TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol. 12, No. 11, November 2014, pp. 7952 ~ 7962
DOI: 10.11591/telkomnika.v12i11.6480

A Chaotic System Based Image Encryption Algorithm using Plaintext-related Confusion

Yong Zhang
School of Software and Communication Engineering, Jiangxi University of Finance and Economics,
Nanchang, P. R. China
E-mail: zhangyong@jxufe.edu.cn

Abstract

A new plaintext-related image encryption system based on the hyper-chaotic Lorenz system is proposed in this paper. In the proposed image encryption system, the hyper-chaotic Lorenz system is employed to generate six pseudo-random matrixes, where, two of them use the "add-modulus" operations to carry out the plaintext-unrelated image diffusion, and the other four matrixes are used to confuse the plaintext-related image. In the image confusion, each pixel will swap its location with another pixel, and the target location is determined by some elements in the plain image and the four matrixes. The proposed image encryption system can resist the chosen/known plaintext attacks due to the application of the plaintext-related image confusion in it. The simulation results show that the proposed encryption system also have the characters of fast encryption/decryption speed, large key space, strong key sensitivity, strong plaintext sensitivity, good statistical properties of cipher images, and large information entropy, etc. Thus, the proposed system can be used in practical communications.

Keywords: *image encryption, hyper-chaotic Lorenz system, plaintext-related confusion, diffusion, security analysis*

1. Introduction

Image encryption is an important research topic in image information security field. For image, due to its huge data volume and strong information redundancy characters, the traditional text-based data encryption methods, such as DES, AES and RSA, etc., are not suitable for its encryption. In recent decades, scientists have explored the chaotic system based image encryption technology, and obtained rich achievements [1-7]. In these image encryption systems, chaotic systems are employed to generate the secret code streams for encryption, and then encrypt the plain images into noise-like via circularly confusion and diffusion.

In some image encryption systems [8-11], their secret code streams (i.e. equivalent secret keys) are generated by iterating the chaotic systems with the secret keys server as the initial values or parameters, and are unrelated with the plain images. These make them vulnerable to the chosen/known plaintext attacks [12-16]. They need to increase their round number of the iterating to improve their capabilities of resisting the chosen/known plaintext attacks, thus their encryption/decryption speeds are reduced greatly. To make the image encryption system fight against the chosen/known plaintext attacks while the encryption speeds are high enough, some scholars have proposed plaintext-related image encryption methods, which can be divided into the following two categories:

(1) Divide the plain image into a plurality of blocks, and encrypt each image block sequentially. The secret code streams of each block are co-generated by the cipher block of its previous block and the secret keys. Thus, the cipher block of the current block is not only related with the secret key, but also indirectly related with its previous plain block. After two rounds of operations, each cipher block is indirectly related with the whole plain image. Therefore, different plain images correspond to different equivalent secret keys, makes the encryption system can resist the chosen/known plaintext attacks [17-22].

(2) Divide the secret key into two levels: the first level secret key is the symmetric key shared by both communication parties; the second level secret key is co-generated by the first level key and the plain images. Employ the first level key to encrypt the plain images to get the intermediate cipher images, and then employ the second level key to encrypt the intermediate

Received July 13, 2013; Revised September 18, 2014; Accepted October 8, 2014

cipher images to get the final cipher images. Thus, different plain images correspond to different second level keys, and thereby correspond to different equivalent keys. This kind of encryption system can resist the chosen/known plaintext attacks [23].

The foresaid two plaintext-related encryption systems have high cryptographic security. However, because plaintext-related image diffusion is used in both of them, and chaotic systems are iterated to generate new secret code streams in their diffusion process, their encryption speeds are slow. In order to improve the encryption speed, a new plaintext-related image encryption system is proposed in this paper. The proposed system uses the plaintext-related confusion and plaintext-unrelated diffusion while needs no round operations, thus its encryption speed is increased without loss of security.

The reminder of this paper is organized as follows: Section 2 details the encryption scheme of the proposed system; Section 3 gives some simulation results; Section 4 analyzes the security performance of the proposed system; Section 5 summaries the paper.

2. Image Encryption Scheme

2.1. Used Chaotic System

The hyper-chaotic Lorenz system is used in this paper. Its equation is as follows:

$$\begin{cases} \dot{x} = a(y-x) + w \\ \dot{y} = cx - y - xz \\ \dot{z} = xy - bz \\ \dot{w} = -yz + rw \end{cases} \tag{1}$$

Where, *a*, *b*, *c* and *r* are the parameters of hyper-chaotic Lorenz system. When a=10, b=8/3, c=28 and $-1.52<r\leq-0.06$, Equation (1) falls into the chaotic state. If r=-1, Equation (1) has four Lyapunov exponents, namely λ_1=0.3381, λ_2=0.1586, λ_3=0 and λ_4=-15.1752.

In Equation (1), the range values of the state variables x_0, y_0, z_0 and w_0 are, respectively, $x_0 \epsilon (-40,40)$, $y_0 \epsilon (-40,40)$, $z_0 \epsilon (1,81)$, $w_0 \epsilon (-250,250)$. $\{x_0, y_0, z_0, w_0\}$ is part of the secret key in the proposed system. When discrete the Equation (1) with the fourth-order Runge-Kutta method, the step size is 0.002.

2.2. Basic Principle of Encryption System

The typical plaintext-related image encryption system based on chaotic system has the structure as shown in Figure 1. They include several rounds of plaintext-unrelated confusion and plaintext-related diffusion operations. The chaotic system is employed to generate the secret code streams.

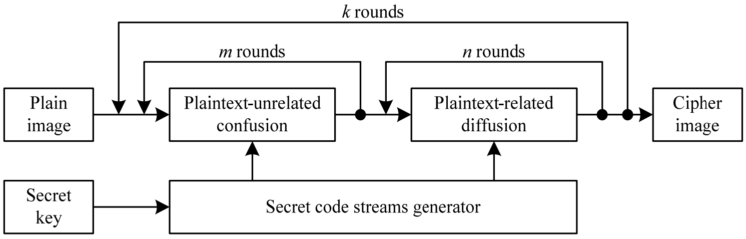
k rounds
m rounds
n rounds
Plain image
Plaintext-unrelated confusion
Plaintext-related diffusion
Cipher image
Secret key
Secret code streams generator

Figure 1. Structure of Typical Image Encryption System

The proposed encryption system is as shown in Figure 2. Different from Figure 1, in our proposed system: (1) The confusion is plaintext-related, while the diffusion is plaintext-unrelated; (2) One confusion and two diffusion operations without rounds needed, what improves the encryption/decryption speed greatly, and do not weaken the security of the encryption system.

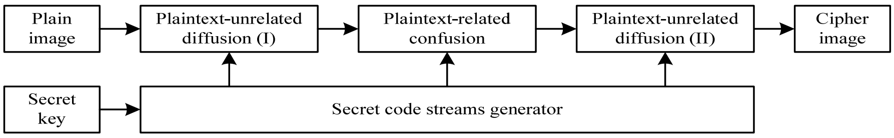
Plain image
Plaintext-unrelated diffusion (I)
Plaintext-related confusion
Plaintext-unrelated diffusion (II)
Cipher image
Secret key
Secret code streams generator

Figure 2. Structure of our Proposed Image Encryption System

2.3. Encryption Scheme

According to Figure 2, our proposed encryption system consists of four parts, namely, secret code stream generator, plaintext-unrelated diffusion I, plaintext-related confusion, and plaintext-unrelated diffusion II. The following four subsections will detail the procedures of the foresaid four parts.

The secret key of the proposed is $\boldsymbol{K}=\{x_0, y_0, z_0, w_0, r_1, r_2\}$, where, $\{x_0, y_0, z_0, w_0\}$ comes from Section 2.1, and r_1 and r_2 are two 8-bit random unsigned integers. Assume that the plain image is denoted by $\boldsymbol{P}$, whose size is $M\times N$, and grayscale level is L-bit.

2.3.1. Secret Code Stream Generating Algorithm

The hyper-chaotic Lorenz system as shown in Eq. (1), is employed to generate six pseudo-random matrixes, denoted by $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$, $\boldsymbol{W}$, $\boldsymbol{U}$ and $\boldsymbol{V}$ (all of size $M\times N$) by using the following concrete steps:

Step 1. Use $\{x_0, y_0, z_0, w_0\}$ in $\boldsymbol{K}$ as the initial values of Equation (1), iterate Equation (1) for r_1+r_2 times to bypass the transient state, and then continue iterating MN times to get four pseudo-random sequences, namely, $\{x_i\}$, $\{y_i\}$, $\{z_i\}$ and $\{w_i\}$, i=1,2,…,MN.

Step 2. Produce matrixes $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$, $\boldsymbol{W}$, $\boldsymbol{U}$ and $\boldsymbol{V}$ from the sequences of $\{x_i\}$, $\{y_i\}$, $\{z_i\}$ and $\{w_i\}$, i=1,2,…,MN, with the following formulas:

$$\boldsymbol{X}(k,l)=\text{floor}((x_{(k-1)\times N+l}+500 \bmod 1)\times 10^{13}) \bmod 2^L \tag{2}$$

$$\boldsymbol{Y}(k,l)=\text{floor}((y_{(k-1)\times N+l}+500 \bmod 1)\times 10^{13}) \bmod 2^L \tag{3}$$

$$\boldsymbol{Z}(k,l)=(\text{floor}(z_{(k-1)\times N+l}\times 10^{13}) \bmod M)+1 \tag{4}$$

$$\boldsymbol{W}(k,l)=(\text{floor}((w_{(k-1)\times N+l}+500 \bmod 1)\times 10^{12}) \bmod N)+1 \tag{5}$$

$$\boldsymbol{U}(k,l)=(\text{floor}((x_{(k-1)\times N+l}+ y_{(k-1)\times N+l}+500 \bmod 1)\times 10^{12}) \bmod M)+1 \tag{6}$$

$$\boldsymbol{V}(k,l)=(\text{floor}((z_{(k-1)\times N+l}+ w_{(k-1)\times N+l}+500 \bmod 1)\times 10^{12}) \bmod N)+1 \tag{7}$$

Where, floor(t) returns the integer not greater than t, '+500' converts the negative state values of x, y and w into the positive numbers.

2.3.2. Plaintext-unrelated Diffusion I

Convert the plain image $\boldsymbol{P}$ into the matrix $\boldsymbol{A}$ by employing pseudo-matrix $\boldsymbol{X}$ with the following steps:

Step 1. Let i=1, j=1.

Step 2. Transform $\boldsymbol{P}(i,j)$ into $\boldsymbol{A}(i,j)$ by using the following formula:

$$\boldsymbol{A}(i,j)=\boldsymbol{P}(i,j)+\boldsymbol{X}(i,j)+r_1 \bmod 2^L \tag{8}$$

Step 3. Let j=j+1.

Step 4. Transform $\boldsymbol{P}(i,j)$ into $\boldsymbol{A}(i,j)$ by using the following formula:

$$\boldsymbol{A}(i,j)=\boldsymbol{P}(i,j)+\boldsymbol{A}(i,j\text{-}1)+\boldsymbol{X}(i,j) \bmod 2^L \tag{9}$$

Step 5. If j<N, then goto Step 3.

Else

```
    $j$=1, $i$=$i$+1.
    If $i$<=$M$, then goto Step 6.
    Else goto Step 8.
End
```

Step 6. Transform ***P***(*i*,*j*) into ***A***(*i*,*j*) by using the following formula:

$$\boldsymbol{A}(i,j)=\boldsymbol{P}(i,j)+\mathrm{sum}(\boldsymbol{A}(i\text{-}1,1 \text{ to } N))+\boldsymbol{X}(i,j) \bmod 2^{L} \tag{10}$$

Where, sum(***t***) returns the sum of all the elements in the vector ***t***.

Step 7. Go to Step 3.

Step 8. End。

2.3.3. Plaintext-related Confusion

Confusion algorithm is used to disrupt the pixel locations of the image, so as to eliminate the correlation between adjacent pixels in the original image. The proposed plaintext-related confusion algorithm transforms the matrix ***A*** into the matrix ***B*** by using the following steps:

Step 1. As for a given pixel coordinate (*i*,*j*) in the image ***A***, calculate the value of coordinate (*m*,*n*) by using the following formula:

$$m=(\boldsymbol{U}(i,j)+\mathrm{sum}(\boldsymbol{A}(\boldsymbol{Z}(i,j), 1 \text{ to } N) \bmod M)+1 \tag{11}$$

$$n=(\boldsymbol{V}(i,j)+\mathrm{sum}(\boldsymbol{A}(1 \text{ to } M, \boldsymbol{W}(i,j)) \bmod N)+1 \tag{12}$$

If $m=i$ or ***Z***(*i*,*j*), or $n=j$ or ***W***(*i*,*j*), or ***Z***(*i*,*j*)=*i*, or ***W***(*i*,*j*)=*j*, then keep the location of ***A***(*i*,*j*) unchanged; else, swap the locations of ***A***(*i*,*j*) and ***A***(*m*,*n*).

Step 2. When the coordinate (*i*,*j*) traverses the whole image ***A*** in the scanning order from left to right and from top to bottom, repeat Step 1 to convert the matrix ***A*** into the matrix ***B***.

2.3.4. Plaintext-unrelated Diffusion II

Different from the algorithm described in Section 2.3.2, the plaintext-unrelated diffusion II in this subsection carries out the forward diffusion operations from the last pixel of the image. The algorithm of diffusion II employs the pseudo-random matrix ***Y*** to transform the matrix ***B*** into the matrix ***C*** by using the following steps:

Step 1. Let *i*=*M*, *j*=*N*.

Step 2. Convert ***B***(*i*,*j*) into ***C***(*i*,*j*) by using the following formula:

$$\boldsymbol{C}(i,j)=\boldsymbol{B}(i,j)+\boldsymbol{Y}(i,j)+r_2 \bmod 2^{L} \tag{13}$$

Step 3. Let *j*=*j*-1.

Step 4. Convert ***B***(*i*,*j*) into ***C***(*i*,*j*) by using the following formula:

$$\boldsymbol{C}(i,j)=\boldsymbol{B}(i,j)+\boldsymbol{C}(i,j+1)+\boldsymbol{Y}(i,j) \bmod 2^{L} \tag{14}$$

```
Step 5. If $j$>1, then goto Step 3.
    Else
        $j$=N, $i$=$i$-1.
        If $i$>=1, then goto Step 6.
        Else goto Step 8.
    End
```

Step 6. Convert ***B***(*i*,*j*) into ***C***(*i*,*j*) by using the following formula:

$$\boldsymbol{C}(i,j)=\boldsymbol{B}(i,j)+\mathrm{sum}(\boldsymbol{C}(i+1,1 \text{ to } N))+\boldsymbol{Y}(i,j) \bmod 2^{L} \tag{15}$$

Step 7. Go to Step 3.

Step 8. End.

The matrix ***C*** is the cipher image.

According to the description of encryption scheme, the decryption process is the reverse of encryption process.

3. Simulation Results

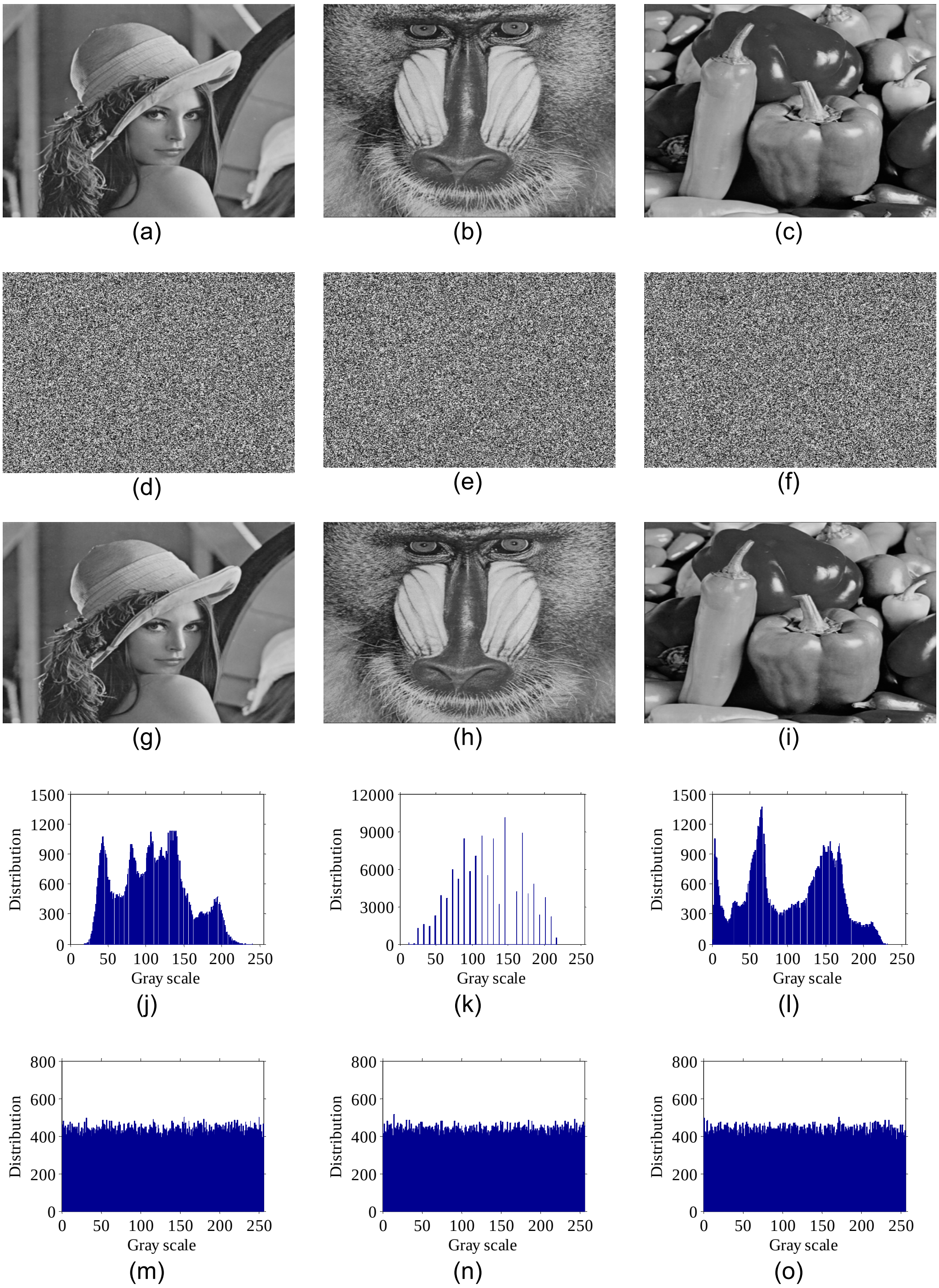

Figure 3. Simulation Results. (a)-(c) Plain images of Lena, Baboon and Pepper, respectively; (d)-(f) Cipher images of (a)-(c), respectively; (g)-(i) Decrypted images of (d)-(f), respectively; (j)-(l) Histograms of (a)-(c), respectively; (m)-(o) Histograms of (d)-(f), respectively

The computer used for simulation is configured with the Intel Duo Core I5 M460@2.53GHz, 2GB DDR3 RAM, Windows 7 and MATLAB 8.1. We employed the proposed system with the secret key of ***K***={3.3133, 12.0546, 40.8879, -34.5677, 35, 201}, to encrypt the plain images of Lena, Baboon and Pepper (all size of 357×317, as shown in Figure 3a-3c, respectively) to obtain their cipher images (as shown in Figs. 3d-3f, respectively). Then, we used the correct secret key ***K*** to decrypt the cipher images (as shown in Figure 3d-3f, respectively) to obtain the recovered images (as shown in Figure 3g-3i, respectively). The histograms of plain images Lena, Baboon and Pepper (as shown in Figure 3a-3c, respectively) are as shown in Figure 3j-3l, respectively. The histograms of cipher images (as shown in Figure 3d-3f, respectively) are as shown in Figure 3m-3o, respectively.

It can be seen from Figure 3 that: (1) The cipher images (as shown in Figure 3d-3f, respectively) are noise-like images without any leakage of visual information; (2) The decrypted images (as shown in Figure 3g-3i, respectively) are identical to the original plain images (as shown in Figure 3a-3c, respectively); (3) The cipher images have flat histograms (as shown in Figure 3m-3o), which is way different from those of plain images (as shown in Figure 3j-3l).

4. Security Analysis

For the image encryption system, the common used security evaluation rules are encryption/decryption speed, key space, statistical characters of cipher images, key sensitivity, plain image sensitivity (i.e. resisting the differential attacks), resisting the chosen/known plaintext attacks and information entropy, etc.

4.1. Encryption and Decryption Speed

Without loss of generality, we only consider the encryption/decryption speed of the image with the size of 357×317. When we use the proposed system to encrypt the plain images with the secret key ***K***={3.3133, 12.0546, 40.8879, -34.5677, 35, 201} (identical to the key used in Section 3), the time consumed for generating the secret code streams in Section 2.3.1 is T_1≈0.25985s, and the time consumed for diffusion-confusion-diffusion operations in Sections 2.3.2-2.3.4 is T_2≈0.09126s. So the time for encrypting one image is T_1+T_2≈0.35111s. When we use the proposed system to encrypt L pieces of images, the average time for encrypting each image is $T_1/L+T_2$, because the algorithm steps for generating the secret code streams only need to be executed once. If L=1000, the average time for encrypting each image is $T_1/L+T_2$≈0.09152s. Note that T_1 is slightly affected by r_1 and r_2. When $r_1=r_2$=255, T_1 takes its maximum of about 0.26125s, and when $r_1=r_2$=0, T_1 takes its minimum f about 0.25930s.

Table 1 lists the encryption/decryption time of our proposed system and that of [17]. It can be seen from Table 1 that for one image encrypting, the proposed system has slower encryption/decryption speed than the system in [17]; while for 1000 pieces of images encrypting, the proposed system has about triple speed than the system in [17].

Table 1. Comparison Results of Encryption/Decryption Time (s)

Encryption scheme	Time for one plain image		Average time for 1000 plain images	
	Encryption	Decryption	Encryption	Decryption
Proposed	0.35111	0.34277	0.09152	0.08318
Ref. [17]	0.23725	0.26294	0.23725	0.26294

4.2. Key Space

The secret key of the proposed system is ***K***={x_0, y_0, z_0, w_0, r_1, r_2}, where, the range values of x_0, y_0, z_0 and w_0 are separately x_0є(-40,40), y_0є(-40,40), z_0є(1,81), w_0є(-250,250); the step size of x_0, y_0 or z_0 is 10^{-13}, and the step size of w_0 is 10^{-12}; r_1 and r_2 are integers range in [0,255] with the step size of 1. So the key space size of the proposed system is about 1.6777×10^{64}, equivalent to the length of key being 213 bits. If we use the computer in Section 3 to perform the exhaustive attack, it will take us an average of about 9.1177×10^{55} years to crack the encryption system. Therefore, the proposed system can resist the brute-force attacks.

4.3. Statistical Characters of Cipher Images

Generally, we can investigate the encryption system against the statistical attacks from two aspects: One is whether the histograms of cipher images are flat; the other is whether the adjacent pixels in the cipher images have strong correlations. Without loss of generality, we take the cipher images shown in Figure 3d-3f as examples. From Figure 3m-3o, we can see that these cipher images have flat histograms. So the following will mainly discuss the correlations of adjacent pixels in the cipher images.

Assuming that we randomly choose *N* pairs of adjacent pixels, denoted by (x_i,y_i), i=1,2,…,*N* (Labeling $\boldsymbol{x}=\{x_i\}$, $\boldsymbol{y}=\{y_i\}$). Then the correlation coefficient r_{xy} between $\boldsymbol{x}$ and $\boldsymbol{y}$ can be calculated using the following formulas:

$$r_{xy} = \frac{\mathrm{cov}(\boldsymbol{x},\boldsymbol{y})}{\sqrt{D(\boldsymbol{x})}\sqrt{D(\boldsymbol{y})}} \tag{16}$$

$$\mathrm{cov}(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(\boldsymbol{x}))(y_i - E(\boldsymbol{y})) \tag{17}$$

$$D(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(\boldsymbol{x}))^2 \tag{18}$$

$$E(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}x_i \tag{19}$$

Where, cov($\boldsymbol{x}$,$\boldsymbol{y}$) represents the covariance of vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, $D(\boldsymbol{x})$ represents the variance of $\boldsymbol{x}$, $E(\boldsymbol{x})$ represents the mean value of $\boldsymbol{x}$, and *N* denotes the length of $\boldsymbol{x}$.

Table 2. Correlation Coefficients of the Plain and Cipher Images

	Horizontal		Vertical		Diagonal	
	Fig. 3a	Fig. 3d	Fig. 3b	Fig. 3e	Fig. 3c	Fig. 3f
Lena	0.952573	-0.004649	0.971203	-0.051114	0.920942	-0.016763
Baboon	0.884423	-0.023753	0.760920	-0.008060	0.722701	-0.036221
Pepper	0.981947	-0.003522	0.980936	0.013633	0.958665	0.007701

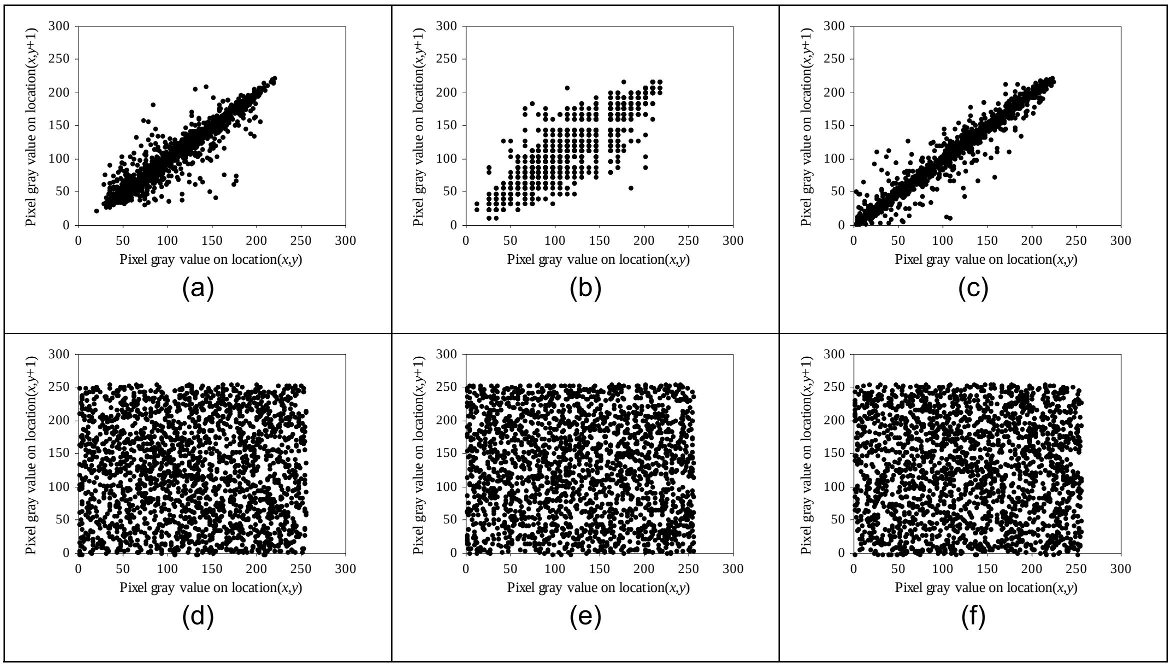

Figure 4. Results of Correlation Analysis; (a)-(c) Correlations in horizontal direction for the plain images of Lena, Baboon and Pepper, respectively; (d)-(f) Correlations in horizontal direction for the cipher images of Lena, Baboon and Pepper, respectively

Here, we take N=2000, and calculate the correlation coefficients of adjacent pixels of plain images (as shown in Figure 3a-3c, respectively) and cipher images (as shown in Figure 3d-3f, respectively) in the horizontal, vertical and diagonal directions. The calculated results are listed in Table 2. Meanwhile, the correlations in the horizontal direction for Figure 3a-3f are illustrated in Figure 4.

As apparent from Table 2 and Figure 4, the adjacent pixels in plain images have strong correlations, and their correlation coefficients are close to 1; while the adjacent pixels in cipher images hardly have correlations, which is close to 0. These demonstrate that the proposed system can fight against the statistical attacks.

4.4. Key Sensitivity Analysis

As for the secret key $\boldsymbol{K}_1=\{x_0, y_0, z_0, w_0, r_1, r_2\}$, any element of $\{x_0, y_0, z_0\}$ changes its value by 10^{-13}, or w_0 changes its value by 10^{-12}, or any element of $\{r_1, r_2\}$ changes its value by 1, to get the new secret key denoted by $\boldsymbol{K}_2$. The sensitivities of the secret key will be analyzed in the following two aspects:

(1) Use the proposed system with the secret keys of $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ to encrypt the plain image $\boldsymbol{P}$ to obtain two cipher images, denoted by $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, respectively. Compare the images of $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ to obtain two indexes, denoted by $Diff_1$ and $Diff_2$ respectively, by using the following formulas:

$$Diff_1=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|\mathrm{Sign}(\boldsymbol{C}_1(i,j)-\boldsymbol{C}_2(i,j))|\times 100\% \quad (20)$$

$$Diff_2=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|\boldsymbol{C}_1(i,j)-\boldsymbol{C}_2(i,j)|}{256}\times 100\% \quad (21)$$

Where, Sign(·) is the sign function. M and N represent the height and width of the image, respectively (hereinafter the same meaning).

For two random noise images, the theoretical values of $Diff_1$ and $Diff_2$ are 255/256≈99.6094% and 21845/65536≈33.3328%, respectively [17].

(2) Use the proposed system, encrypt the plain image $\boldsymbol{P}_1$ with secret key $\boldsymbol{K}_1$ to get cipher image $\boldsymbol{C}$, then decrypt the image $\boldsymbol{C}$ with secret key $\boldsymbol{K}_2$ to get the decrypted image, denoted by $\boldsymbol{P}_2$. Compare images $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ to obtain two indexes, denoted by $Diff_3$ and $Diff_4$ respectively, by using the following formulas:

$$Diff_3=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|\mathrm{Sign}(\boldsymbol{P}_1(i,j)-\boldsymbol{P}_2(i,j))|\times 100\% \quad (22)$$

$$Diff_4=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|\boldsymbol{P}_1(i,j)-\boldsymbol{P}_2(i,j)|}{256}\times 100\% \quad (23)$$

If $\boldsymbol{P}_1$ takes the plain images of Lena, Baboon and Pepper (as shown in Figure 3a-3c, respectively), and $\boldsymbol{P}_2$ is a random noise image, the theoretical values of $Diff_3$ are all 255/256≈99.6094%, and the theoretical values of $Diff_4$ are about 28.6105%, 28.2020% and 30.8265%, respectively [17].

Here, take the plain images of Lena, Baboon and Pepper (as shown in Figure 3a-3c, respectively) as examples, and do the trials for 100 times for each image to test the secret key sensitivities. Then list the calculated average values of $Diff_1$, $Diff_2$, $Diff_3$ and $Diff_4$ in Table 3 (where, the values in parentheses are theoretical values for each index). In each trial, use the following formulas to generate the random secret key of $\boldsymbol{K}_1=\{x_0, y_0, z_0, w_0, r_1, r_2\}$:

$$x_0=-40+80\times rand \quad (24)$$

$$y_0=-40+80\times rand \quad (25)$$

$$z_0=1+80\times rand \quad (26)$$

$$w_0=-250+500\times rand \quad (27)$$

$$r_1=(\mathrm{floor}(rand\times 100000)) \bmod 256 \quad (28)$$

$$r_2=(\text{floor}(rand\times100000)) \bmod 256 \quad (29)$$

Where, *rand* is a MATLAB function used to generate the 0-1 uniformly distributed random numbers, and floor(*t*) returns the integer less than or equal to *t*.

Table 3. Results of Key Sensitivity Tests

	$Diff_1$(99.6094%)	$Diff_2$(33.3328%)	$Diff_3$(99.6094%)	$Diff_4$
Lena	99.6096%	33.3390%	99.6109%	28.6173% (28.6105%)
Baboon	99.6053%	33.3368%	99.6072%	28.1998% (28.2020%)
Pepper	99.6109%	33.3262%	99.6116%	30.8287% (30.8265%)

As can be seen from Table 3, the calculated values of $Diff_1$, $Diff_2$, $Diff_3$ and $Diff_4$ are very close to their theoretical values, indicating that the proposed encryption system is very sensitive to the secret keys. These also show that each key in the key space is valid.

4.5. Resisting the Differential Attack

NPCR (number of pixels change rate) and UACI (unified average changing intensity) are usually used to measure the ability of encryption system to fight against the differential attacks [3]. Assume that the plain images $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ are identical except that $\boldsymbol{P}_2(i,j)=(\boldsymbol{P}_1(i,j)+1)$ mod 256 for a certain coordinate (i,j). We use the proposed encryption system to encrypt the plain images $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ with the same secret key to get two cipher images, denoted by $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, respectively. Then the definitions of NPCR and UACI can be expressed as follows:

$$\text{NPCR}=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|\text{Sign}(\boldsymbol{C}_1(i,j)-\boldsymbol{C}_2(i,j))|\times100\% \quad (30)$$

$$\text{UACI}=\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|\boldsymbol{C}_1(i,j)-\boldsymbol{C}_2(i,j)|}{255}\times100\% \quad (31)$$

For two random noise images, the theoretical values of NPCR and UACI are 255/256≈99.6094% and 257/768≈33.4635%, respectively [17].

Here, consider the plain images of Lean, Baboon and Pepper all with the size of 128×128, 256×256, 357×317 and 512×512. For each image, do 100 trials to calculate the average values of NPCR and UACI, and then list the results in Table 4. Note that a random secret key is generated by Equation (24)-(29) for encrypting each image.

Table 4. Results of Plaintext Sensitivity Tests

Plain image size	NPCR (99.6094%)			UACI (33.4635%)		
	Lena	Baboon	Pepper	Lena	Baboon	Pepper
128×128	99.6095%	99.6044%	99.6089%	33.4528%	33.4863%	33.4122%
256×256	99.6101%	99.6112%	99.6077%	33.4546%	33.4316%	33.4894%
357×317	99.6101%	99.6082%	99.6103%	33.4927%	33.4551%	33.4979%
512×512	99.6108%	99.6099%	99.6090%	33.4679%	33.4755%	33.4807%

As can be seen from Table 4, the test values of NPCR and UACI are very close to the theoretical values of NPCR and UACI, indicating that the proposed system can fight against the differential attacks effectively.

4.6. Resisting Chosen/known Plaintext Attacks

For different plain images in a certain image encryption system, if the equivalent secret keys or part of the equivalent secret keys are kept unchanged, the attacker can crypt-analyze the equivalent keys of the encryption system via choosing or acquiring multi-pairs of plain and cipher images [12-16]. In order to resist the chosen/known plaintext attacks, it is necessary that different plain images correspond to different equivalent secret keys in the encryption system. In our proposed image encryption system, although the diffusion is plaintext-unrelated, the confusion is plaintext-related, so that the different plain images correspond to different equivalent keys. Therefore, the proposed system can resist the chosen/known plaintext attacks.

4.7. Information Entropy

The information entropy reflects the uncertainty of the image information. The bigger the information entropy is, the more uncertain the image information is, and the more unintelligible the image is. For the L-level grayscale image, denote the emergence probability of gray value i by $p(m_i)$, and then the information entropy can be expressed as the following formula:

$$H(m) = -\sum_{i=0}^{L-1} p(m_i)\log_2(p(m_i)) \tag{32}$$

For an 8-bit random noise grayscale image, the theoretical value of information entropy is 8.

Here, L=256. We use the plain images of Lena, Baboon and Pepper all of size 128×128, 256×256, 357×317 and 512×512 and their corresponding cipher images to calculate the information entropy values, and list the results in Table 5. Without loss of generality, the secret key ***K*** used is fixed on {3.3133, 12.0546, 40.8879, -34.5677, 35, 201} (same as the key used in Section 3).

Table 5. Results of Information Entropy Tests

Image size	Entropy of Lena		Entropy of Baboon		Entropy of Pepper	
	Plain	Cipher	Plain	Cipher	Plain	Cipher
128×128	7.34907	7.99680	7.25776	7.99646	7.58134	7.99617
256×256	7.36839	7.99903	7.35535	7.99919	7.57459	7.99915
357×317	7.37914	7.99823	4.35025	7.99855	7.56647	7.99840
512×512	7.38398	7.99929	7.45159	7.99932	7.57076	7.99922

As can be seen from Figure 5, the information entropy values of plain images are way different from the theoretical value; while the information entropy values of cipher images are close to the theoretical value. These demonstrate that the cipher images have no information leakage. Consequently, the proposed system can resist the various information entropy attacks.

5. Conclusion

This paper proposes a new plaintext-related image encryption system, which includes plaintext-unrelated image diffusion and plaintext-related image confusion. For multi-images encryption, the proposed image encryption system is much faster than the traditional plaintext-related encryption system on encryption speed due to no round operations needed in its encryption process. Because the confusion is plaintext-related, different plain images correspond to different equivalent keys, so the proposed system can resist the chosen/known plaintext attacks. Finally, the simulation results show that the proposed system possesses the characters of huge key space, strong key sensitivity, strong plaintext sensitivity, good statistical properties of the cipher images, large information entropy, etc. Therefore, the proposed image encryption system can be applied in actual communications.

Acknowledgements

This work was fully supported by the Natural Science Foundation of Jiangxi Province (Grant No. 20122BAB201036).

References

[1] Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. *Int J Bifurc Chaos*. 1998; 8(6): 1259-1284.

[2] Lian SG, Sun JS, Wang ZQ. A block cipher based on a suitable use of the chaotic standard map. *Chaos Solitons Fractals*. 2005; 26(1): 117-129.

[3] Chen GR, Mao YB, Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals*. 2004; 21(3): 749-761.

[4] Ye GD. Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recognition Letters*. 2010; 31(5): 347-354.

[5] Patidar V, Pareek NK, Purohit G, Sud KK. A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption. *Optics Communications*. 2011; 284(19): 4331-4339.

[6] Liu LL, Zhang Q, Wei X. A RGB image encryption based on DNA encoding and chaos map. *Computers and Electrical Engineering*. 2012; 38(2): 1240-1248.

[7] Armand Eyebe Fouda JS, Yves Effa J, Sabat SL, Ali M. A fast chaotic block cipher for image encryption. *Commun Nonlinear Sci Numer Simulat*. 2014; 19(3): 578-588.

[8] Zhang Q, Guo L, Wei X. A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. *Optik*. 2013; 124(18): 3596-3600.

[9] Pareek NK, Patidar V, Sud KK. Image encryption using chaotic logistic map. *Image and Vision Computing*. 2006; 24(9): 926-934.

[10] Wang XY, Yang L, Liu R, Kadir A. A chaotic image encryption algorithm based on perceptron model. *Nonlinear Dyn*. 2010; 62(3): 615–621.

[11] Huang CK, Nien HH. Multi chaotic systems based pixel shuffle for image encryption. *Optics Communications*. 2009; 282(11): 2123-2127.

[12] Zhang Y, Wen W, Su M, Li M. Cryptanalyzing a novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system. *Optik*. 2014; 125(4): 1562-1564.

[13] Wang XY, Chen F, Wang T, Xu D, Ma Y. Attack to an image encryption based on chaotic Logistic map. *Int J Modern Physics B*. 2013; 27(31): 1350196-1~9.

[14] Zhang Y, Li CQ, Li Q, Zhang D, Shu S. Breaking a chaotic image encryption algorithm based on perceptron model. *Nonlinear Dyn*. 2012; 69(3):1091-1096.

[15] Solak E, Rhouma R, Belghith S. Cryptanalysis of a multi-chaotic systems based image cryptosystem. *Optics Communications*. 2010; 283(2): 232-236.

[16] Zhang Y. Cryptanalysis of an image encryption algorithm based on chaotic modulation of Arnold dual scrambling and DNA computing. Advanced Science Focus, 2014; 2(1): 1-16.

[17] Zhang Y. Plaintext related image encryption scheme using chaotic map. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(1): 635-643.

[18] Eslami Z, Bakhshandeh A. An improvement over an image encryption method based on total shuffling. *Optics Communications*. 2013; 286(1): 51-55.

[19] Zhang G, Liu Q. A novel image encryption method based on total shuffling scheme. *Optics Communications*. 2011; 284(12): 2775-2780.

[20] Ye R. A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism. *Optics Communications*. 2011; 284(22): 5290-5298.

[21] Ye GD, Wong KW. An efficient chaotic image encryption algorithm based on a generalized Arnold map. *Nonlinear Dynamics*. 2012; 69(4): 2079-2087.

[22] Abd El-Latif AA, Li L, Zhang T, Wang N, Song X, Niu X. Digital image encryption scheme based on multiple chaotic systems. *Sens Imaging*. 2012; 13(2): 67-88.

[23] Zhang Y, Xia JL, Cai P, Chen B. Plaintext related two-level secret key image encryption scheme. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2012; 10(6): 1254-1262.