A Novel Color Image Encryption Algorithm Based on Chaotic Maps
Huibin Lu, Xia Xiao

Abstract
Due to the character of higher security of three-dimensional chaotic systems, a new algorithm based on Chen and Lorenz systems is proposed to encrypt color images. In order to increase the security of the cryptosystem, the algorithm firstly integrates image information into the Lorenz map, and then the image information is also mixed into the Chen map via the Lorenz map. Theoretical analysis and simulation results demonstrate that the key space is larger and the precision of the keys is higher than Xiao et al.[1], so the algorithm can resist the brute-force attacks. In addition, the distribution of grey values of the encrypted image has a random-like behavior, so the algorithm has strong ability against attacks.

Keywords: Chaos, Color Image, Encryption Algorithm

1. Introduction
With information network and communication technologies developing, how to protect the delivered information over the Internet and through wireless networks from attacking has become a vital issue. Chaotic maps are very complicated nonlinear dynamic systems, which are applied in the field of figure correspondence and encryption [1-4] because they are very sensitive to initial conditions and can generate good pseudorandom sequences. It is well known that a good encryption algorithm should be sensitive to the cipher keys, and the key space should be large enough to resist brute-force attacks. Due to many color pictures used in our life and three-dimensional chaotic systems which have more excellent encryption properties than low-dimensional chaotic systems, this paper proposes a new color image encryption algorithm based on two 3D chaotic maps. One is Lorenz map and the other is Chen map. The Lorenz model is one of the most classical three-dimensional chaotic models and one of the first models which contain chaotic behavior and chaotic attractors [5]. Since Matthews proposed the chaotic encryption algorithm in 1989, many researchers have been devoted to image encryption field based on chaotic systems. Then, Chen chaotic system was first presented by Prof. G. Chen in 1999 [6], although its equations are quite similar to Lorenz's, Chen and Lorenz are not equivalent mainly due to the parameter \( c \) in front of the state variable \( y \) in the second equation of Chen system, which leads to more complicated dynamic characters, and the features play a very important role in real-time secure image transmission. Recently, in [7], firstly, the two-dimensional cat map is extended to three-dimensional map, and then the authors employed the new 3D cat map and another chaotic system to encrypt the image. To overcome the drawbacks such as small key space and weak security of low-dimensional chaotic map, a new chaos-based algorithm is proposed in [8], which shows high-level security and acceptable efficiency. In this pursuit, the paper also designs a new encryption scheme. This paper utilizes two 3D chaotic maps to enhance the cryptosystem security. The new algorithm includes three phases. Firstly, the algorithm integrates image information into the Lorenz map. Secondly, we can get exceptionally pseudo-random sequences by the fourth order Runge-Kutta algorithm with a suitable step. Thirdly, the positions of the image pixels are shuffled by the Lorenz map and then the gray values of the permuted image are encrypted by the Chen system. Compared with Guan et al.[3], all positions of the image pixels are all changed according to the new algorithm, while in Guan et al., the position of (0,0)-pixel always keeps unchanged after iterations by Arnold cat map.

2. The two 3D chaotic models
To overcome those encryption limitations brought by low-dimensional chaotic systems, the two 3D chaotic systems are employed to encrypt the information.

2.1. Lorenz system

The equation of Lorenz chaotic system is described as following:

\[
\begin{aligned}
\frac{dx}{dt} &= \sigma(y-x) \\
\frac{dy}{dt} &= rx - xz - y \\
\frac{dz}{dt} &= xy - bz \\
\end{aligned}
\]  

Where \(\sigma\), \(r\) and \(b\) are parameters. When \(\sigma=10\), \(r>24.74\), \(b=8/3\), the system is chaotic. The trajectory of Lorenz system can be obtained by the fourth order Runge-Kutta algorithm, and choosing suitable step is also very vital to the pseudo-random performance of the sequence. We can get the above conclusion by a lot of tests: When step \(h=0.1\), the Lorenz sequence has better performance. Take the example of \(x\), we choose the same parameters: \(x_0=10\), \(y_0=20\), \(z_0=30\), \(\sigma=10\), \(r=28\), \(b=8/3\), \(n=6000\) and different steps: 0.01 shown by Fig.1 and 0.1 shown by Fig.2.

![Figure 1. The x sequence where h is 0.01](image1)

![Figure 2. The x sequence where h is 0.1](image2)

According to Fig.1 and Fig.2, the sequence shown by Fig.2 has smaller blank widow and better chaotic pseudo-random performance under the condition of the same iteration time than the sequence shown by Fig.1.

2.2. Chen system

Chen chaotic model is described as following:

\[
\begin{aligned}
\dot{x} &= a(y-x) \\
\dot{y} &= (c-a)x - xz + cy \\
\dot{z} &= xy - bz \\
\end{aligned}
\]  

Where \(a\), \(b\) and \(c\) are parameters. If one chooses \(a=35\), \(b=3\), \(c=28\), the system has a chaotic orbit. Take the example of \(x\), we choose the same parameters: \(x_0=0\), \(y_0=1\), \(z_0=0\), \(a=35\), \(b=3\), \(c=28\), \(n=6000\) and different steps: 0.001 shown by Fig.3 and 0.055555 shown by Fig.4.
From Fig.3 and Fig.4, we can remark when step is 0.055555; the pseudo-random sequence has better performance.

3. Design of the encryption scheme

3.1. Permutation algorithm

Step1: Obtain the $R$, $G$ and $B$ matrixes (the three color components Red, Green and Blue) of the color image of size $m \times n \times 3$, respectively. Afterwards, each color’s matrix (including $R$, $G$ and $B$) is converted into a vector of integers within $\{0, 1\ldots 255\}$. Then, the so obtained three vectors represent the plaintext which will be encrypted.

Step2: The $R$, $G$ and $B$ matrixes are rearranged respectively according to the following method [9]:

$$R = \begin{bmatrix} 01 & 02 & 03 \\ 11 & 12 & 13 \end{bmatrix} \quad XX = [21 \ 22 \ 23 \ 31 \ 32 \ 33]$$

Where the $R$ matrix represents the Red component of the plain-image, and the $XX$ matrix is generated by the Lorenz map.

According to the permutation scheme, the position of any pixel in $R$ is different with its position in $R3$, which will lead to be strong for favors attacks.
3.2. Encryption algorithm

Step1: The color image of size \( m \times n \times 3 \) is converted into its RGB components, and then Eq.(3) is employed to generate the values of \( k_1, k_2 \) and \( k_3 \) which are used to modify the keys in the proposed algorithm.

\[
k_1 = \text{sum}(\text{sum}(R)), \quad k_2 = \text{sum}(\text{sum}(G)), \quad k_3 = \text{sum}(\text{sum}(B))
\]

\[
k = (k_1 + k_2 + k_3)/10 \times 8
\]  

(3)

Step2: To resist statistical analysis, Shannon suggests that diffusion and confusion should be utilized in any cryptosystem [10]. According to the rules, the algorithm blends the pixel values of the plain-image into the initial values of the Lorenz map via the following Eq.(4):

\[
x_0 = 10 + k, \quad y_0 = 20 + k, \quad z_0 = 30 + k
\]  

(4)

Step3: Three sequences \( XX, YY \) and \( ZZ \) of size \( m \times n \) generated by Lorenz map are used to permute the \( R, G \) and \( B \) matrices of the plain-image according to Section 3.1.

Step4: Then the three sequences of \( XXC, YYC \) and \( ZZC \) of size \( m \times n \) are generated by the Chen map, whose three initial values are \( XX(100), YY(500) \) and \( ZZ(800) \) which are generated by the Lorenz map.

Step5: And then \( XXC, YYC \) and \( ZZC \) are changed by the following Eq.(5). Take an example of \( XXC \).

\[
XXC(i) = \text{mod}(\text{floor}(10^9 \times \text{abs}(XXC(i))),255) + 1
\]  

(5)

Where \( \text{abs}(x) \) returns the absolute value of \( x \), \( \text{floor}(x) \) rounds \( x \) to the nearest integers less than or equal to \( x \), and \( \text{mod}(x, y) \) returns the remainder after division.

Step6: Eq.(6) is utilized to modify the above values of \( XXC, YYC \) and \( ZZC \).

\[
\begin{align*}
XXC(i) &= \text{bitxor}(\text{mod}\left((k_2 + k_3), 256\right), XXC(i)) \\
YYC(i) &= \text{bitxor}(\text{mod}\left((k_1 + k_3), 256\right), YYC(i)) \\
ZZC(i) &= \text{bitxor}(\text{mod}\left((k_1 + k_2), 256\right), ZZC(i))
\end{align*}
\]  

(6)

Step7: Then the encrypted image can be obtained by the following Eq. (7).

\[
\begin{align*}
C_R(i) &= \text{bitxor}(XXC(i), R2(i)) \\
C_G(i) &= \text{bitxor}(YYC(i), G2(i)) \\
C_B(i) &= \text{bitxor}(ZZC(i), B2(i))
\end{align*}
\]  

(7)

Where the \( R, G \) and \( B \) matrices of the plain-image are rearranged into the \( R2, G2 \) and \( B2 \) matrices according to Section 3.1.

According to Eq.(3), Eq.(4), Eq.(6) and Eq.(7), it is clear that the generation of the key streams depends on the plaintext through all the color components \( R, G \) and \( B \), and every pixel value of the encryption matrix \( C \) includes the information of all the color components \( R, G \) and \( B \), i.e. the diffusion has been taken full advantage of. These features strengthen the cryptosystem security.

4. Simulated experiments and security analysis

In this section, simulation results will be given to demonstrate the satisfactory security of the new algorithm, including the most important ones like key space analysis, statistical analysis, and resistance attack test. The experiments were done by Matlab7.0, and used the color test image of size \( 275 \times 275 \).

The Lorenz map parameters: \( \sigma=10, r=28, b=8/3 \); the initial values: \( x_0 = 10 + k, y_0 = 20 + k, z_0 = 30 + k \) (\( k \) is related to the color image), step \( h \) is 0.1, iteration time: \( m \times n + 2500 \).

The Chen map parameters: \( a=35, b=3, c=28 \); the initial values: \( x_1 = XX(100), y_1 = YY(500), z_1 = ZZ(800) \) (\( XX(100), YY(500) \) and \( ZZ(800) \) are generated by Lorenz map), step \( h \) is 0.055555, iteration time: \( m \times n + 1500 \). The simulated results are shown by Fig.6.
Figure 6. Encryption results. (a) Original image. (b) Encrypted image. (c) The $R$ histogram of the original image. (d) The $R$ histogram of the encrypted image. (e) The $G$ histogram of the original image. (f) The $G$ histogram of the encrypted image. (g) The $B$ histogram of the original image. (h) The $B$ histogram of the encrypted image.

4.1 Key security analysis

Key security in the paper is guaranteed by the Lorenz map and the Chen map. The algorithm keys include: three initial values $(x_0, y_0, z_0)$, step $h$, and iteration number $F$; three initial values $(x_1, y_1, z_1)$,
step $hC$ and iteration number $FC$. If the precision is $10^{-15}$, the key space size will be $10^{120}$; while the key space in Ref. [1] is $10^8$. Moreover, the parameters of two chaotic maps are other parts of the secret keys. So the key space is large enough to make all kinds of brute-force attacks infeasible.

Key sensitivity test has been carefully performed and completely carried out with the image of size $275 \times 275$, with results summarized by Table 1.

<table>
<thead>
<tr>
<th>Name</th>
<th>Precision</th>
<th>$h$</th>
<th>$F$</th>
<th>$x_0$</th>
<th>$y_0$</th>
<th>$z_0$</th>
<th>$hC$</th>
<th>$FC$</th>
<th>$x_1$</th>
<th>$y_1$</th>
<th>$z_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$10^{-16}$</td>
<td>1</td>
<td>$10^{-15}$</td>
<td>$10^{-14}$</td>
<td>$10^{-14}$</td>
<td>$10^{-17}$</td>
<td>1</td>
<td>$10^{-14}$</td>
<td>$10^{-14}$</td>
<td>$10^{-14}$</td>
</tr>
</tbody>
</table>

We can see that one key with little movement (e.g., $hC$ is changed $10^{-17}$) will result in an incorrect decrypted image. Therefore we can conclude that the encryption algorithm is very sensitive to keys, and it can also resist the various attacks based on sensibility.

### 4.2. Statistical analysis

(1) Correlation analysis of two adjacent pixels [11]

It is well known that the adjacent pixels of an image have very high correlation coefficients in horizontal, vertical and diagonal directions. And correlation coefficients are used to quantify correlation property. Generally, the following formula is employed to test the correlation between two adjacent pixels in original image and encrypted image. We randomly select 1000 pairs of two adjacent pixels from an image. Calculate the correlation coefficient $r_{xy}$ of each pair via the following equations:

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

$$\text{cov}(x, y) = E(x - E(x))(y - E(y))$$

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

Where $x$ and $y$ denote two adjacent pixels, and $N$ is the total number of duplets $(x, y)$ obtained from the image. Table 2 lists the correlation coefficients of the color image and its cipher-image.

<table>
<thead>
<tr>
<th>Model</th>
<th>$R$</th>
<th>$RJ$</th>
<th>$G$</th>
<th>$GJ$</th>
<th>$B$</th>
<th>$BJ$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.9508</td>
<td>-0.005</td>
<td>0.9707</td>
<td>0.0018</td>
<td>0.9579</td>
<td>0.0002</td>
</tr>
<tr>
<td></td>
<td>0.9718</td>
<td>0.0032</td>
<td>0.9754</td>
<td>-0.0063</td>
<td>0.9818</td>
<td>0.0018</td>
</tr>
<tr>
<td></td>
<td>0.9435</td>
<td>-0.0019</td>
<td>0.9453</td>
<td>0.0029</td>
<td>0.9532</td>
<td>0.0153</td>
</tr>
</tbody>
</table>

Where $R$, $G$ and $B$ are the matrixes of the original image; while $RJ$, $GJ$ and $BJ$ are the matrixes of the cipher-image.

According to Table 2, we can remark that the correlation coefficients of the plain-image are equal to 1, implying that high correlation exists among pixels, while the correlation coefficients of the cipher-image are equal to 0, implying that no detectable correlation exists among pixels. Therefore the proposed algorithm can protect the cipher-image from statistical attacks.

(2) Histogram analysis

From Fig. 6, we can see that the histograms of the encrypted image are nearly uniform between 0 and 255, i.e. their statistical properties are totally different from the plain-image. Therefore, the proposed algorithm has high security against known-secret-text attacks and statistical attacks.

(3) Plain-text sensibility analysis

The opponent commonly may observe change of the results by means of making a very few differences (e.g., modify only one pixel) of the encrypted image. Therefore, a very vital relationship between the plain-image and the cipher-image may be revealed. If a significant change in the cipher-
image can be caused by a slight change in the plain-image by means of diffusion and confusion, then the algorithm would make differential attacks practically useless. In order to test the influence of a one-pixel change on the image encrypted by the new algorithm, \( NPCR \) (number of pixels change rate) and \( UACI \) (unified average changing intensity) are utilized to calculate the property. Their definitions are listed as follows [7]:

\[
NPCR = \frac{\sum_{i,j} D(i,j)}{m \times n} \times 100\% \tag{10}
\]

\[
UACI = \frac{\sum_{i,j} |HM(i,j) - HM1(i,j)|}{255 \times m \times n} \times 100\% \tag{11}
\]

The test needs two plain-images: the plain-image and the other image obtained by changing one pixel value of the plain-image. Then we encrypt the two images with the same keys to generate the corresponding cipher-images \( HM \) and \( HM1 \). Where the grey values of the pixel at position \((i, j)\) of \( HM \) and \( HM1 \) are denoted as \( HM(i,j) \) and \( HM1(i,j) \), respectively; \( m \) and \( n \) are width and height of the cipher-image. \( D(i, j) \) is determined by \( HM(i,j) \) and \( HM1(i,j) \), if \( HM(i,j) - HM1(i,j) = 0 \), \( D(i,j) = 0 \); otherwise, \( D(i,j) = 1 \).

<table>
<thead>
<tr>
<th>The R matrix of the encrypted image</th>
<th>The G matrix of the encrypted image</th>
<th>The B matrix of the encrypted image</th>
</tr>
</thead>
<tbody>
<tr>
<td>NPCR (%)</td>
<td>99.617%</td>
<td>99.611%</td>
</tr>
<tr>
<td>UACI (%)</td>
<td>33.699%</td>
<td>33.476%</td>
</tr>
</tbody>
</table>

From Table 3, we can observe that the values are very close to the ideal values (\( NPCR = 99.609\% \) and \( UACI = 33.46354\% \))[12], i.e. A very little change of the plain-image pixel values will lead to a significant change of the cipher-image.

(4) Information entropy analysis [13]

The entropy \( H(x) \) of a message source \( x \) can be quantified by Eq.(12).

\[
H(x) = - \sum_{i=1}^{x} p(x_i) \log_2(p(x_i)) \tag{12}
\]

Where \( x \) is the total number of symbols, \( x_i \in x \); where \( p(x_i) \) represents the probability of occurrence of \( x_i \), and \( \log_2 \) denotes the base 2 logarithm.

For the three matrixes \( R \), \( G \) and \( B \) of the image, the corresponding entropies are 7.99758, 7.99708 and 7.99749. And \( H(x) \) is equal to 8, which is the ideal situation. So these results mean that the cipher-images are close to a random source and the proposed algorithm is secure against entropy attack.

4.3. Resistance attack test

Because the encrypted image will be delivered by channel, which isn’t always ideal and exists all kinds of noises, we must consider the capacity of resisting noises of the proposed algorithm. In addition, a good algorithm should recover the plain-image even if the data of the encrypted image is destroyed. Fig.7 shows the results of resistance attack test.
According to Fig.7, we can observe that the algorithm can recover the plain-image even if the encrypted image is destroyed, so the new algorithm can resist cutting and noise.

5. Acknowledgement

This research was supported by the Education Department Foundation of Hebei Province (Grant No. 2007493), P.R. China.

6. References