An Image Encryption Algorithm Based on Random Walk and Hyperchaotic Systems

Cong Xu*, Jingru Sun† and Chunhua Wang‡
College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, P. R. China
∗xucong@163.com
†sunjr@hnu.edu.cn
‡wch1227164@hnu.edu.cn

Received April 26, 2019; Revised July 5, 2019

In this paper, we propose an image encryption algorithm based on random walk and two hyperchaotic systems. The random walk method is adopted to scramble the position of pixels within a block. Furthermore, the permutation operation between blocks is presented to enhance the scramble effect. Thus, high correlation among pixels of original image is broken by permutation. Moreover, the chosen plaintext attack is used to test the anti-attack ability of the proposed algorithm. By analyzing experimental results and comparing with other image encryption algorithms, we show that the proposed algorithm has better performance and higher security.

Keywords: Chaos; image encryption; random walk; hyperchaos; block image.

1. Introduction

Digital image is a popular multimedia format, and image encryption is a hot research topic in image processing and information security. Image has some inherent features, such as high correlation among pixels and huge size of data, thus traditional encryption techniques cannot meet the demand of image encryption [Li et al., 2007]. Due to the characteristics of randomness, uncertainty, and initial conditions’ sensitivity, chaotic system has good prospects in the field of image encryption.

Since the chaos-based encryption scheme was first introduced by Matthews [1988], a great deal of chaos-based encryption methods have been put forward. There are two main recent design ideas in view of the encryption schemes based on chaotic system, one is to enhance the security of key stream by utilizing new types of chaotic systems or hyperchaotic systems, the other idea is to design more complex algorithms by new technologies, such as bit-level permutation, DNA computing, compression sensing, cellular automata.

For the first design idea, some researchers have proposed image encryption schemes based on new chaotic systems instead of existing chaotic systems/maps. By integrating two existing one-dimensional chaotic maps, a novel simple chaotic system was introduced for image encryption [Zhou et al., 2014]. After comparing the performance of real chaotic system with that of complex chaotic system, it is found that the latter is more suitable for image encryption, thus a high security color encryption scheme based on two complex chaotic systems was introduced for image encryption [Zhou et al., 2017a]. A new compounded piecewise-linear map for encryption by considering both safety and computing expense, the scheme has outstanding cryptography features and good efficiency. In addition, with the characteristics of better unpredictability, more complex structure and larger key

*Author for correspondence
C. Xu et al.

For the second design idea, compared with pixel-level image encryption schemes, bit-level encryption schemes can affect both the pixel value and the pixel position in the permutation process, so more and more scholars have introduced bit-level encryption schemes. Wang et al. [2014] proposed a bit-level encryption scheme by using a perceptron model. In [Xu et al. 2016], the cryptographical algorithm was constructed by using chaotic maps, making the bits swap to arbitrary bit planes. In [Zhou & Wang 2020], hyperchaotic system has many applications in image encryption. Zhou et al. [2014] Norouzi et al. [2014] Li et al. [2015] Yin & Wang [2016].

For the second design idea, compared with pixel-level image encryption schemes, bit-level encryption schemes can affect both the pixel value and the pixel position in the permutation process, so more and more scholars have introduced bit-level encryption schemes. Wang et al. [2014] proposed a bit-level encryption scheme by using a perceptron model. In [Xu et al. 2016], the cryptographical algorithm was constructed by using chaotic maps, making the bits swap to arbitrary bit planes. In [Zhou & Wang 2020], hyperchaotic system has many applications in image encryption. Zhou et al. [2014] Norouzi et al. [2014] Li et al. [2015] Yin & Wang [2016].

Generally, for the second design idea, namely, designing more complex encryption algorithm, some algorithms are based on mathematical transformations, and some others are based on the rules designed by authors. Those transformations and rules are deterministic, which lack randomness and unpredictability. To solve the problem above, we propose a new encryption scheme based on random walk and hyperchaos. The plaintext is divided into blocks of fixed size first, and the blocks do not overlap. Then all pixels within each block image are scrambled by random walk matrix which records the path of the random walk. In the generation process of random walk matrix, the starting position and direction of movement are controlled by the random numbers which are obtained by the chaotic system, moreover, the number of movement times is unpredictable, which is affected by the starting position and direction of movement. Furthermore, permutation method between blocks is applied to scramble the positions of pixels effectively. The correlation of neighborhood pixels within blocks and that of adjacent pixels between blocks can be both broken by the permutation methods in our paper. We also present the diffusion method to change the value of pixels. To strengthen the security of the scheme, the same chaotic sequence is not reused in the permutation and diffusion process. In addition, if we use the fixed secret keys when encrypting different images, the attackers will have opportunities to obtain the secret keys by using plaintext attacks, and then use the secret keys to decrypt other cipher images. Wang et al. [2018a] Chen et al. [2018].

2. Basic Theories

2.1. Random walk matrix

The term “random walk” was introduced by Pearson [1905]. Random walks are popular in the
An Image Encryption Algorithm Based on Random Walk and Hyperchaotic Systems

sciences, and they are interesting from both theoretical and practical perspectives. There is a mathematical problem which is called a random walk problem, it can be described as: there is a man in a room, and the room size is $m \times n$, he stands in the designated starting position, then he random moves to one of the eight directions around him continuously. The process of random walk is not completed until all the positions of the room have been walked.

We simulate the process of random walk in a blank matrix, then the random walk path can be recorded in the matrix which we call random walk matrix (RWM), according to the rules as follows: each position records the number of times he walked in. It is clear that the process of random walk involves three parameters, they are the size of the matrix, the starting position and the direction of each movement. In this paper, we set the size of the matrix to be the same as the sub-block image, the starting position and direction of each movement are given by random numbers which are generated by the chaotic system.

2.2. Chaotic systems

Chaotic systems are adopted to obtain key sequences in our scheme, the hyperchaotic Lorenz system [Wang & Wang, 2008] and Chen’s hyperchaotic system [Gao et al., 2006] are represented by Eqs. (1) and (2), respectively,

$$\begin{align*}
\dot{x} &= \alpha (y - z) + w, \\
\dot{y} &= \gamma x - y - xz, \\
\dot{z} &= xy - \beta z, \\
\dot{w} &= -yz + \eta w,
\end{align*}$$

(1)

the system displays chaotic behavior with fixed or changed parameters ($\alpha = 10, \beta = 8/3, \gamma = 28$ and $-1.52 < \eta \leq -0.06$), and we choose $\eta = -1$ in our experiment

$$\begin{align*}
\dot{x} &= \alpha (y - x), \\
\dot{y} &= -xy + nx + my - w, \\
\dot{z} &= xy - bz, \\
\dot{w} &= x + k,
\end{align*}$$

(2)

with $a = 36, b = 3, m = 28, n = 16$ and $k \in [-0.7, 0.7]$, the system has rich dynamics of hyperchaos, and $k$ is set as 0.2 in our experiments.

3. The Process of Image Encryption and Decryption

We assume the size of the original image $P$ is $M \times N$, and that of the random walk matrix is $m \times n$. Figure 1 gives the flow diagram of our proposed encryption scheme.

3.1. The generation of initial keys

It is obvious that hash values of two images with only 1-bit difference have obvious distinctions, and

![Diagram of the scheme](image.png)

Fig. 1. The flow diagram of the scheme.
the initial keys related to MD5 hash values of plain image will enhance the sensitivity of the scheme. The 128-bit hash value \( K \) is divided into 16 blocks, and each block includes 8-bits, denoted as follows:

\[
K = k_1, k_2, k_3, \ldots, k_{16}. \tag{3}
\]

According to the following computing method, four initial keys of hyperchaotic Lorenz system can be obtained.

\[
\begin{align*}
x'_0 &= x_0 + \text{abs} \left( \frac{k_1 + k_3 + k_4 + k_5}{256} \right) \\
y'_0 &= y_0 + \text{abs} \left( \frac{k_1 + k_3 + k_4 + k_5}{256} \right) \\
z'_0 &= z_0 + \text{abs} \left( \frac{k_1 + k_3 + k_4 + k_5}{256} \right) \\
w'_0 &= w_0 + \text{abs} \left( \frac{k_1 + k_3 + k_4 + k_5}{256} \right)
\end{align*}
\tag{4}
\]

where \( x_0, y_0, z_0, w_0 \) are the initial given values, \( \text{abs}(a) \) to obtain the absolute value of \( a \), \( \text{fix}(a) \) takes the integer part of \( a \). For another chaotic system as discussed in Sec. 2.1, the starting location is \((p_x, p_y)\) and the direction of each movement is controlled by \( X(d) \), where \( d \) is a decimal number of movement times. Then the random walk matrix \( A_t \) is obtained. For example, the starting position in a blank matrix is \((4 \mod(1500), 1), 17 \mod(1500), 1)\), the number of times he walked in this position.

\[
\begin{align*}
x'_{10} &= x_{10} + \frac{k_3 \oplus k_5 \oplus k_7 \oplus k_9}{256} \\
y'_{10} &= y_{10} + \frac{k_3 \oplus k_5 \oplus k_7 \oplus k_9}{256} \\
z'_{10} &= z_{10} + \frac{k_3 \oplus k_5 \oplus k_7 \oplus k_9}{256} \\
w'_{10} &= w_{10} + \frac{k_3 \oplus k_5 \oplus k_7 \oplus k_9}{256}
\end{align*}
\tag{5}
\]

where \( x_{10}, y_{10}, z_{10}, w_{10} \) are the initial given values, \( a \oplus b \) indicates exclusive OR operation.

### 3.2. The generation of random walk matrix

In the paper, we simulate the random walk in a blank matrix controlled by the chaotic system, then generate random walk matrices (RWM) for permutation.

**Step 1.** Use the initial keys \( x'_0, y'_0, z'_0, w'_0 \) produced in Sec. 3.1 to produce four pseudo-random sequences \( x, y, z \) and \( w \) by iterating the hyperchaotic Lorenz system \( N_0 + 8L \) times, where \( L = M \times N \). To eliminate transient effect, the former \( N_0 \) values of each sequence are discarded.

\[
N_0 = \text{mod}(H, 1500), \tag{6}
\]

where \( H \) is the decimal form of the MD5 hash value, \( \text{mod}(a, b) \) denotes modulo operation.

**Step 2.** The starting position \((p_x, p_y)\) of random walk is computed by Eq. (5),

\[
\begin{align*}
p_x(i) &= \text{mod}(\text{floor}(y(i) \times 10^{13}), m) + 1, \\
p_y(i) &= \text{mod}(\text{floor}(z(i) \times 10^{13}), n) + 1,
\end{align*}
\tag{7}
\]

where \( \text{floor}(a) \) finds the elements of \( a \) to the largest integer that is smaller than \( a \), and \( i = 1, 2, 3, \ldots, L/(m \times n) \).

**Step 3.** The direction of random walk is controlled by \( X \), and \( X \) is computed by Eq. (6).

\[
X(i) = \text{mod}(\text{floor}(x(i) \times 10^{13}), 8), \tag{8}
\]

**Step 4.** Set \( i = 1 \).

**Step 5.** Perform random walk on a \( m \times n \) blank matrix \( A_t \) as discussed in Sec. 2.1, the starting location is \((p_x(i), p_y(i))\) and the direction of each movement is controlled by \( X(d) \), where \( d = 1, 2, 3, \ldots, t \), and \( t \) is the number of movement times. Then the random walk matrix \( A_t \) is obtained. For example, the starting position in a blank matrix is \((4, 1)\) as shown in Fig. 2(a), and when the random walk is completed, we obtained the RWM \( A_t \) as shown in Fig. 2(b), each position of \( A_t \) records the number of times he walked in this position.

**Step 6.** Perform Step 5 repeatedly by setting \( i = i + 1 \). Then random walk matrices \( A_i \) \((i = 1, 2, 3, \ldots, L/(m \times n)) \) are obtained.

### 3.3. Permutation

Considering the increasing size of the image, we manipulate scramble operation by block. Our
The permutation process contains the permutation within the block and the permutation between the blocks.

Firstly, the \((M \times N)/(m \times n)\) sub-blocks are obtained by segmenting images into the block, we set \(B_i\) indicating the \(i\)th block, here, numbering blocks according to top-to-bottom and left-to-right, and the size of each sub-block is \(m \times n\). In general, \(M\) should be an integral multiple of \(m\), \(N\) should be an integral multiple of \(n\), if it is not satisfied, the image is padded with black color, so the algorithm has no limit for image size.

### 3.3.1. Permutation within blocks based on random walk matrix

In this phase, we scrambled positions of pixels within each sub-block image based on RWM. The permutation operation within blocks is described by a pseudo-code in Table 1.

Table 1. Pseudo-code of the permutation within blocks.

<table>
<thead>
<tr>
<th>Input: Random walk matrix (A), sub-block (B)</th>
<th>Output: Permuted sub-blocks (B')</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Convert (A), (B) into one-dimension sequence</td>
<td></td>
</tr>
<tr>
<td>2. for (i = 1: (M \times N)/(m \times n))</td>
<td></td>
</tr>
<tr>
<td>3. index sequence (i \rightarrow A_i) is sorted in ascending order</td>
<td></td>
</tr>
<tr>
<td>4. for (j = 1: (m \times n))</td>
<td></td>
</tr>
<tr>
<td>5. (B'_i(j) = B_i(l(j)))</td>
<td></td>
</tr>
<tr>
<td>6. end</td>
<td></td>
</tr>
<tr>
<td>7. end</td>
<td></td>
</tr>
</tbody>
</table>

After it. The permutation operation between blocks is described by pseudo-code in Table 2.

### 3.3.2. Permutation between blocks

After the permutation based on RWM is employed to each sub-block, the permutation method between block images is employed, and pixels of block image \(B'_i\) \((i = 1, 2, 3, \ldots, (M \times N)/(m \times n))\) will be swapped with pixels of other block images located

\[
X_i(i) = \text{mod}(\text{floor}((z_1(i) + y_1(i)) \times 10^{14}), 256),
\]

\[
Z_i(i) = \text{mod}(\text{floor}((z_1(i) + w_1(i)) \times 10^{14}), 256).
\]

"Fig. 2. An instance of random walk: (a) An 8 × 8 blank matrix, and the starting location is (4, 5) and (b) random walk matrix is obtained.

Table 2. Pseudo-code of the diffusion operation between blocks.

<table>
<thead>
<tr>
<th>Input: Permuted sub-blocks (B')</th>
<th>Output: UITextViewFeatures (F')</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Convolve (\tilde{B}') with (H)</td>
<td></td>
</tr>
<tr>
<td>2. for (i = 1: (M \times N)/(m \times n))</td>
<td></td>
</tr>
<tr>
<td>3. (F'_{i}(i) = \text{mod}(\text{floor}((z_1(i) + y_1(i)) \times 10^{14}), 256),)</td>
<td></td>
</tr>
<tr>
<td>4. for (j = 1: (m \times n))</td>
<td></td>
</tr>
<tr>
<td>5. (F'_{i}(j) = \text{mod}(\text{floor}((z_1(j) + w_1(j)) \times 10^{14}), 256),)</td>
<td></td>
</tr>
<tr>
<td>6. end</td>
<td></td>
</tr>
<tr>
<td>7. end</td>
<td></td>
</tr>
</tbody>
</table>

In (9) and (10), diffusion operation is described by pseudo-code in Table 2.

### 3.4. Diffusion

We utilized a key stream strongly related to plain image in the diffusion process for a good diffusion effect. Iterating Chen’s hyperchaotic system \(N_0 + MN\) times, we obtained the sequence \(x_1, y_1, z_1\) and \(w_1\), where the initial keys of the system are as in Sec. 3.1. The former \(N_0\) values of each sequence are discarded to eliminate the transient effect. Then we can obtain the key stream \(X_1, Z_1\) from Eqs. (9) and (10), and the diffusion operation is described by pseudo-code in Table 2.
random walk matrix.

4. Results of Simulation

3.5. The process of decryption

Decryption is the converse of encryption.

4. Results of Simulation

Plain images for testing are Lena (512 × 512), Butterfly (512 × 768), Terrace (1200 × 1280) and Bridge (color 256 × 256). In the encryption, the color image is divided into R, G and B components, then the three components are encrypted separately. The secret keys are \((x_0 = 3.3133, y_0 = 12.0546, z_0 = 40.8897, w_0 = -34.5677, x_10 = 0.3838, y_10 = 0.9876, z_{10} = 32.1234, w_{10} = 0.6565)\).

Figure 4 exhibits the results of encryption and decryption by using the proposed scheme. Obviously, all plain images and decrypted images are identical. Additionally, images of various sizes can be effectively encrypted by the proposed algorithm, that is, the encryption scheme has no limit for image size.

5. Security Analysis

In all of these simulations, testing images are the 512 × 512 pixels with 8-bit gray level. The security analysis is shown as below.

5.1. Analysis of key space

The secret keys of our scheme contain the given initial keys and hash values of plain image. The initial values are double-precision, whose key space is \((10^{16})^3 = 10^{48} \approx 2^{160}\). The key space of 128-bit hash value is \(2^{384}\). The total key space is \(2^{384} \times 2^{48} = 2^{432}\). For a secure scheme, the key space should be larger than \(2^{100}\) [Alvarez & Li 2006]. Clearly, it is not feasible for attackers to break the proposed scheme by using brute-force attacks.

5.2. Analysis of key sensitivity

The original keys are set as \((x_0 = 3.3133, y_0 = 12.0546, z_0 = 40.8897, w_0 = -34.5677, x_{10} = 0.3838, y_{10} = 0.9876, z_{10} = 32.1234, w_{10} = 0.6565)\).
An Image Encryption Algorithm Based on Random Walk and Hyperchaotic Systems

Fig. 4. Results of encryption and decryption: From the first to the third column are plain images, cipher images and decrypted images.
C. Xu et al.

![Image](image_url)

Fig. 5. Test of key sensitivity for encryption: (a) Plain image, (b) ciphertext with original keys, (c) ciphertext with modified keys and (d) difference between the two ciphertexts.

\[y_{10} = 0.9876, \ z_{10} = 32.1234, \ w_{10} = 0.6565, \]

the modified keys are set as \((x_0 = 3.3133, \ y_0 = 12.0546, \ z_0 = 40.8897, \ w_0 = -34.5677, \ x_{10} = 0.3638 + 10^{-14}, \ y_{10} = 0.9876, \ z_{10} = 32.1234, \ w_{10} = 0.6565)\).

In the process of encryption, the Lena image is encrypted by the original keys and modified keys, respectively. As Fig. 5 shows, ciphertext with the original keys is shown in Fig. 5(b), then we use the modified keys to encrypt the plain image again and obtain another ciphertext. By observing Fig. 5(d), it is clear that a slight modification in keys makes a tremendous difference in ciphertext. In the process of decryption, the ciphertext of Lena is decrypted by the original keys and modified keys, respectively. As Fig. 5 shows, the original keys restored the plain image successfully, but the plain image cannot be recovered correctly by using the slightly modified keys.

5.3. Analysis of statistical attacks

5.3.1. Analysis of histogram

Histogram is used to show the number of grayscale values with certain values. The cipher image with a uniform histogram can be obtained by a high security image encryption algorithm. The histograms of Lena image are shown in Fig. 6. In contrast with the histogram of plain image, the ciphertext has uniform distribution of pixel, which means that our scheme could resist the statistical attack.
5.3.2 Analysis of correlation coefficient

An image correlation coefficient means a statistical relationship between neighborhood pixels. High correlation between neighborhood pixels of plain image should be broken by an ideal encryption scheme. From the plain image and the corresponding cipher image, we randomly choose a number of pixels to calculate the correlation coefficients in three different directions by Eqs. (11)–(14).

\[ \gamma_{xy} = \frac{\text{cov}(x, y)}{\sqrt{D(x)D(y)}} \]  
(11) 

\[ E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i \]  
(12) 

\[ D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2 \]  
(13) 

\[ \text{cov}(x, y) = E(x - E(x))(y - E(y)) \]  
(14)

The gray values of neighborhood pixels are \( x \) and \( y \), respectively. \( N \) is the total number of pixels chosen. The correlation distribution in three directions are displayed in Fig. 6. By comparing the correlation coefficients between neighborhood pixels of plain image and that of cipher image (as listed in Table 4), we could get a conclusion that cipher images have a low correlation between neighborhood pixels, thus, our proposed scheme has the good ability to defend the statistical analysis attacks.
5.3.3. **Analysis of information entropy**

In information theory, the information entropy is used to measure information quantity, and the entropy can be calculated by Eq. (15)

\[
H(m) = \sum_{i=0}^{2^N-1} p(x_i) \log_2 \frac{1}{p(x_i)},
\]

where the probability of symbol \(x_i\)'s presence is denoted as \(p(x_i)\), and the number of bits per pixel is \(N\). Table 2 lists the entropy values of test images, and the calculation results are approaching the ideal theoretical value 8, this indicates the proposed algorithm can realize randomness for cipher images by encrypting plain images.

5.4. **Analysis of differential attacks**

Differential attackers break cryptographic algorithms by comparing and analyzing the corresponding cipher images of two plain images with a tiny difference. NPCR and UACI [Mao et al. 2004] are two criteria to measure the capability of resistance to differential attacks.

\[
\text{NPCR} = \frac{1}{M \times N} \sum_{x=1}^{M} \sum_{y=1}^{N} D(x, y) \times 100\%,
\]

\[
\text{UACI} = \frac{1}{M \times N} \left[ \sum_{x=1}^{M} \sum_{y=1}^{N} \left| C_1(i, j) - C_2(i, j) \right| \right] \times 100\%,
\]

\[
D(x, y) = \begin{cases} 
0, & C_1(x, y) = C_2(x, y), \\
1, & C_1(x, y) \neq C_2(x, y),
\end{cases}
\]

where pixel values of cipher images are denoted as \(C_1(x, y)\) and \(C_2(x, y)\), and cipher images are corresponding to the two original images with only 1-bit difference.
Fig. 8. The correlation distribution of (a) Lena and (e) ciphered image. (b)–(d) and (f)–(h) Correlation distribution in horizontal, vertical and diagonal direction, respectively.
Table 4. Correlation coefficients.

<table>
<thead>
<tr>
<th>Image</th>
<th>Horizontal Plain</th>
<th>Vertical Plain</th>
<th>Diagonal Plain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>0.9852</td>
<td>-0.0122</td>
<td>0.9583</td>
</tr>
<tr>
<td>Boat</td>
<td>0.9669</td>
<td>0.0111</td>
<td>-0.0237</td>
</tr>
<tr>
<td>Baboon</td>
<td>0.7045</td>
<td>0.0237</td>
<td>0.7345</td>
</tr>
<tr>
<td>Girl</td>
<td>0.9873</td>
<td>-0.0062</td>
<td>0.9749</td>
</tr>
<tr>
<td>Pepper</td>
<td>0.9815</td>
<td>-0.0010</td>
<td>0.9637</td>
</tr>
</tbody>
</table>

Table 5. The entropies of cipher images.

<table>
<thead>
<tr>
<th>Image</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>7.9993</td>
</tr>
<tr>
<td>Boat</td>
<td>7.9992</td>
</tr>
<tr>
<td>Baboon</td>
<td>7.9993</td>
</tr>
<tr>
<td>Girl</td>
<td>7.9995</td>
</tr>
<tr>
<td>Pepper</td>
<td>7.9993</td>
</tr>
</tbody>
</table>

The results in Table 5 show that the values of NPCR and UACI are approaching the ideal theoretical values 99.61% and 33.46% only through one round of encryption. In other words, our scheme can withstand the differential attack effectively.

5.5. Analysis of some typical attacks

As well known, the anti-attack ability is an important criterion for evaluating the safety of encryption algorithms, and chosen plaintext attack is the most effective attack in modern cryptography, so the ability of anti-chosen plaintext attack needs to be improved when designing encryption scheme [Chai et al., 2019]. In our scheme, the initial keys are related to MD5 hash value of the original image. Therefore, the different key streams will be used.

Fig. 9. Experimental results of special images: (a) and (d) Special images, (b) and (e) encrypted images and (c) and (f) histogram of encrypted images.
Table 7. The encryption results of special images.

<table>
<thead>
<tr>
<th></th>
<th>Entropy</th>
<th>Horizontal</th>
<th>Vertical</th>
<th>Diagonal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cipher image for</td>
<td>7.9993</td>
<td>−0.0021</td>
<td>0.0044</td>
<td>−0.0069</td>
</tr>
<tr>
<td>all black image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cipher image for</td>
<td>7.9992</td>
<td>−0.0200</td>
<td>0.0131</td>
<td>0.0031</td>
</tr>
<tr>
<td>all white image</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When encrypting different images. Hence, the key stream retrieved with one chosen-plain image cannot decrypt other cipher images correctly, which illustrates that chosen plaintext attacks are not valid.

5.5.1. Experimental results’ analysis of special plain images

Some cryptanalysts try to attack an encryption algorithm by using all black or white images as plain image. We select special images as input images, and the size of images for testing are 512 × 512. In Fig. 9 and Table 7, the histograms with uniform distribution show that attackers cannot obtain available information by performing the encryption scheme with special images.

5.5.2. Attack test

Some image encryptions have good performance analysis’ results, but they are easily attacked by chosen plaintext [Wang et al., 2012; Pak & Huang, 2015; Zhen et al., 2014; Song et al., 2014; Ye, 2010]. In order to verify the anti-chosen plaintext attack ability of the proposed encryption scheme, we adopted chosen plaintext attack to test the proposed algorithm and Yin’s algorithm [Yin & Wang, 2018]. As shown in Fig. 10, the image of Lena is encrypted by using our algorithm and Yin’s algorithm, respectively. The recovered images of cipher images are obtained after using chosen plaintext

![Fig. 10. Attack test: (a) and (d) Lena image (512 x 512), (b) and (e) cipher image of Lena using our proposed algorithm and Yin's algorithm, respectively and (c) and (f) recovered images of (b) and (e), respectively.](image-url)
Table 8. The encryption time comparison results.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Images</th>
<th>Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>Lena (256 × 256)</td>
<td>0.132</td>
</tr>
<tr>
<td></td>
<td>Butterfly (512 × 768)</td>
<td>0.592</td>
</tr>
<tr>
<td></td>
<td>Terrace (1200 × 1280)</td>
<td>0.943</td>
</tr>
<tr>
<td>Yin’s [Yin &amp; Wang, 2018]</td>
<td>Lena (256 × 256)</td>
<td>0.417</td>
</tr>
<tr>
<td>Zhu’s [Zhu et al., 2017]</td>
<td>Lena (256 × 256)</td>
<td>1.532</td>
</tr>
<tr>
<td>Chai’s [Chai et al., 2018]</td>
<td>Lena (256 × 256)</td>
<td>0.580</td>
</tr>
</tbody>
</table>

Table 9. Results of comparison.

<table>
<thead>
<tr>
<th>Comparing Parameter</th>
<th>Ours</th>
<th>Li’s [Li et al., 2018]</th>
<th>Wang’s [Wang et al., 2018b]</th>
<th>Sukalyan’s [Sukalyan et al., 2019]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack analysis</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Key space</td>
<td>$2^{48}$</td>
<td>$2^{32}$</td>
<td>$4.5 \times 10^{114}$</td>
<td>$16^{45}$</td>
</tr>
<tr>
<td>Entropy</td>
<td>7.9993</td>
<td>7.9993</td>
<td>7.9971</td>
<td>7.9965</td>
</tr>
<tr>
<td>NPCR</td>
<td>0.9963</td>
<td>0.9963</td>
<td>0.9909</td>
<td>0.9907</td>
</tr>
<tr>
<td>UACI</td>
<td>0.3347</td>
<td>0.3348</td>
<td>0.3345</td>
<td>0.3218</td>
</tr>
<tr>
<td>Execution time(s)</td>
<td>0.425</td>
<td>0.926</td>
<td>1.243</td>
<td>0.494</td>
</tr>
</tbody>
</table>

In contrast to the encryption schemes in [Wang et al., 2018b; Sukalyan et al., 2019], the information entropy is more close to the ideal value. The NPCR and UACI values in this paper are also pretty close to the ideal values. In addition, the attack analysis illustrated the proposed scheme has the ability of anti-chosen plaintext attack. Moreover, execution time of the proposed scheme is faster than the schemes in [Li et al., 2018; Wang et al., 2018b; Sukalyan et al., 2019]. Therefore, the proposed scheme is feasible and effective.

7. Conclusion

An encryption scheme based on random walk and two hyperchaotic systems is proposed. It is divided into three phases: permutation within block, permutation between blocks and diffusion. In the proposed permutation within block, pixels of each block image are scrambled based on random walk matrix. The permutation method can effectively break the high correlation between neighborhood pixels in plain image. Additionally, the initial keys are related to MD5 hash value of original image. We carry out many experimental tests and analyses, the results indicate that our encryption scheme provides high security.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61971185), and
the Open Fund Project of Key Laboratory in Hunan Universities (No. 18K010).

References


