Chaotic image encryption scheme using two-by-two deoxyribonucleic acid complementary rules

Shuliang Sun

Chaotic image encryption scheme using two-by-two deoxyribonucleic acid complementary rules

Shuliang Sun\textsuperscript{a,b,*}
\textsuperscript{a}Fuqing Branch of Fujian Normal University, School of Electronics and Information Engineering, Fuqing, China
\textsuperscript{b}Fuqing Branch of Fujian Normal University, Innovative Information Industry Research Center, Fuqing, China

Abstract. An image encryption technique has been proposed using deoxyribonucleic acid (DNA) operations and chaotic map in this scheme. First, initial conditions of row encryption and column encryption are calculated. Then, a two-dimensional sine iterative chaotic map with infinite collapse (ICMIC) modulation map (2D-SIMM) is adopted to produce chaotic sequences. Extended exclusive OR (XOR) is executed to enhance security. A mask matrix is produced by 2D-SIMM. It performs XOR operation with the DNA-encoded matrix. Finally, the revised DNA-encoded matrix is performed two-by-two DNA complementary rules and executed DNA decoding to obtain the cipher image. Experiment results prove that the proposed scheme is secure enough and can resist various attacks. © The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: 10.1117/1.OE.56.11.116117]

Keywords: two-by-two deoxyribonucleic acid complementary rule; expanded exclusive OR; two-dimensional sine ICMIC modulation map.

Paper 171145 received Jul. 21, 2017; accepted for publication Nov. 13, 2017; published online Nov. 30, 2017.

1 Introduction

Recently, many image encryption methods have been proposed. The diffusion and confusion operations that are proposed by Shannon\textsuperscript{1} in cryptography are also used in image encryption. Traditional encryption methods are improper for image encryption such as low efficiency, enormous data, high correlation, and so on.\textsuperscript{2} Chen et al.\textsuperscript{3} presented a real-time cryptosystem. The confusion and diffusion operations of this scheme were performed based on a lookup table. Chen et al.\textsuperscript{4} also put forward an image encryption algorithm based on gray code. It was performed with high efficiency. The chaotic map is highly sensitive to initial values and system parameters, unpredictable, pseudorandom, and ergodic.\textsuperscript{5} It is very suitable for an image encryption system. Liu et al.\textsuperscript{6} proposed a double image encryption method based on random pixel exchanging and phase encoding in gyrator domains. Mao et al.\textsuperscript{7} presented a fast image encryption method, which was based on three-dimensional chaotic baker maps. Sivakumar and Venkatases\textsuperscript{8} proposed an image encryption scheme. Knight’s travel path and true random number were adopted in this method. Wang et al.\textsuperscript{9} presented an efficient image encryption scheme using a two-step phase-shifting interference method, in fractional Fourier transform and random mixed encoding. Wang and Luan\textsuperscript{10} proposed an image encryption scheme using reversible cellular automata and chaos.

Since deoxyribonucleic acid (DNA) computing supports high parallelism, it is also superior in massive storage and extremely low power consumption. Many DNA-based methods have been proposed nowadays.\textsuperscript{11–15} Zhen et al.\textsuperscript{12} presented an image encryption scheme based on chaotic sequence, DNA encoding, and entropy. Rehman et al.\textsuperscript{13} proposed a method that was based on chaos and DNA complementary rules for gray images. The most significant and least significant parts of each block were encoded with different methods. Wang et al.\textsuperscript{16} proposed an image encryption technique based on DNA sequence and coupled map lattice. The scheme could resist different attacks and enhance the system’s security. Wang et al.\textsuperscript{17} also designed an image encryption method based on a two-dimensional (2-D) logistic map and DNA sequence. DNA addition, DNA subtraction, and DNA complementary rules were used to obtain the ciphered image. Belazi et al.\textsuperscript{18} designed an equivalent mathematical model of the cryptosystem and algebraic analysis was given. By finding equivalent keys, key space was reduced. The authors also proposed a recovering scheme with lower complexity than the actual decryption method. In Ref. 19, a 2-D logistic map was employed for row circular permutation and column circular permutation. Initial values and system parameters of the chaotic system were calculated first. Abd-El-Hafiz et al.\textsuperscript{20} proposed two measures for the evaluation permutation skills. Two parameters were proposed in the program.

The rest of this paper is arranged as follows. Section 2 briefly introduces the two-dimensional sine iterative chaotic map with infinite collapse (ICMIC) modulation map (2D-SIMM), random number generation, DNA operations, and improved expanded exclusive OR (XOR) operation. Section 3 describes the proposed scheme. Section 4 depicts the simulation results. Section 5 presents security analysis and the conclusion is described in Sec. 6.

2 Preliminary Work

2.1 Two-Dimensional Sine ICMIC Modulation Map

2D-SIMM\textsuperscript{21} is defined as

\begin{equation}
\begin{align*}
x_{i+1} &= a \sin(\pi y_i) \sin(b/x_i) \\
y_{i+1} &= a \sin(\pi x_{i+1}) \sin(b/y_i)
\end{align*}
\end{equation}

where \(a\) and \(b\) are the positive system parameters. While \(a = 1\) and \(b = 5\), the system of 2D-SIMM is a hyper chaotic map. Compared with the 2-D sine logistic modulation
map (2D-SLMM)\textsuperscript{22} and 2-D logistic map.\textsuperscript{17} 2D-SIMM has better ergodicity, larger key space, more complex dynamical behaviors, and phase space trajectory.\textsuperscript{21} More secure chaotic sequences could be produced with the system. 2D-SIMM is employed in this paper to generate chaotic sequences.

### 2.2 Random Number Generation

The sequences produced by 2D-SIMM are decimal real numbers. The fractional part of a real number is changed into equivalent binary format. The former 16 bits of binary value are separated into two halves each with 8 bits.\textsuperscript{8} They are executed XOR operation to generate a random number finally.

The process can be described as follows. Suppose the value generated by 2D-SIMM is 0.63636. The binary bit streams of the fractional part are 1010001011101000. The first half is (10100010), and the other half is (11101000).2 The generated random number \((R) = T_{15-8}\) XOR \(T_{7-0} = (01001010)\).

### 2.3 Deoxyribonucleic Acid Operations

DNA is two twisted strands, which are composed of four bases: adenine (A), cytosine (C), thymine (T), and guanine (G).\textsuperscript{13} (A) and (G), respectively, bond with complement (T) and (C), and vice versa. 00, 01, 10, and 11 are represented as A, C, G, and T, respectively. Each pixel is transformed into a DNA sequence with length 4 in 8-bit gray image. The rule for DNA XOR is shown in Table 1.

The DNA complementary rule must satisfy that\textsuperscript{17}

\[
\begin{align*}
\{ x \neq D(x) \neq D[D(x)] \neq D[D[D(x)]] \} \quad & (2) \\
\{ x = D[D(x)] = D[D[D(x)]] \} \\
\end{align*}
\]

where \(D(x)\) is different from \(x\) in at least 1-bit position.

There are six DNA complementary rules as follows:

1. G \(\rightarrow\) T, T \(\rightarrow\) A, A \(\rightarrow\) C, C \(\rightarrow\) G;
2. G \(\rightarrow\) T, T \(\rightarrow\) C, C \(\rightarrow\) A, A \(\rightarrow\) G;
3. G \(\rightarrow\) C, C \(\rightarrow\) A, A \(\rightarrow\) T, T \(\rightarrow\) G;
4. G \(\rightarrow\) C, C \(\rightarrow\) T, T \(\rightarrow\) A, A \(\rightarrow\) G;
5. G \(\rightarrow\) A, A \(\rightarrow\) T, T \(\rightarrow\) C, C \(\rightarrow\) G; and
6. G \(\rightarrow\) A, A \(\rightarrow\) C, C \(\rightarrow\) T, T \(\rightarrow\) G.

A complementary rule\textsuperscript{23} is defined, which processes the alphabet in doubles instead of one by one. Assume that \((xx)\) is the token and \(D(xx)\) defines its complement.

### 2.4 Improved Expanded XOR Operation

The improved expanded XOR operation\textsuperscript{24} is applied to enhance the security and to increase the complexity of information. For two inputs \(x = \sum_{i=0}^{7} x_i \cdot 2^i\) and \(r = \sum_{i=0}^{10} r_i \cdot 2^i\), the Extended XOR (eXOR) operation can be defined as

\[
eXOR(x, r) = \sum_{i=0}^{7} \text{not}(x_i \oplus r_i \oplus r_{i+3}) \cdot 2^i ,
\]

where not(\(x\)) flips a single bit \(x\), and the “\(x \oplus y\)” represents the XOR operation. It has the following performance:

If \(eXOR(x, r) = t\), then \(eXOR(t, r) = x\).

This property can be proved by Table 3.

### Table 1 XOR operation of DNA sequence.

\[
\begin{array}{cccc}
\oplus & A & C & G \\
A & A & C & G \\
C & C & A & T \\
G & G & T & A \\
T & T & G & C \\
\end{array}
\]

### Table 2 A legal two-by-two complementary rule.

\[
\begin{array}{cccc}
\text{Token} & \text{Complement} & \text{Token} & \text{Complement} \\
AA & CG & CG & TC \\
TC & GC & GC & CT \\
CT & TT & TT & AG \\
AG & TA & TA & GA \\
GA & CA & CA & AT \\
AT & CC & CC & GG \\
GG & TG & TG & GT \\
GT & AC & AC & AA \\
\end{array}
\]

Here, the same property must apply as follows:

\[
\{ xx \neq D(xx) \neq D[D(xx)] \neq D[D[D(xx)]] \} \neq \ldots \neq D^{15}(xx) ,
\]

\[
xx = D^{16}(xx)
\]

Notice that the double complement of \(xx\) is \(D[D(xx)]\). \(D[D[D(xx)]]\) is its triple complement and \(D^{15}(xx)\) is its 15-fold complementary. The number of two-by-two complementary rules is 15! (1307674368000). It is far more than traditional complementary rules that total 3! (6) legal rules. The method could expand key space for a cryptographic system efficiently. A legal complementary rule is shown in Table 2.

### Table 3 The results of not \((x_i \oplus r_i \oplus r_{i+3})\).

\[
\begin{array}{cccc}
\text{x_i} & 00 & 01 & 10 \\
0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{cccc}
\text{r_i+r_{i+3}} & \\
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
\end{array}
\]
3 Image Encryption and Decryption Scheme

3.1 Secret Key and Random Number Generation

In this scheme, gray level images with the size of $M \times N$ are applied to demonstrate the proposed scheme. The initial values $x_0$ and $y_0$ could be calculated as follows:

$$x_0^1 = \left(\frac{x_0^0 + N_0}{1000}\right) \mod 1, \quad y_0^1 = (y_0^0 + x_0^1) \mod 1, \quad (5)$$

$$x_0^2 = (x_0^0 + y_0^1) \mod 1, \quad y_0^2 = (y_0^0 + x_0^2) \mod 1, \quad (6)$$

where $x_0^0, x_0^2, y_0^0, y_0^1, y_0^2$ and $N_0$ represent the initial secret keys, $(x_0^0, y_0^0)$ denotes the initial condition of row encryption, and $(x_0^2, y_0^2)$ denotes the initial condition of column encryption. “$x \mod y$” refers to the module operation.

3.2 Matrix-Level Encryption

3.2.1 Row encryption

The steps of row encryption are displayed as follows:

Step 1. $i = 1$. The initial values $(x_i^0, y_i^0)$ are obtained by Eq. (5). Iterate 2D-SIMM $N_0$ times and the sequences are discarded in order to avoid the transient effect.

Step 2. Continue to iterate 2D-SIMM again and then obtain the new values $(x, y)$.

Step 3. The fractional parts of the values $(x, y)$ are converted into binary streams $(S, T)$.

Step 4. The 22 most significant bits of $S$ are employed to build random numbers $k_1$, and the 16 most significant bits of $T$ are adopted to build random numbers $k_2$.

$S_{21-11} \leftarrow$ first half 11 bits,
$S_{10-0} \leftarrow$ second half 11 bits, and
$k_1 \leftarrow S_{21-11} \text{XOR} S_{10-0}$.

$T_{15-8} \leftarrow$ first half 8 bits,
$T_{7-0} \leftarrow$ second half 8 bits, and
$k_2 \leftarrow T_{15-8} \text{XOR} T_{7-0}$.

$$k_1 = \text{mod}(k_1, 2047) + 1, \quad (7)$$

$$k_2 = \text{mod}(k_2, N - 1) + 1, \quad (8)$$

where $k_1, k_2$ are the integers, and $k_1 \in [1, 2047]$, $k_2 \in [1, N - 1]$.

Step 5. For $i^{th}$ row pixels $P(i, \ldots)$, do $P(i, \ldots) = e\text{XOR}[P(i, \ldots), k_1]$.

Step 6. A $k_2$-bit right cyclic shift is performed on $P(i, \ldots)$.

Step 7. $i = i + 1$ and update $(x, y)$ by

$$\begin{align*}
  x &= x + s_1 - \lfloor x + s_1 \rfloor, \\
  y &= y + s_1 - \lfloor y + s_1 \rfloor,
\end{align*} \quad (9)$$

where $s_1$ is the mean of $P(i, \ldots)$.

Step 8. Do steps 1 to 7 again until $i > M$.

3.2.2 Column encryption

The steps of column encryption are shown as follows:

Step 1. $j = 1$. The initial values $(x_0^2, y_0^2)$ are produced by Eq. (6). Iterate 2D-SIMM $N_0$ times and the sequences are discarded for avoiding transient effect.

Step 2. Iterate 2D-SIMM once again and get new $(x, y)$.

Step 3. The fractional part of the values $(x, y)$ is converted into binary streams $(U, V)$.

Step 4. The 22 most significant bits of $U$ are employed to build random numbers $k_3$, and the 16 most significant bits of $V$ are employed to build random numbers $k_4$.

$U_{21-11} \leftarrow$ first half 11 bits,
$U_{10-0} \leftarrow$ second half 11 bits, and
$k_3 \leftarrow U_{21-11} \text{XOR} S_{10-0} - 1$.

$V_{15-8} \leftarrow$ first half 8 bits,
$V_{7-0} \leftarrow$ second half 8 bits, and
$k_4 \leftarrow U_{15-8} \text{XOR} S_{7-0}$.

$$k_3 = \text{mod}(k_3, 2047) + 1, \quad (10)$$

$$k_4 = \text{mod}(k_4, M - 1) + 1, \quad (11)$$

where $k_3, k_4$ are the integers, and $k_3 \in [1, 2047]$, $k_4 \in [1, M - 1]$.

Step 5. For $j^{th}$ column pixels $P(\cdot, j)$, do $P(\cdot, j) = e\text{XOR}[P(\cdot, j), k_3]$.

Step 6. Connect $P(\cdot, j)$ into a circle. Shift the pixels to up $k_4$ steps.

Step 7. $j = j + 1$ and update $(x, y)$ by

$$\begin{align*}
  x &= x + s_2 - \lfloor x + s_2 \rfloor, \\
  y &= y + s_2 - \lfloor y + s_2 \rfloor, \quad (12)
\end{align*}$$

where $s_2$ is the mean of $P(\cdot, j)$.

Step 8. Do steps 1 to 7 in a loop until $j > N$.

New pixel matrix $P$ is obtained finally.

3.3 DNA-Level Encryption

DNA encryption is depicted as follows:

Step 1. DNA encoding is performed on the new matrix $P$, and DNA-encoded matrix $P_b$ is obtained with size $M \times 4N$.

Step 2. A chaotic sequence $C = (c_1, c_2, \ldots, c_{MN})$ is generated under initial condition $x_0^0$ and $y_0^0$ using Eq. (1). Here, $c_i = x_i (i = 1, 2, \ldots, MN)$.

Step 3. Convert $c_i$ $(1 \leq i \leq MN)$ into corresponding binary format. The former 8 bits are encoded using DNA encoding rule. The length of sequence $C_1$ is $4MN$.

Step 4. Rearrange the sequence $C_1$ to form a matrix $Q$ with size $M \times 4N$.

Step 5. The two matrices $Q$ and $P_b$ are executed DNA XOR operation to generate matrix $H$. 

---

The text is a continuation of the document, discussing the encryption scheme in detail, including the generation of secret keys, random number generation, and the encryption process through row and column operations, as well as DNA-level encryption.
\[ H(1) = Q(1) \oplus P_b(1) \oplus \text{mod}(N_0, 256), \]
\[ H(t) = Q(t) \oplus P_b(t) \oplus H(t-1), \quad t = 2, 3, \ldots, 4M. \]

Step 6. Iterate 2D-SIMM once again and get the new \((x', y')\)

\[ z = \text{mod}([x'] \times 10^{14}], 15!\) + 1, \quad t = 2, 3, \ldots, 4M. \]

Step 7. Iterate 2D-SIMM \(N_0\) times and the sequences are discarded with the initial values \(x^0_1, y^0_1, x^0_2, y^0_2, \ldots, x^0_M, y^0_M\).

Step 8. Iterate 2D-SIMM \(f_0\) \(= \text{max}(M, 2N)\) times again. Then, two chaotic sequences \(E = (x_1, x_2, \ldots, x_M)\) and \(F = (y_1, y_2, \ldots, y_{2N})\) are generated.

Step 9. Transform \(E\) and \(F\) into matrices \(K_1(M, 1)\) and \(K_2(1, 2N)\). Multiply \(K_1\) and \(K_2\) to obtain matrix \(K\) with size \(M \times 2N\).

\[ L(i, j) = \text{mod}([K(i, j) \times 10^{10}], 15!\) + 1. \quad i \in [1, M], j \in [1, 2N], \quad \text{and} \quad L(i, j) \leq 1, 15. \]

Step 10. Use the \(z\)th two-by-two DNA complementary rule to operate on matrix \(H\).

\[ K(i, j) = \begin{cases} 0, & \text{for } L(i, j) \text{ is equal to } 0 \text{ and } w \text{ is integer}, \\
\text{mod}([K(i, j) \times 10^{10}], 15!\), & \text{otherwise}.
\end{cases} \]

Step 11. Convert DNA cipher matrix \(H\) into decimal number.

Step 12. Cipher image is obtained finally.

### 4 Simulation Results

The experiments of the proposed algorithm are simulated on the MATLAB 2010b platform. In this paper, Lena, Baboon, peppers, Terrace, and Jokul are used as original images. The size of the plain image is \(256 \times 256\), as shown in Fig. 1. The initial keys are set \((x^0_1, x^0_2, y^0_1, y^0_2, \text{and } N_0) = (0.3462, 0.5484, 0.7425, 0.8562, \text{and } 150).\)

The experimental results are demonstrated in Fig. 2. Figures 2(a)–2(c) are plain images, cipher images, and decrypted images.

The decrypted image is lossless and the same as the plain image with the proposed scheme.

Two schemes of Ref. 16 and 21 are employed for contrasts of performance evaluations. Encrypted images with different methods are shown in Fig. 3.

### 5 Security Analysis

An excellent encryption algorithm could resist many kinds of attacks, such as a statistical attack, brute-force attack, differential attack, and plaintext attack, and so on.

#### 5.1 Key Space and Sensitivity Analysis

An excellent image encryption algorithm should be very sensitive to secret keys, and key space should be large enough to resist the brute-force attack. In the proposed scheme, the secret keys are \((x^{01}_1, x^{02}_1, y^{01}_1, y^{02}_1, \text{and } N_0).\) The number of two-by-two DNA complementary rules is a factorial of 15 (15!). If the precision of the system is \(10^{-16}\), then the key space of the proposed scheme is \(10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 10^{16} \times 15! \approx 1.3 \times 10^{76}.\) It will be large enough to withstand an exhaustive attack.

In the paper, secret keys are set \((x^{01}_1, x^{02}_1, y^{01}_1, y^{02}_1, \text{and } N_0) = (0.3462, 0.5484, 0.7425, 0.8562, \text{and } 150).\) If a tiny alteration \((10^{-6})\) is brought in one of the initial values, the others remain the same. The decrypted images are depicted in Fig. 4. The difference between improper decrypted images [Figs. 4(b)–4(f)] and the plain image is almost 99.7%. So, the proposed scheme is very sensitive to the system key.

#### 5.2 Histogram Analysis

An excellent encryption scheme should provide the flat histogram of the encrypted image. The histograms of a plain image and its encrypted image are shown in Fig. 5. It indicates that the numbers of every pixel value of the encrypted image are nearly even. It demonstrates that a statistical attack is invalid to the proposed algorithm. Histograms of plain and cipher images are displayed in Figs. 5(a) and 5(b).
5.3 Correlation Analysis

The correlation coefficient \( r_{xy} \) between two adjacent pixels \( x \) and \( y \) are defined as

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

where \( \text{cov}(x, y) = \frac{1}{N} \sum_{i=1}^{N} [x_i - E(x)][y_i - E(y)] \), \( E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i \), \( D(x) = \frac{1}{N} \sum_{i=1}^{N} [x_i - E(x)]^2 \).

7225 pairs of adjacent pixels from the plain image and encrypted image are chosen in the horizontal, vertical, and diagonal directions. Figure 6 shows the correlation of two adjacent pixels in the Lena image and its cipher image. It can be shown that correlation is very high in the plain image but correlation is extremely low in the cipher image.

Table 4 shows the values of correlation coefficients of two adjacent pixels in Fig. 6. The results in the proposed scheme are compared with the results in Refs. 16, 21, 22, and 25. The results reveal that the proposed algorithm is rather good.

5.4 Information Entropy

Information entropy is one of the most important features of randomness. If \( m \) is the information source, then information entropy is calculated as follows:
Fig. 4 Decrypted images with different secret keys: (a) decrypted image using the right key, (b) decrypted image with $x_0^0 + 10^{-6}$, (c) decrypted image with $x_0^0 + 10^{-6}$, (d) decrypted image with $y_0^0 + 10^{-6}$, (e) decrypted image with $y_0^0 + 10^{-6}$, and (f) decrypted image with $N_0 + 1$.

Fig. 5 Histograms of plain and cipher images (Baboon, Terrace, and Lena): (a) histogram of plain image and (b) histogram of encrypted image.
\[ H(m) = - \sum_{i=1}^{L} p(m_i) \log_2 p(m_i), \quad (17) \]

where \( p(m_i) \) denotes the probability of symbol \( m_i \) and \( L \) is the total number of \( m_i \). The maximum information entropy is 8. The information entropy of cipher images is listed in Table 5.

### 5.5 Differential Attack

Number of pixels change rate (NPCR) and unified average changing intensity (UACI) are two parameters that are most widely adopted to measure the sensitivity to the plain image.\(^\text{21}\) NPCR and UACI are used to test the system to resist differential attacks. NPCR and UACI are calculated as

\[ \text{NPCR} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i, j) \times 100\%, \quad (18) \]

\[ \text{UACI} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left| C_1(i, j) - C_2(i, j) \right| \times \frac{255}{100\%}, \quad (19) \]

where

\[ D(i, j) = \begin{cases} 0, & \text{if } C_1(i, j) = C_2(i, j) \\ 1, & \text{else} \end{cases} . \]

\[ \text{NPCR} = 99.61\% \text{ and UACI} = 33.32\%. \]

The experiments demonstrate that the proposed scheme could effectively resist a plaintext attack and differential attack.

### 5.6 Noise Attack

During processing, transmission or an attack from an intruder, the encrypted image will be inevitable to confront with noise. The robust image encryption scheme should withstand a slight noise attack. Figure 7 shows the decrypted Lena with different kinds of noises and intensities. As shown in Fig. 7, the decrypted image can be recovered even the encrypted image with noise.

---

**Figure 6** Correlation analysis in three directions of plain image and cipher image: upper frame with horizontal distribution, center with vertical distribution, and lower with diagonal distribution.

**Table 4** Analysis of correlation coefficients.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Horizontal</th>
<th>Vertical</th>
<th>Diagonal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plain image</td>
<td>0.9425</td>
<td>0.9701</td>
<td>0.9248</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.0013</td>
<td>0.0021</td>
<td>-0.0024</td>
</tr>
<tr>
<td>Ref. 16</td>
<td>-0.0028</td>
<td>0.0032</td>
<td>0.0052</td>
</tr>
<tr>
<td>Ref. 21</td>
<td>0.0033</td>
<td>-0.0028</td>
<td>-0.0039</td>
</tr>
<tr>
<td>Ref. 22</td>
<td>0.0048</td>
<td>-0.0037</td>
<td>0.0034</td>
</tr>
<tr>
<td>Ref. 25</td>
<td>-0.0062</td>
<td>0.0076</td>
<td>-0.0053</td>
</tr>
</tbody>
</table>

**Table 5** Information entropy of cipher image.

<table>
<thead>
<tr>
<th>Image</th>
<th>Lena</th>
<th>Terrace</th>
<th>Peppers</th>
<th>Jokul</th>
<th>Baboon</th>
</tr>
</thead>
<tbody>
<tr>
<td>( H(m) )</td>
<td>7.9971</td>
<td>7.9972</td>
<td>7.9973</td>
<td>7.9969</td>
<td>7.9974</td>
</tr>
</tbody>
</table>

NPCR = \( \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} D(i, j) \times 100\% \)
5.7 Known-Plaintext and Chosen-Plaintext Attacks

In the proposed scheme, the iteration condition of 2D-SIMM will be changed by the encrypted pixel value. Even if an invader uses all black or white images as the chosen plain image, the proposed scheme could resist attacks. That is because row encryption, column encryption, and DNA-level encryption will be different if the plain image is changed. Therefore, the proposed scheme could resist known-plaintext and chosen-plaintext attacks.

6 Conclusion

In this paper, an image encryption scheme is proposed based on two-by-two DNA complementary rules and 2D-SIMM. First, the plain image is performed confusion and permutation operations. Then, DNA-level encryption is executed. The extended XOR operation is applied to help the image encryption scheme to resist plaintext attacks. The experimental results prove that the scheme could afford a differential attack, brute-force attack, statistical attack, and plaintext attack. The security of the system is very high. The proposed method is suitable for practical application.

Disclosures

I declare that there is no conflict of interest in the manuscript.

Acknowledgments

This research was financially supported by the National Natural Science Foundation of China (Grant No. 61272469), the Natural Science Foundation of Fujian Province (Grant No. 2016J05153), and the Outstanding Youth Scientific Research Training Program of Fujian Province (2017).

References


Shuliang Sun received his BS degree from Hangzhou Dianzi University in 2003, his MS degree from Guangxi University in 2006, and his PhD from Tongji University in 2011. He is a teacher at the School of Electronics and Information Engineering, Fuqing Branch of Fujian Normal University, China. His research interests include optical pattern recognition, optical image processing, and optical communication.