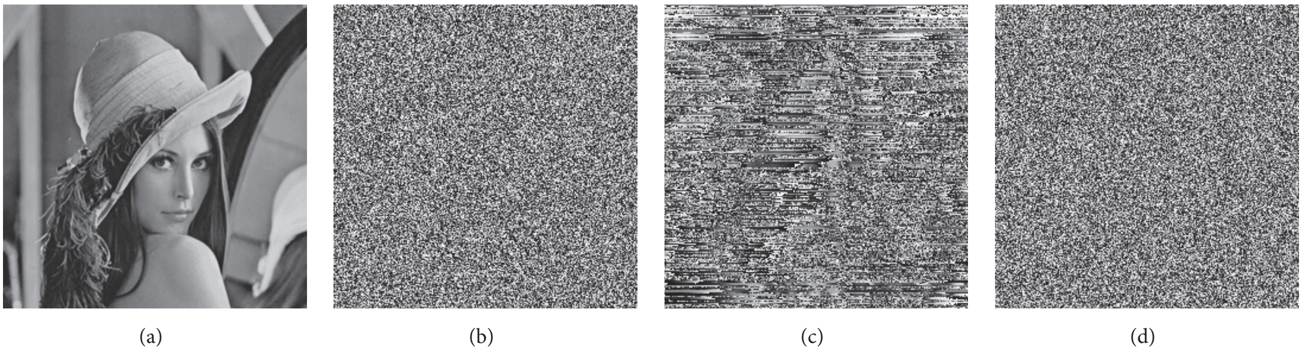

Figure 1: Step 1 of a known-plaintext attack: (a) known plain image; (b) cipher image corresponding to (a); (c) scrambled image corresponding to (a); and (d) secret matrix.

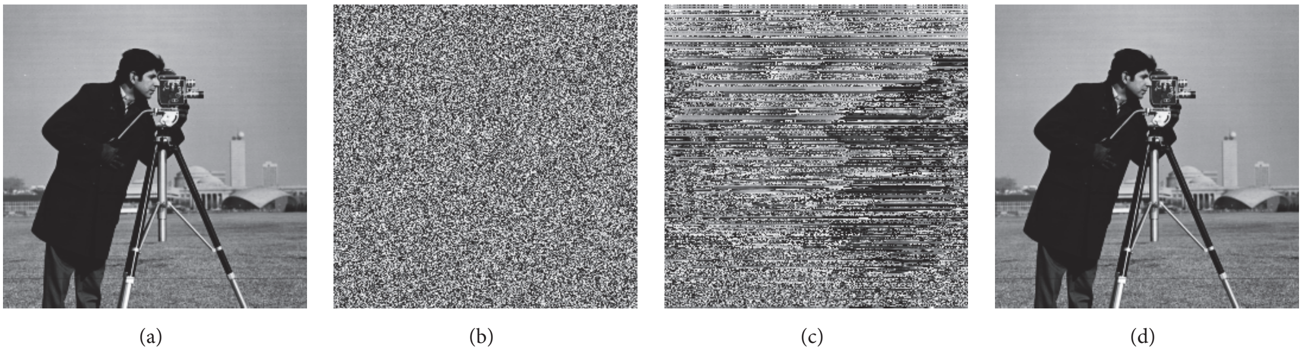

Figure 2: Step 2 of a known-plaintext attack: (a) plain image; (b) cipher image corresponding to (a); (c) scrambled image retrieved from (b); and (d) deciphering image of (b).

First, a random sequence of length $m * e$ is generated by iterations, where the parameter e is computed by (18) and $8m$ is the bits of each row of the plain image.

$$e = \log_2(8m). \tag{18}$$

Then, the sequence is discretized into dibits using four thresholds. The high and low bits of the dibits are arranged into vectors $\mathbf{h}_s = \{h_s(t)\}_{t=1}^{m*e}$ and $\mathbf{l}_s = \{l_s(t)\}_{t=1}^{m*e}$, separately.

Assume that $\mathbf{d}_s = \{d_s(i)\}_{i=1}^{m}$ denotes the shift direction (left or right) and $\mathbf{p}_s = \{p_s(i)\}_{i=1}^{m}$ denotes the steps of shift. Based on the definition above, the two vectors can be computed as follows.

Step 1. Initialize the vectors $\mathbf{d}_s$ and $\mathbf{p}_s$ and the counter variable n,

$$\begin{gathered}\mathbf{d}_s = \mathbf{p}_s = \mathbf{O}, \\ n = 0.\end{gathered} \tag{19}$$

Step 2. (a) Take e consecutive bits separately from $\mathbf{h}_s$ and $\mathbf{l}_s$ in turn,

$$\begin{aligned}\mathbf{B}_d &= \text{strcat}\left(h_s(n+k)\right) \quad k = 1, \ldots, e, \\ \mathbf{B}_l &= \text{strcat}\left(l_s(n+k)\right) \quad k = 1, \ldots, e.\end{aligned} \tag{20}$$

(b) Obtain the ith pair of elements of the vectors $\mathbf{d}_s$ and $\mathbf{p}_s$,

$$\begin{aligned}d_s(i) &= \text{mod}\left(\text{bin2dec}\left(\mathbf{B}_d\right), 2\right), \\ p_s(i) &= \text{bin2dec}\left(\mathbf{B}_l\right).\end{aligned} \tag{21}$$

(c) Update the variable n,

$$n = n + e. \tag{22}$$

The permutation process is similar to the process described in Section 2.1. Here, vector $\mathbf{I}_{0i}$ is right circular-shifted with $p_s(i)$ steps if $d_s(i)$ equals 0 and left circular-shifted if $d_s(i)$ equals 1.

Since the scrambled image cannot be computed directly from the plain image, the known-plaintext attack would not break the original encryption scheme. However, the following analysis shows that it is insufficient to improve the permutation phase solely.

4.2. Improving Diffusion

4.2.1. Vulnerability to Chosen-Plaintext Attack. After improving permutation, there are two secret sequences. One sequence generates the matrix $\mathbf{I}_{\text{cipher\_row\_col}}$ used in the diffusion phase and the other generates $\mathbf{h}_s$ and $\mathbf{l}_s$ used in the permutation phase. Compared to the original algorithm, security is increased; however, the improved encryption scheme can still be broken by a chosen-plaintext attack.

If $\mathbf{I}_z$ is a matrix where all elements are zero, the permutation phase has no influence on the $\mathbf{I}_z$. Then, the encrypted image $\mathbf{I}_{Z\_\mathrm{ENC}}$ can be expressed as follows:

$$\begin{aligned}\mathbf{I}_{Z\_\mathrm{ENC}} &= \mathbf{I}_z \oplus \mathbf{I}_{\mathrm{cipher\_row\_col}} = \mathbf{O} \oplus \mathbf{I}_{\mathrm{cipher\_row\_col}} \\ &= \mathbf{I}_{\mathrm{cipher\_row\_col}}.\end{aligned} \tag{23}$$

Here, we find that the secret matrix $\mathbf{I}_{\mathrm{cipher\_row\_col}}$ is equal to the encryption output $\mathbf{I}_{Z\_\mathrm{ENC}}$. Next, random vectors $\mathbf{h}_s$ and $\mathbf{l}_s$ used in the permutation phase can be computed by using a matrix $\mathbf{I}_1$, which is defined as follows:

$$\mathbf{I}_1 = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 1 \\ \cdots & & \cdots & & \cdots \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}. \tag{24}$$

The cipher and scrambled images corresponding to $\mathbf{I}_1$ are denoted as $\mathbf{I}_{1\_\mathrm{ENC}}$ in (25) and $\mathbf{I}_{1s}$, respectively:

$$\mathbf{I}_{1\_\mathrm{ENC}} = \mathbf{I}_{1s} \oplus \mathbf{I}_{\mathrm{cipher\_row\_col}}. \tag{25}$$

The secret matrix $\mathbf{I}_{\mathrm{cipher\_row\_col}}$ is calculated based on (23). Then, we obtain the scrambled image $\mathbf{I}_{1s}$ by applying an XOR operation between the cipher image $\mathbf{I}_{1\_\mathrm{ENC}}$ and the secret matrix $\mathbf{I}_{\mathrm{cipher\_row\_col}}$.

$$\begin{aligned}\mathbf{I}_{1s} &= \mathbf{I}_{1\_\mathrm{ENC}} \oplus \mathbf{I}_{\mathrm{cipher\_row\_col}} \\ &= \mathbf{I}_{1s} \oplus \mathbf{I}_{\mathrm{cipher\_row\_col}} \oplus \mathbf{I}_{\mathrm{cipher\_row\_col}}.\end{aligned} \tag{26}$$

By transforming each element of $\mathbf{I}_{1s}$ into an 8-bit binary number, we can obtain a matrix $\mathbf{I}_{1s}^b$ of size $m \times 8m$, which has only one "1" in each row. Note that we must obtain the column position of "1." For example, if the element "1" in the ith row is located at the jth column, we can conclude that the ith row of $\mathbf{I}_1$ has been right circular-shifted with j steps. Assume that all the shift steps are saved in a vector $\mathbf{l}_d = \{l_d(i)\}_{i=1}^m$, where $l_d(i) = j$ can be obtained easily.

The deciphering steps are described as follows.

Step 1. Obtain $\mathbf{I}_{\mathrm{cipher\_row\_col}}$ via the chosen plain image $\mathbf{I}_z$.

Step 2. Obtain the scrambled image $\mathbf{I}_s$ by an XOR operation between $\mathbf{I}_{\mathrm{ENC}}$ and $\mathbf{I}_{\mathrm{cipher\_row\_col}}$ based on (17).

Step 3. Obtain the permutation steps vector $\mathbf{l}_d$ using the chosen plain image $\mathbf{I}_1$.

Step 4. Perform opposite circular shift on $\mathbf{I}_s$; that is, the ith row of the scrambled image $\mathbf{I}_s$ is circularly shifted towards the left direction with $l_d(i)$ steps.

4.2.2. Improvement. The deciphering algorithm (Section 4.2.1) is performed successfully because it is easy to calculate the secret matrix $\mathbf{I}_{\mathrm{cipher\_row\_col}}$. If we design a method to prevent an attacker from obtaining this matrix, the encryption would be secure. Therefore, the diffusion phase should also be improved. An operation, called double crossover diffusion (DCD) [20], is added to the end of the original scheme.

Assume that $\mathbf{I}_v = \{I(t)\}_{t=1}^{m^2}$ is a stretched vector of cipher image $\mathbf{I}_{\mathrm{ENC}}$. The DCD mechanism requires a random sequence, which is generated by the logistic map.

$$u_{i+1} = \alpha u_i (1 - u_i), \quad u_i \in (0, 1). \tag{27}$$

If parameter α is chosen in the range $[3.5699456, 4]$ and the initial value u_0 is in the range $[0, 1]$, the system is in a chaotic state. We can run the logistic map from u_0 to generate a chaotic sequence $\mathbf{u} = \{u(t)\}_{t=1}^{m^2}$. Then, by sorting this vector, a vector $\mathbf{q} = \{q(t)\}_{t=1}^{m^2}$ is obtained, where $u(q(t))$ is the tth largest element of the chaotic sequence $\mathbf{u}$.

The general structure of the improved procedure is presented in Algorithms 1 and 2. As shown in Algorithm 1, the cipher image is first reshaped to a vector $\mathbf{I}_v$ and ordered by random sequence $\mathbf{q}$. As shown in Algorithm 2, the sorted vector $\mathbf{I}_v$ is divided into two blocks and encrypted sequentially by the left and right parts. The key vector **key** in one block (Algorithm 2) is connected to the vector $\mathbf{q}$ and the prior encrypted pixels of the other block.

5. Performance of the Improved Cryptosystem

This section presents experimental results and analyzes the performance of the improved cryptosystem. Here, we obtained plain images from the USC-SIPI image database [25]. Note that images of other sizes can also be encrypted effectively by the proposed algorithm. The PRNG parameters are $x_0^1 = 0.68775492511773$, $x_0^2 = -0.0134623354671$, $r_1 = 5.938725025421$, and $r_2 = 1.257490188615$. The logistic map parameters are $\alpha = 3.99$ and $u_0 = 0.2113$. However, the parameter C_0 in crossover diffusion is 52, which is not considered as the key in this paper. Thus, there are six parameters in the key vector $\mathbf{K}_1 = [x_0^1, x_0^2, r_1, r_2, \alpha, u_0]$, where α, r_1, r_2 are the control parameters and x_0^1, x_0^2, u_0 are the initial parameters. All experiments were implemented by using MATLAB R2010b and were executed on a personal computer (Intel Core i5-5300U 2.3 GHz CPU, 4 GB memory).

5.1. Security

5.1.1. Key Space. Key space is a significant index to measure the ability to resist brute-force attacks and an effective key space must contain very large and unique keys. Considering the type of PRNG and the logistic map, the six parameters can be expressed by double-real precision (15 decimals). Thus, the key space can achieve $(10^{15})^6 = 10^{90}$, which is nearly 299 bits. As a result, the size of the key space in the improved algorithm is sufficient to resist exhaustive attacks.

5.1.2. Key Sensitivity. An important security indicator for an image encryption system is the sensitivity to the key; that is, small changes to the key lead to significant changes in the cipher image. Assume that a new key $\mathbf{K}_2$ can be

```
Read the cipher image $\mathbf{I}_{\mathrm{ENC}}$, the vector $\mathbf{q}$ from input. Reshape $\mathbf{I}_{\mathrm{ENC}}$ into a vector $\mathbf{I}_v$.
for $t = 1 : m^2$
    $\mathbf{I}_v(t) = \mathbf{I}_v(q(t))$.
end
$\mathbf{I}_v$ = double_crossover_diffusion ($\mathbf{I}_v$, $\mathbf{q}$).
for $t = 1 : m^2$
    $\mathbf{I}_v(q(i)) = \mathbf{I}_v(i)$.
end
Reshape $\mathbf{I}_v$ into a new matrix $\mathbf{I}_{\mathrm{ENC}}$ of size $m \times m$.
```

Algorithm 1: Overall structure of DCD.

```
Read the vector $\mathbf{I}_v$ and $\mathbf{q}$ from input, set $C_0 = 52$ and $le = m^2$.
$Key(1) = \mathrm{mod}(C_0 + q(1), 256)$.
$C(1) = \mathrm{bitxor}(I_v(1), Key(1))$.
$Key(le/2 + 1) = \mathrm{mod}(C(1) + q(le/2 + 1), 256)$.
$C(le/2 + 1) = \mathrm{bitxor}(I_v(le/2 + 1), Key(le/2 + 1))$.
for $i = 2 : le/2$
    $Key(i) = \mathrm{mod}(C(le/2 + i - 1) + q(i), 256)$.
    $C(i) = \mathrm{bitxor}(I_v(i), Key(i))$.
    $Key(le/2 + i) = \mathrm{mod}(C(i) + q(le/2 + i), 256)$.
    $C(le/2 + i) = \mathrm{bitxor}(I_v(le/2 + i), Key(le/2 + i))$.
end
$Key(1) = \mathrm{mod}(C(le) + q(1), 256)$.
$C(1) = \mathrm{bitxor}(C(1), Key(1))$.
$Key(le/2 + 1) = \mathrm{mod}(C(1) + q(le/2 + 1), 256)$.
$C(le/2 + 1) = \mathrm{bitxor}(C(le/2 + 1), Key(le/2 + 1))$.
for $i = 2 : le/2$
    $Key(i) = \mathrm{mod}(C(le/2 + i - 1) + q(i), 256)$.
    $C(i) = \mathrm{bitxor}(C(i), Key(i))$.
    $Key(le/2 + i) = \mathrm{mod}(C(i) + q(le/2 + i), 256)$.
    $C(le/2 + i) = \mathrm{bitxor}(C(le/2 + i), Key(le/2 + i))$.
end
$\mathbf{I}_v = \mathbf{C}$.
```

Algorithm 2: Double_crossover_diffusion function.

obtained by transforming $\mathbf{K}_1$ (adding 10^{-15} to r_1 and leaving the other parameters unchanged). Figures 3 and 4 represent the improved algorithm's sensitivity to the key. The contrasts before and after adding 10^{-15} to r_1 are presented in Figure 3, and it shows that small changes to the key generate an entirely different cipher images. Figure 4 shows that a correct secret key can achieve a completely accurate decryption, while a key with a small change cannot.

Likewise, let $\mathbf{E}_1 = \{E_1(i,j)\}_{i=1,j=1}^{m,\ m}$ and $\mathbf{E}_2 = \{E_2(i,j)\}_{i=1,j=1}^{m,\ m}$ represent two cipher images using different keys $\mathbf{K}_1$ (the initial key) and $\mathbf{K}_2$ (adding 10^{-15} to u_0). To evaluate quantitatively the sensitivity to the key, the percentage of different pixels of the two cipher images is computed. Simulation results are shown in Table 1. As shown in Table 1, over 99.5% of the pixels of one encrypted image differ from the pixels of the other image. Thus, the improved encryption scheme is highly sensitive to the secret key.

5.1.3. Differential Attack Analysis. The ability to resist differential attacks depends on the "sensitivity to the changes" in plain image. Stronger sensitivity affords greater resistance to differential attacks. Let $\mathbf{P}_1 = \{P_1(i,j)\}_{i=1,j=1}^{m,\ m}$ denote a plain image. After selecting a pixel of this plain image randomly and flipping its least significant bit, we can obtain a new plain image $\mathbf{P}_2 = \{P_2(i,j)\}_{i=1,j=1}^{m,\ m}$. The corresponding cipher images are $\mathbf{E}_1 = \{E_1(i,j)\}_{i=1,j=1}^{m,\ m}$ and $\mathbf{E}_2 = \{E_2(i,j)\}_{i=1,j=1}^{m,\ m}$, respectively. To represent the sensitivity of the encryption scheme to plain image, two parameters, the Number of Pixels Change Rate (NPCR) and the Unified Average Changing Intensity (UACI), are introduced and defined by

$$\begin{aligned} \mathrm{NPCR} &= \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i,j) \times 100\%, \\ \mathrm{UACI} &= \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \frac{\left| E_1(i,j) - E_2(i,j) \right|}{2^n - 1} \times 100\%, \end{aligned} \tag{28}$$

where n is the bits of the pixels and M and N represent the width and height of the test images, respectively. If $E_1(i,j)$ is equal to $E_2(i,j)$, then $D(i,j) = 0$; otherwise, $D(i,j) = 1$. Ideal NPCR and UACI values can be calculated by the following equations, respectively.

$$\mathrm{NPCR}_E = \left(1 - 2^{-n}\right) \times 100\%, \tag{29}$$

$$\mathrm{UACI}_E = \frac{1}{2^{2n}} \frac{\sum_{i=1}^{2^n - 1} i(i+1)}{2^n - 1} \times 100\%. \tag{30}$$

If $n = 8$, the ideal NPCR and UACI values are 99.6094% and 33.4635%, respectively.

The NPCR and UACI values of the five images are shown in Table 2. To demonstrate the broad applicability of the improved scheme, we used different natural plain images and selected pixel multiple times randomly to obtain the average values. Note that our improved scheme requires only one-round encryption process and ideal values of NPCR and

Table 1: Percentage difference between two cipher images before and after adding 10^{-15} to u_0 (%).

Image	Boat	Baboon	Couple	Peppers	Lena	Cameraman
Difference	99.62	99.61	99.61	99.62	99.59	99.58

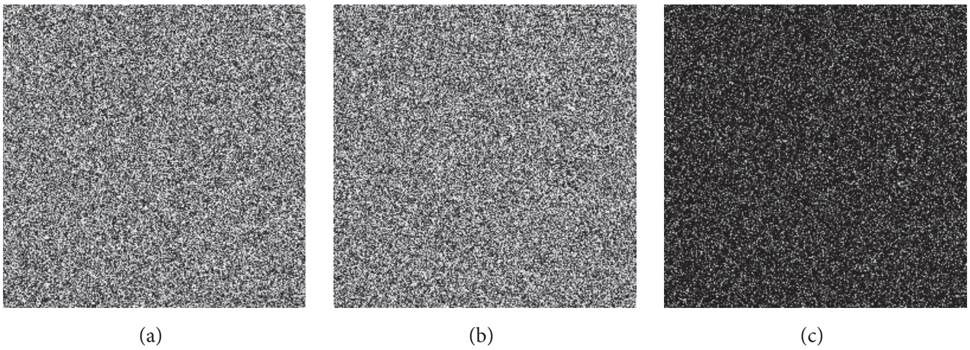
(a) (b) (c)

Figure 3: Sensitivity to small changes in the key: (a) image encrypted by key $\mathbf{K}_1$; (b) image encrypted by key $\mathbf{K}_2$; and (c) difference image between (a) and (b).

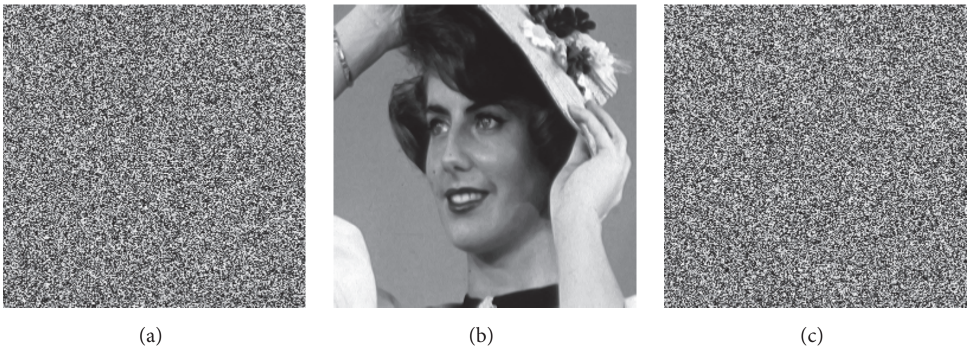
(a) (b) (c)

Figure 4: Image decryption using correct and incorrect keys: (a) image encrypted by key $\mathbf{K}_1$; (b) deciphering image by correct key $\mathbf{K}_1$; and (c) deciphering image by incorrect key $\mathbf{K}_2$.

Table 2: NPCR and UACI values of different images by different algorithms.

Image	NPCR (%)				
	Ref. [21]	Ref. [22]	Ref. [23]	Original	Improved
Lena	99.61	99.55	99.61	**99.62**	99.60
Peppers	99.61	99.58	**99.62**	99.61	99.61
Couple	99.63	99.56	99.63	99.63	**99.64**
Cameraman	**99.63**	99.57	99.61	99.60	**99.63**
Boat	**99.63**	99.56	99.62	99.61	99.59
Image	UACI (%)				
	Ref. [21]	Ref. [22]	Ref. [23]	Original	Improved
Lena	33.56	33.34	33.63	33.48	**33.47**
Peppers	33.63	33.35	33.74	33.51	**33.40**
Couple	33.64	33.37	33.72	33.49	**33.45**
Cameraman	33.71	33.37	33.69	**33.52**	33.59
Boat	33.61	33.42	33.60	**33.49**	33.32

UACI have been achieved without encrypting more than one round. As shown in Table 2, the NPCR values are close to the ideal value (i.e., 99.6094%), and the UACI values range from 33.3% to 33.6%. Relative to the NPCR, the proposed scheme is better than [22] and comparable to [17, 21, 23]. For UACI, the improved scheme demonstrates the best performance. Hence, changing one pixel of plain image influences nearly all pixels of the cipher image with the improved scheme, and this demonstrates the better security property of the proposed scheme against differential attacks.

5.2. Computational and Complexity Analysis. The permutation phase of the improved scheme is the same as that of the original scheme. As a result, the total complexity of the permutation phase is approximate to $\Theta(8 \times m \times m)$. The time complexity of the diffusion process is $\Theta(m \times m)$, which is similar to that of the original algorithm. Therefore, the total time complexity of the proposed algorithm is $\Theta(8 \times m \times m)$. The execution time of the improved scheme, the original scheme, and other algorithms are given in Table 3. For accuracy, we encrypted different natural images multiple times to obtain the average value by using different algorithms. As shown in Table 3, the operating speed of the improved scheme is better

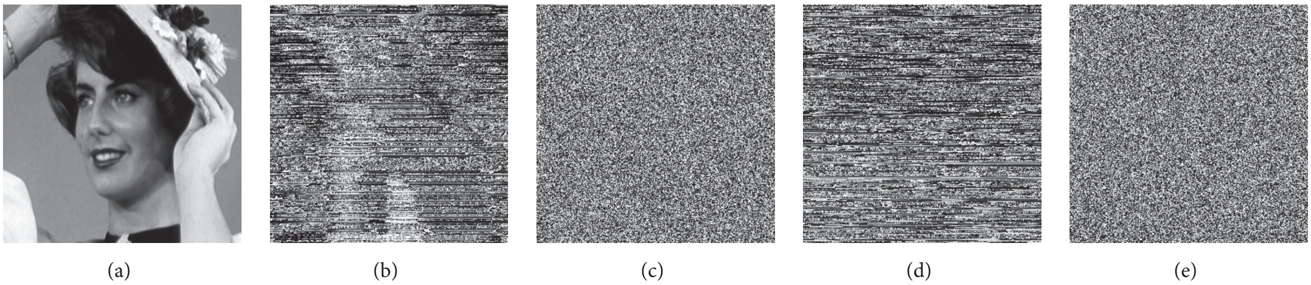

Figure 5: Comparison of original and improved algorithms: (a) plain image (lady); (b) scrambled image (original algorithm); (c) cipher image (original algorithm); (d) scrambled image (improved algorithm); and (e) cipher image (improved algorithm).

Table 3: Execution time.

Scheme	Image size			
	128 × 128	256 × 256	512 × 512	1024 × 1024
Ref. [21]	0.1895	0.3414	1.5026	9.3228
Ref. [22]	0.3125	0.5029	2.2335	11.7568
Ref. [23]	**0.1605**	**0.2921**	**1.3528**	**8.2878**
Original	0.2379	0.4261	2.0396	11.1652
Improved	0.1811	0.3384	1.4918	9.2796

than those in [17, 21, 22] but less than that in [23]. Moreover, the execution efficiency of the improved scheme is somewhat higher than that of the original scheme. It is because the shift steps and the direction of each row are determined by the available key vectors $\mathbf{d}_s$ and $\mathbf{p}_s$ in our permutation phase. In contrast, in the original scheme, they are computed row by row. Besides, the execution time of the improved algorithm would be at least an order of magnitude slower than those reported in [17, 21–23] when they were executed in a different environment without optimization.

5.3. Visual and Statistical Analysis

5.3.1. Histogram Analysis. Histogram analysis can demonstrate the uniformity of a distribution. A flat histogram indicates a good cipher image obtained by a secure encryption algorithm. The comparative experiments with the original and improved algorithms are shown in Figure 5, and the corresponding histograms are shown in Figure 6. In Figure 5(b), it retains more information of plain image than the one in Figure 5(d), because the original permutation is based on the certain numbers of 1s of each row, while the improved scheme is based on a secret sequence. If two adjacent rows are with equal numbers of 1s, they would be circular-shifted with the same steps along with the same direction. In Figures 6(c) and 6(e), one can deduce that the distributions of the two cipher images are nearly uniform. Therefore, both algorithms can resist statistical attack.

5.3.2. Correlation Analysis. A plain image typically has high neighbor correlation among pixels that can be weakened by an encryption algorithm. Note that better algorithms lead to weaker correlation. The correlations of adjacent pixels of plain image are compared to those of the cipher image, and the results are shown in Figures 7–9. Relative to the correlation, the improved algorithm is comparable to the original algorithm. The correlation coefficients are computed by

$$r_{xy} = \frac{\sum_{i=1}^{M}\left(x_i - (1/M)\sum_{j=1}^{M} x_j\right)\left(xn_i - (1/M)\sum_{j=1}^{M} xn_j\right)}{\sqrt{\sum_{i=1}^{M}\left(x_i - (1/M)\sum_{j=1}^{M} x_j\right)^2 \sum_{i=1}^{M}\left(xn_i - (1/M)\sum_{j=1}^{M} xn_j\right)^2}}, \quad (31)$$

where x_i and xn_i denote the ith pair of neighboring pixels of the image and M is the total number of pairs. The correlative coefficients of different algorithms are shown in Tables 4 and 5. As shown in Table 4, all correlation coefficients of the plain images are close to 1, and those of the cipher images are close to 0. Based on the data shown in Table 5, it can be concluded that the coefficients of the improved algorithm in the horizontal, vertical, and diagonal directions are less than those of algorithms found in [21–23], and the improved algorithm's coefficients in the horizontal and vertical directions (but not the diagonal direction) are less than those of the original algorithm. By using the original algorithm, coefficients in the horizontal direction are greater than those in the vertical and diagonal directions. It is because two adjacent rows with equal numbers of 1s are circular-shifted with the same steps in the same direction, which increases the correlation coefficient in the horizontal direction. This shortcoming of the original algorithm has been remedied by the proposed improved algorithm. The analysis above demonstrates that the improved algorithm can effectively break the correlation between adjacent pixels.

5.3.3. Information Entropy Analysis. Information entropy is an important indicator reflecting the randomness of information. A secure cipher image demonstrates large statistical randomness and large information entropy. The information entropy of an image can be computed as follows:

$$H(\mathbf{x}) = \sum_{i=0}^{2^n-1} p(x_i)\log_2\left(p(x_i)\right), \quad (32)$$

where $p(x_i)$ denotes the probability of pixel value x_i and 2^n is the total status number of $\mathbf{x}$. The signal source of entropy

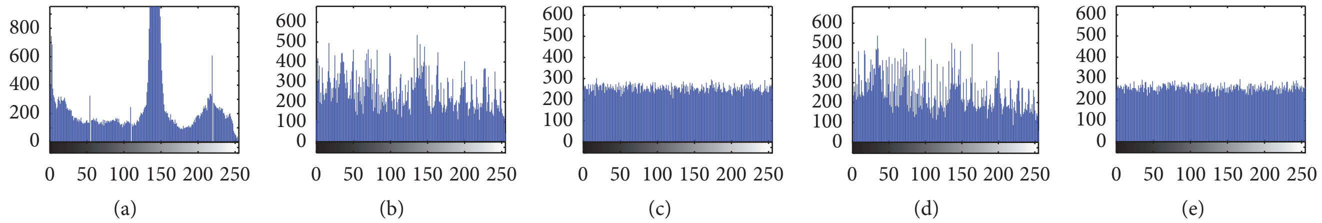

Figure 6: Histograms corresponding to the images in Figure 5.

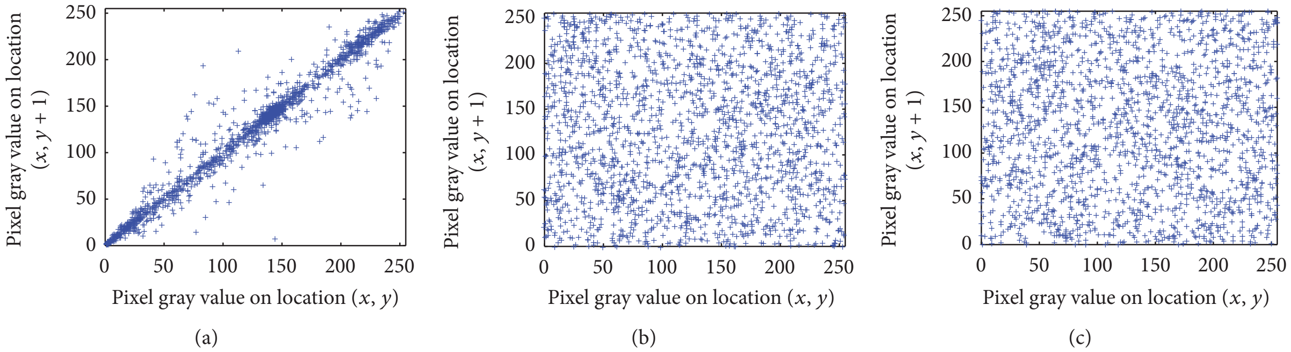

Figure 7: Correlation distribution of adjacent pixels in the horizon direction: (a) plain image; (b) cipher image (original algorithm); and (c) cipher image (improved algorithm).

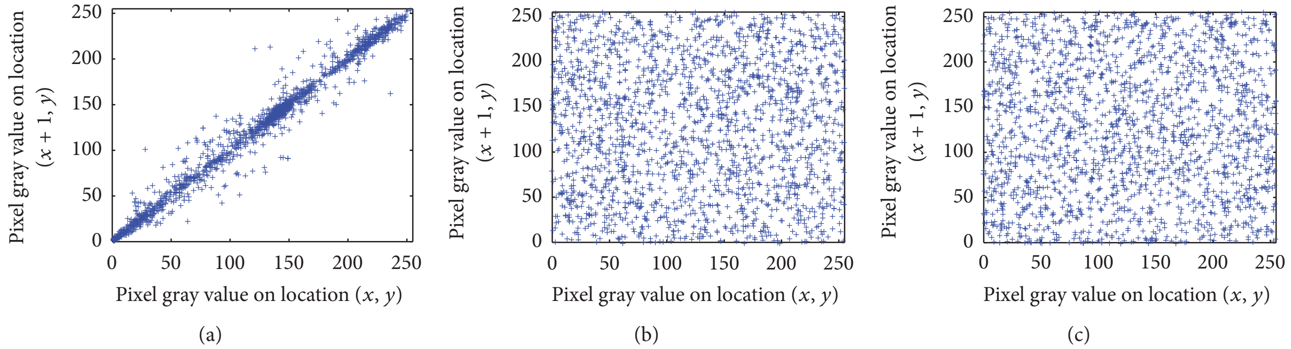

Figure 8: Correlation distribution of adjacent pixels in the vertical direction: (a) plain image; (b) cipher image (original algorithm); and (c) cipher image (improved algorithm).

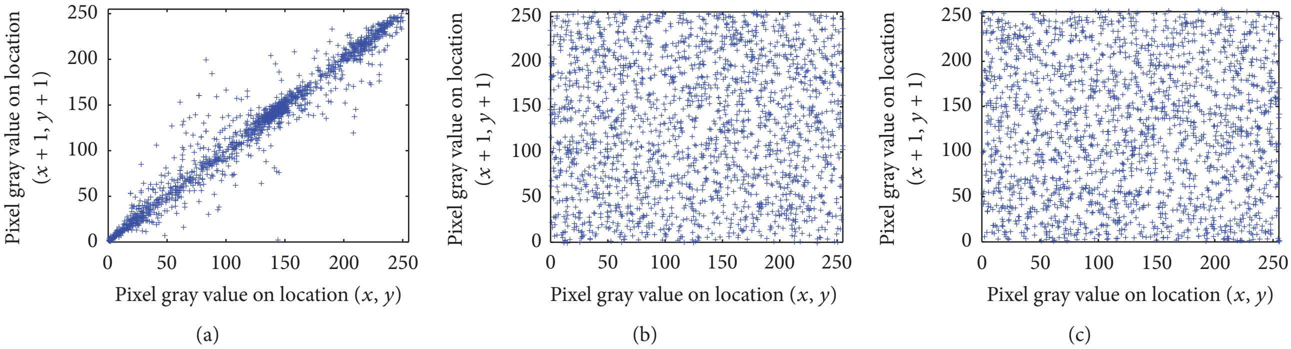

Figure 9: Correlation distribution of adjacent pixels in diagonal direction: (a) plain image; (b) cipher image (original algorithm); and (c) cipher image (improved algorithm).

Table 4: Correlation coefficients in different directions by different algorithms.

Image	Direction	Plain image	Algorithm				
			Ref. [21]	Ref. [22]	Ref. [23]	Original	Improved
Lena	Horizontal	0.9456	−0.0086	−0.0066	0.0011	0.0098	0.0057
	Vertical	0.9727	−0.0102	−0.0089	0.0098	−0.0013	−0.0001
	Diagonal	0.9213	−0.0125	0.0424	−0.0227	−0.0009	0.0031
Peppers	Horizontal	0.9664	−0.0089	0.0194	0.0071	0.0099	−0.0017
	Vertical	0.9738	−0.0113	−0.0091	−0.0065	−0.0016	−0.0033
	Diagonal	0.9423	0.0045	0.0123	−0.0165	−0.0012	0.0035
Couple	Horizontal	0.9299	−0.0178	−0.0122	−0.0236	0.0145	−0.0056
	Vertical	0.8878	−0.0025	0.0262	−0.0045	0.0035	−0.0017
	Diagonal	0.8382	0.0001	−0.0257	0.0016	0.0046	−0.0085
Cameraman	Horizontal	0.9335	−0.0211	0.0063	−0.0047	0.0138	0.0030
	Vertical	0.9592	−0.0103	−0.0142	−0.0195	0.0031	−0.0020
	Diagonal	0.9087	0.0054	0.0168	0.0279	−0.0010	0.0009
Boat	Horizontal	0.9494	−0.0054	−0.0189	−0.0295	0.0078	0.0028
	Vertical	0.9553	−0.0009	0.0003	−0.0150	0.0027	0.0008
	Diagonal	0.9118	0.0026	−0.0204	−0.0224	0.0050	0.0035

Table 5: Average correlation coefficients of different algorithms.

Direction	Algorithm				
	Ref. [21]	Ref. [22]	Ref. [23]	Original	Improved
Horizontal	0.01236	0.01268	0.01320	0.01177	**0.00398**
Vertical	0.00704	0.01174	0.00778	0.00208	**0.00143**
Diagonal	0.00502	0.02350	0.01822	**0.00322**	0.00415

Table 6: Information entropies of different images based on different algorithms.

Image	Plain image	Algorithm				
		Ref. [21]	Ref. [22]	Ref. [23]	Original	Improved
Lena	7.4347	7.9991	7.9951	7.9965	7.9980	7.9963
Peppers	7.5680	7.9973	7.9965	7.9958	7.9985	7.9985
Couple	7.1685	7.9987	7.9951	7.9980	7.9978	7.9976
Cameraman	7.0097	7.9966	7.9955	7.9964	7.9966	7.9985
Boat	7.0944	7.9979	7.9960	7.9959	7.9978	7.9981
Mean value		**7.9979**	7.9956	7.9965	7.9977	7.9978

n can generate 2^n characters. Thus, the ideal information entropy value should be 8 for a 256-gray-level cipher image. If the information entropy of a cipher image is close to 8, the distribution is close to a random distribution. The information entropies of different images obtained by using different algorithms are summarized in Table 6, which shows that the entropies of all encryption algorithms are close to 8. The improved algorithm has average information entropy of 7.9978, which is better than [17, 22, 23] and comparable to [21].

6. Conclusion

This paper has analyzed a recent chaos-based image encryption algorithm with circular inter-intra-pixels bit-level permutation. It was found that the scheme can be cracked by known-plaintext attacks. In addition, two defects in the original scheme have been identified and discussed. One is the fixed steps and directions of the circular shift, which causes the vulnerability to known-plaintext attacks. The other is the lack of secure diffusion, which causes the vulnerability to chosen-plaintext attacks. To improve the security of the original scheme, we employ the PRNG in the improved scheme to generate random directions and steps in the permutation process and execute the single-round DCD in the diffusion phase. Moreover, the performances of the improved scheme, including statistical, differential, adjacent correlative, and time complexity analyses, have been tested experimentally. The improved scheme has been proven superior or at least comparable to the original and other existing schemes. In the future, we plan to optimize the newly proposed scheme to improve encryption speed.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants 61602158 and U1404604), the China Postdoctoral Science Foundation (Grant 2016M600030), the Key Program of the Higher Education Institutions of Henan Province (Grant 14A520086), the Science Foundation for Young Scholars of Henan Normal University (Grant 2014QK20), and the Ph.D. Scientific Research Foundation of Henan Normal University (Grant qd14134).

References

[1] Y. Zhang, D. Xiao, W. Wen, and H. Nan, "Cryptanalysis of image scrambling based on chaotic sequences and Vigenère cipher," *Nonlinear Dynamics*, vol. 78, no. 1, pp. 235–240, 2014.

[2] F. Özkaynak and S. Yavuz, "Analysis and improvement of a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system," *Nonlinear Dynamics*, vol. 78, no. 2, pp. 1311–1320, 2014.

[3] H. Liu and Y. Liu, "Security assessment on block-Cat-map based permutation applied to image encryption scheme," *Optics and Laser Technology*, vol. 56, pp. 313–316, 2014.

[4] R. Boriga, A. C. Dăscălescu, and I. Priescu, "A new hyperchaotic map and its application in an image encryption scheme," *Signal Processing: Image Communication*, vol. 29, no. 8, pp. 887–901, 2014.

[5] J.-X. Chen, Z.-L. Zhu, C. Fu, L.-B. Zhang, and Y. Zhang, "An efficient image encryption scheme using lookup table-based confusion and diffusion," *Nonlinear Dynamics*, vol. 81, no. 3, pp. 1151–1166, 2015.

[6] M. Zanin and A. N. Pisarchik, "Gray code permutation algorithm for high-dimensional data encryption," *Information Sciences*, vol. 270, pp. 288–297, 2014.

[7] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 8, no. 6, pp. 1259–1284, 1998.

[8] W. Zhang, K.-W. Wong, H. Yu, and Z.-L. Zhu, "A symmetric color image encryption algorithm using the intrinsic features of bit distributions," *Communications in Nonlinear Science & Numerical Simulation*, vol. 18, no. 3, pp. 584–600, 2013.

[9] Z. Zhu, W. Zhang, K. Wong, and H. Yu, "A chaos-based symmetric image encryption scheme using a bit-level permutation," *Information Sciences*, vol. 181, no. 6, pp. 1171–1186, 2011.

[10] L. Teng and X. Wang, "A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive," *Optics Communications*, vol. 285, no. 20, pp. 4048–4054, 2012.

[11] E. Vaferi and R. Sabbaghi-Nadooshan, "A new encryption algorithm for color images based on total chaotic shuffling scheme," *Optik*, vol. 126, no. 20, pp. 2474–2480, 2015.

[12] S. M. Wadi and N. Zainal, "Decomposition by binary codes-based speedy image encryption algorithm for multiple applications," *IET Image Processing*, vol. 9, no. 5, pp. 413–423, 2015.

[13] X. Wang and H.-L. Zhang, "A color image encryption with heterogeneous bit-permutation and correlated chaos," *Optics Communications*, vol. 342, pp. 51–60, 2015.

[14] Y.-Q. Zhang and X.-Y. Wang, "A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice," *Information Sciences*, vol. 273, pp. 329–351, 2014.

[15] H. Liu and X. Wang, "Color image encryption using spatial bit-level permutation and high-dimension chaotic system," *Optics Communications*, vol. 284, no. 16-17, pp. 3895–3903, 2011.

[16] Y. Zhou, W. Cao, and C. L. P. Chen, "Image encryption using binary bitplane," *Signal Processing*, vol. 100, no. 7, pp. 197–207, 2014.

[17] A.-V. Diaconu, "Circular inter-intra pixels bit-level permutation and chaos-based image encryption," *Information Sciences*, vol. 355-356, pp. 314–327, 2016.

[18] D. Arroyo, J. Diaz, and F. B. Rodriguez, "Cryptanalysis of a one round chaos-based Substitution Permutation Network," *Signal Processing*, vol. 93, no. 5, pp. 1358–1364, 2013.

[19] Y. Zhang and D. Xiao, "Cryptanalysis of S-box-only chaotic image ciphers against chosen plaintext attack," *Nonlinear Dynamics*, vol. 72, no. 4, pp. 751–756, 2013.

[20] C.-X. Zhu, Y.-P. Hu, and K.-H. Sun, "New image encryption algorithm based on hyperchaotic system and ciphertext diffusion in crisscross pattern," *Journal of Electronics and Information Technology*, vol. 34, no. 7, pp. 1735–1743, 2012.

[21] X. Wang, L. Teng, and X. Qin, "A novel colour image encryption algorithm based on chaos," *Signal Processing*, vol. 92, no. 4, pp. 1101–1108, 2012.

[22] X. Wang, L. Liu, and Y. Zhang, "A novel chaotic block image encryption algorithm based on dynamic random growth technique," *Optics and Lasers in Engineering*, vol. 66, no. 66, pp. 10–18, 2015.

[23] Z. Hua, Y. Zhou, C.-M. Pun, and C. L. P. Chen, "2D Sine Logistic modulation map for image encryption," *Information Sciences*, vol. 297, pp. 80–94, 2015.

[24] A.-C. Dăscălescu, R. E. Boriga, and A.-V. Diaconu, "Study of a new chaotic dynamical system and its usage in a novel pseudorandom bit generator," *Mathematical Problems in Engineering*, vol. 2013, Article ID 769108, 2013.

[25] USC-SIPI Image Database and University of South California, "Signal and Image Processing Institute," 2016, http://sipi.usc.edu/database.