An image security system using bit-level operation based on hyper-chaos system

Weinan Wang
College of Mathematics and Physics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Abstract
In order to overcome the shortcoming of the presented image encryption schemes [1, 2], a novel self-adaptive image security system using bit-level operation based on hyper-chaos is presented. In the proposed scheme, a 2D Logistic chaotic map is used to scramble the position of image pixels. Then, in order to obtain the effect of diffusion, the pseudorandom sequences generated by hyper-chaotic system are utilized to change the values of permuted image pixels. Simulation experiment results indicate that the proposed algorithm is of high security. It can not only possess a large key space but can also resist exhaustive attack, differential attack and chosen-plaintext attack, etc.

PACS: 42. 30. Kq, 42. 30.Va
Key words: Image encryption; 2D Logistic; Hyper-chaos; Chaos-based cipher

1. Introduction
With the rapid development of computer networks, multi-media and digital communications, more people are paying attention to transmission of images and long-distance video conference. At the same time, sharing of digital products exposes the security of themselves to unauthorized individuals who may want to copy, destroy or intercept some information. As a result, the security and protection of the digital images transmission in open networks is gradually becoming a hot issue. More effective solutions are required to preserve those confidential information and digital products. Traditional ciphers such as Data Encryption Standard (DES), Advanced Encryption Standard (AES) and International Data Encryption Algorithm (IDEA), etc, can not meet the challenge of protecting image data efficiently since images are characterized by a bulk of data and have high redundancy and strong correlation between different pixels. Consequently, chaos-based encryption methods emerged.

The chaos system is a nonlinear dynamical system which has some good features such as sensitivity to initial values and system parameters, mixing properties and ergodicity, etc. Since Matthews [3] first proposed that chaotic system could be used in cryptography in 1989, various image encryption schemes based on chaotic system are proposed by researchers [4-7]. Fridrich suggested that an image encryption algorithm based on chaos should include two processes [7]: permutation and diffusion. And the extant image encryption algorithms based on chaotic system most have these two stages. In permutation stage, the pixel matrix of the plain image is disordered by matrix transformation. In the diffusion stage, the values of the pixels are changed by a pseudorandom sequence generated from chaos systems. Compared with the permutation stage, diffusion of the image may reach a better encryption effect since the values of the pixels have been overlaid by a chaotic sequence. For example, Lian et al. [6] proposed a spatiotemporal-chaos-based image encryption scheme with a relative low time cost. Patidar et al. [8] proposed an image encryption algorithm combining Logistic map and 2D Standard map. Moreover, other model-based algorithms were also suggested by researchers [9-15]. Yen et al. [12] proposed Bit Recirculation Image Encryption (BRIE) which was a secure cryptosystem. Zhu et al. [13] proposed a novel bit-level-based encryption algorithm comprising of binary bits operation defined in classical substitution-diffusion architecture. However, some of the bit-level-based encryption schemes are insecure and weak to differential attack and chosen-plaintext attack. Xiang et al. [14] proposed an image encryption which only encrypts the higher 4 bits of a pixel binary value while the lower 4 bits are left unchanged. Palacios et al. [15] suggested a remedy which is also insecure and can not resist differential attack. Recently, Xiao et.al [1] and Gao et.al [2] both presented novel image encryption schemes based on dynamic sequences of multiple chaotic systems, which possesses good statistical properties and fast encryption speed. However, both of the algorithms lack enough diffusion effect and hence cannot resist differential attack. Concerning the previous research results which have unsatisfactory performance, the image encryption can be improved if combining pixel operation with bit-level operation of pseudorandom sequences or utilizing multi-dimensional chaotic maps which have a more complex structure. Therefore, In order to overcome the shortcoming of the image encryption schemes mentioned above, further studies are still needed. In this paper, we proposed a novel self-adaptive image encryption algorithm combining pixel-level operation and bit-level operation based on hyper-chaos system. First, a 2D Logistic chaotic map is employed to shuffle pixel matrix of the plaintext. Furthermore, an 8-bits binary sequence is generated from the hyper-chaos system to make gray-level transformation of the shuffled matrix.

The rest of this paper proceeds as follows: in Section 2, an introduction of relevant theory
regarding chaotic maps is demonstrated. The proposed encryption algorithm is described in detail in Section 3. In Section 4, simulation experiment results are shown. Section 5 presents the security analysis together with algorithm comparison. The conclusion is drawn in Section 6.

2. Basic theory of multi-dimensional chaos system

2.1. 2D Logistic map

A 2D Logistic map is used in the proposed encryption algorithm and it can be defined as Eq (1):

\[
\begin{align*}
    x_{i+1} &= \mu_1 x_i (1-x_i) + \gamma_1 y_i^2 \\
    y_{i+1} &= \mu_2 y_i (1-y_i) + \gamma_2 (x_i^2 + x_i y_i)
\end{align*}
\] (1)

When \(2.75 < \mu_1 \leq 3.4\), \(2.75 < \mu_2 \leq 3.45\), \(0.15 < \gamma_1 \leq 0.21\), \(0.13 < \gamma_2 \leq 0.15\), \(x_i, y_i \in (0,1]\), the system is in chaotic state, and all the initial values and system parameters can be kept as secret keys.

2.2. Hyper-chaos system

In the proposed algorithm, a hyper-chaos system is employed to generate a pseudorandom sequence to make gray-level transformation of the image, which can be modeled by Ref. [16]:

\[
\begin{align*}
    a &= (x_2 - x_4), \\
    b &= -x_1 x_2 + d x_2 + c x_2 - x_4, \\
    c &= x_1 x_2 - b x_3, \\
    k &= x_1 + k.
\end{align*}
\] (2)

Where \(a, b, c, d\) and \(k\) are system parameters. When \(a=36, b=3, c=28, d=-16\) and \(-0.7 \leq k \leq 0.7\), the hyper-chaos system is in hyper-chaotic state. At the same time, it has more complex dynamics structure than chaotic systems which guarantees a more pseudorandom sequence. Hence the hyper-chaos system is much safer than chaos systems in cryptography algorithms. The hyper-chaos attractors are shown in Fig. 1. With parameters \(a=36, b=3, c=28, d=-16\) and \(k = 0.3\).

3. Proposed image encryption cryptosystem

3.1. Pseudorandom sequence generation

3.1.1 Method of generating index number for scrambling

A 2D Logistic map and a give \(\{x_0, y_0\}\) is adopted to iterate \(S\) times, where \(S\) is the sum of XOR results of the whole original image and is kept as secret key. Hence, a new \(\{x_{S+1}, y_{S+1}\}\) regarded as the new initial values of the system is derived. By continuing iterating Eq (1), we can obtain a modified sequence:

\[
\begin{align*}
    m &= mod(a_k \times 10^{14} + S, M), \\
    n &= mod(b_k \times 10^{14} + S, N),
\end{align*}
\] (3)

Where \(k = 1, 2, 3, \ldots, M\), \(M\) is chosen according to the size \(M \times N\) of original image and function \(mod(x, y)\) returns the remainder after division.

3.1.2 Method of extracting binary sequence from hyper-chaos

Four pseudorandom sequences are derived (2), and then we map the four values of each round into \((0, 1)\) by \(mod 1\) and a binary sequence matrix would be obtained according to the following threshold function \(f(x)\):

\[
\begin{align*}
    f(x) = \begin{cases} \\
        0, 0 < t \leq 0.25, \\
        1, 0.25 < t < 0.5, \\
        0, 0.5 \leq t < 0.75, \\
        1, 0.75 \leq t \leq 1.
    \end{cases}
\end{align*}
\] (4)

For example, as for the \(i^{th}\) round of hyper-chaos, we get \(w_i=11.3120, x_i=0.9831, y_i=10.8123, z_i=9.1131\), then four values of \(f(x)\) can be obtained: \(f(w_1)=01, f(x_1)=11, f(y_1)=11, f(z_i)=00\) and one new binary sequence, i.e. \(01111100\), is generated according to the order from \(w_i\) to \(z_i\). Furthermore, The pixels of the permuted image are arranged to a set \(A=\{A_1, A_2, \ldots, A_{M\times N}\}\) in the order from left to right and then top to bottom. Hence, each pixel’s corresponding 8-bit encryption-number can be obtained by the
formula \( m = \text{mod}(k, 4) \) and Table 3, where \( k \) is the \( k \)th pixel to be encrypted. For instance, in the \( k \)th round of iteration of hyper-chaos when we encrypt the \( k \)th pixel of the set \( A \), we get \( m = 1 \), then the new 8-bits binary number is generated by the order of \( x_k \rightarrow y_k \rightarrow z_k \rightarrow w_k \). Consequently, a binary sequence \( B = \{ B_1, B_2, \ldots, B_{M \times N} \} \) is generated from the hyper-chaotic system.

<table>
<thead>
<tr>
<th>( m )</th>
<th>Order of forming the new bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( w_k \rightarrow x_k \rightarrow y_k \rightarrow z_k )</td>
</tr>
<tr>
<td>1</td>
<td>( x_k \rightarrow y_k \rightarrow z_k \rightarrow w_k )</td>
</tr>
<tr>
<td>2</td>
<td>( y_k \rightarrow z_k \rightarrow w_k \rightarrow x_k )</td>
</tr>
<tr>
<td>3</td>
<td>( z_k \rightarrow w_k \rightarrow x_k \rightarrow y_k )</td>
</tr>
</tbody>
</table>

3.2. Design of the encryption scheme

**Step1:** XOR all the pixels of the plain image, then we get the sum \( S \) of XORing the original image and \( \{x_{S+1}, y_{S+1}\} \) generating from Eq (1) are regarded as the new initial values of 2D Logistic chaotic system.

**Step2:** For each round of iteration of 2D Logistic, different \( m \) and \( n \) are generated by Eq (3). Scramble the image pixel matrix from the first row to the last row of each column, then the first column to the last column of each row, which can be denoted as follows:

```csharp
for (column = 1; column <= 256; column++)
{
    for (row = 1; row <= 256; row++)
    {
        PIXEL (m [row], column) \oplus\ Pixel (n [row], column);
    }
}
for (row = 1; row <= 256; row++)
{
    for (column = 1; column <= 256; column++)
    {
        PIXEL (row, m [column]) \oplus\ Pixel (row, n [column]);
    }
}
```

**Step3:** Convert the pixel value of the shuffled image into a binary sequence \( A = \{A_1, A_2, \ldots, A_{M \times N}\} \) whose size is \( M \times N \).

**Step4:** The permuted image is encrypted with the above key stream in Eq (5):

\[
C_i = A_i \oplus \text{mod}(B_i + c_{i-1}, 256),
\]

where \( C_i \) means the \( i \)th cipher-text pixel. An illustration of encryption process is demonstrated in Fig.2.

The decryption process is the reverse operation of the encryption described above. The binary sequence is generated from hyper-chaos and the original image can be obtained through reverse XORing operation and shuffling operation.

4. Simulation experiments

The experiment is simulated on a PC with an Intel Core 2 Duo, 2.0 GHz CPU and 2GB memory, running on C# programming language. The standard 256×256 Lena image with 8-bits gray-level is chosen as the plain image. The secret keys are chosen as: \( a_0=0.2135, \ b_0=0.375, \ w_0=-11.0431, \ x_0=-7.9989, \ y_0=26.8729, \ z_0=0.1 \) and \( m, n \) are both set 3. Histograms of the plain image and cipher image are shown in Fig.3, respectively. From the histograms listed below, we can easily see that the cipher
image possesses the characteristic of well uniform distribution which means the proposed image encryption can resist statistical attack effectively.

5. Security analysis
An effective encryption scheme should resist all kinds of attacks, such as statistical attack, exhaustive attack, etc. In this section, several security analyses are performed below to analyze the security of the proposed encryption algorithm.

5.1. Key space analysis
In the proposed algorithm, initial values of 2D Logistic and hyper-chaos: \(a_0, b_0, w_0, x_0, y_0, z_0\) are set as secret keys. If the double precision of each key is \(10^{-14}\), the key space of the algorithm is \(10^{40}\), which means a large enough key space. In addition, the system parameters \(\mu_1, \mu_2, \gamma_1, \gamma_2\) can also kept as secret keys. Therefore, the encryption algorithm has a large enough key space and can resist exhaustive attack.

5.2. Key sensitivity analysis
To test the key sensitivity of the proposed algorithm, some secret key tests are carried out here. The keys \(a_0, b_0, w_0, x_0, y_0, z_0\) are set as keys. Then we make a slight change of one of the secret keys, for example, \(w_0 = 11.30000000000001\) and other keys are kept unchanged. Fig.4.(a) shows the result decrypted with the wrong keys. We can see the corresponding histogram of the decrypted image is fairly uniform. Furthermore, it is the same with other keys. The test result mentioned above indicates that even a slight change of the keys will make the decryption ineffective and no information about the original image will be obtained. It demonstrates that the proposed encryption algorithm has the ability to resist exhaustive attack.

5.3. Correlation coefficient analysis
As we all know that the correlation between adjacent pixels is very high. According to Shannon’s theory \([17]\), a safe cryptosystem should include permutation and diffusion to frustrate the attacks based on statistical analysis. Hence, an effective encryption scheme should lower the correlation between pixels. In order to test the correlation between pixels after encryption using the proposed algorithm, we randomly select 1000 pair adjacent pixels from horizontal, vertical and diagonal directions of the cipher-image, respectively. The statistical analysis experiment is carried out using the following formulas:

\[
\text{Corr}(x, y) = \frac{\text{Conv}(x, y)}{\sqrt{D(x)} \sqrt{D(y)}}
\]

Where \(\text{Conv}(x, y) = \frac{1}{N} \sum_{i} [x_i - E(x)][y_i - E(y)]\), \(D(x) = \frac{1}{N} \sum_{i} [x_i - E(x)]^2\), \(E(x) = \frac{1}{N} \sum_{i} x_i\), and \(x, y\) are the gray values of any two adjacent pixels in the image. Table 2 shows the correlation of pixels in the original image and its corresponding cipher image. From Table 2 we can easily see that the correlation from different directions of the original image is nearly to 1 while those of the cipher image is very small, which is close to 0. It indicates that the high correlation between pixels of the original image has been destroyed by the proposed encryption algorithm, which means that our algorithm is secure and effective enough to resist statistical attack. The distribution of horizontally and vertically adjacent pixels in original image ‘Lena’ and cipher image are shown in Fig.5. (a),(b).

Table 2 Correlation coefficient of two adjacent pixels in original image and cipher image

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal</td>
<td>0.974705</td>
<td>0.032733</td>
<td>-0.037013</td>
<td>-0.001781</td>
</tr>
<tr>
<td>Vertical</td>
<td>0.936715</td>
<td>-0.013145</td>
<td>0.021112</td>
<td>-0.001446</td>
</tr>
<tr>
<td>Diagonal</td>
<td>0.919089</td>
<td>-0.059202</td>
<td>0.046295</td>
<td>2.325E-04</td>
</tr>
</tbody>
</table>

5.4. Differential attack analysis
In order to find out the relationship between the original image and cipher image, attackers often
utilize differential attack to get the information of the original image. Encrypt the image before and after making a tiny change (often one pixel), and the two encrypted images will be obtained and compared, which is differential attack. In this section, two tests which are called NPCR (number of pixels change rate) and UACI (unified average changing intensity) are carried out. They are defined according to the formulas as follows:

\[
C(i, j) = \begin{cases} 
1, & \text{if } T_1(i, j) \neq T_2(i, j) \\
0, & \text{if } T_1(i, j) = T_2(i, j)
\end{cases}
\]

(8)

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

(9)

\[
UACI = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \left| T_1(i, j) - T_2(i, j) \right|}{255 \times M \times N} \times 100\% 
\]

(10)

Where \( M \) and \( N \) are the height and width of the image, \( T_1 \) and \( T_2 \) are two encrypted images whose original images have a one-pixel difference between each other. The results listed in Fig.6 and Table 3 demonstrate the results of NPCR and UACI. From Fig.6 and Table 3, we can see that the NPCR and UACI results of the algorithms in Ref [1, 2] should achieve the satisfactory performance (NPCR>0.996, UACI>0.334) for at least 18 rounds of encryption; however, the proposed encryption algorithm only needs two rounds of encryption to reach the satisfactory performance. Under regular conditions, the proposed encryption system can encrypt image or other multi-media products with a much lower time cost than the algorithms in Ref [1, 2]. Therefore, simulation results and comparison results imply that the new encryption scheme has reached an excellent performance compared with other algorithms and can resist differential attack effectively.

Table 3 Comparison of NPCR and UACI for different rounds

<table>
<thead>
<tr>
<th>Round number</th>
<th>NPCR</th>
<th>UACI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.05175E-05</td>
<td>1.52587E-05</td>
</tr>
<tr>
<td>2</td>
<td>6.10351E-05</td>
<td>1.52587E-05</td>
</tr>
<tr>
<td>3</td>
<td>1.22070E-04</td>
<td>1.52587E-05</td>
</tr>
<tr>
<td>4</td>
<td>2.44140E-04</td>
<td>1.52587E-05</td>
</tr>
<tr>
<td>5</td>
<td>4.88281E-04</td>
<td>1.52587E-05</td>
</tr>
</tbody>
</table>

6. Conclusions

We introduce a novel image encryption algorithm using bit-level operation based on hyper-chaos system in this paper. Firstly, a 2D Logistic map is employed to scramble the columns and rows of the pixel matrix of plain image. Furthermore, we utilize the hyper-chaos system to generate a matrix consisting of the 8-bit binary numbers according to the threshold function. Then XOR operation is done between the two matrices. Simulation experiment results and security analysis demonstrate that the proposed algorithm has a large key space and high encryption efficiency. In addition, the algorithm can resist exhaustive attack, statistical attack and differential attack, etc.

Acknowledgement

This paper was supported by the National Natural Science Foundation of China under Grant No. 61003256, the Natural Science Foundation of CQ CSTC (Nos 2009BB2282), the Doctor Foundation of Chongqing University of Posts and Telecommunications (A2009-01), the Foundation of Chongqing Key Laboratory of Electronic Commerce and Logistics (Nos ECML1003 and ECML1010) and the Chongqing municipal education commission research funds (KJ100505).

References


**Figures Caption**

Fig. 1. The attractors of the system with $k=0.3$. (a) $x_1-x_2$ plane (b) $x_1-x_3$ plane.

Fig. 2. Encryption process.

Fig. 3. Original image, cipher image and their corresponding histograms.(a) Original image, (b) cipher image, (c) Histogram of the original image, (d) Histogram of the cipher image.

Fig. 4. The sensitivity test to key $w_0$. (a) Decrypted with wrong keys, (b) Histogram of the decrypted image.

Fig. 5. The distribution of horizontally adjacent pixels in the original image and cipher image.(a) The distribution of horizontally adjacent pixels in the original image, (b) The distribution of horizontally adjacent pixels in the cipher image.

Fig. 6. NPCR and UACI for the plain image ‘Lena’ encrypted by algorithms in [1],[2] and the proposed algorithm. (a) NPCR of the proposed algorithm, (b) UACI of the proposed algorithm, (c) NPCR of the algorithm of [1], (d) UACI of the algorithm of [1], (e) NPCR of the algorithm of [2], (f) UACI of the algorithm of [2].
Fig. 1.

```
<table>
<thead>
<tr>
<th>Original image</th>
<th>Permutation</th>
<th>Diffusion</th>
<th>Cipher image</th>
</tr>
</thead>
</table>
```

Keys from Plaintext

Pseudorandom sequence generated from 2D Logistic 2D logistic

Pseudorandom binary sequence from hyper-chaos

Fig. 2.

Fig. 3.
Fig. 4.

Fig. 5.
通讯作者联系方式：
姓名：王伟男
地址：重庆邮电大学数理学院
电话：15520047158
Email: billwang.super@yahoo.com.cn.