TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.1, January 2014, pp. 635 ~ 643
DOI: http://dx.doi.org/10.11591/telkomnika.v12i1.3482

Plaintext Related Image Encryption Scheme Using Chaotic Map

Yong Zhang
School of Software and Communication Engineering, Jiangxi University of Finance and Economics
Nanchang, PR China, Ph./Fax: +086-15270015009/79183845702
e-mail: zhangyong@jxufe.edu.cn

Abstract

A plaintext related image blocking encryption algorithm is proposed in this paper, which includes two kinds of operations on inner-block confusion and inter-block diffusion. Firstly, a float-point lookup table need to be generated by iterating chaotic system with the secret keys as the initial values and parameters; Secondly, choose one of the entries in the look-up table according to the pixel value derived from the plain-image as initial value of chaotic system, and iterate it to produce one secret code sequence for inner-block confusion; Thirdly, by using one pixel value of the former block to locate another entry in the look-up table, which to be employed as the new initial value of the chaotic system, iterate it to yield another secret code sequence for inter-block diffusion; Finally, through two rounds of the block-by-block processes, the plain-image will be transformed into the cipher-image. The simulation results show that the proposed method has good characters, such as large key space, fast encryption speed, strong key sensitivity, and high security against the brute-force attack and the chosen plaintext attack, etc.

Keywords*: image encryption, chaotic map, plaintext related cipher, cryptanalysis*

1. Introduction

A great deal of research on chaotic system based image encryption technology has sprung up since 1989 [1-9] and several of them have been challenged since 2003. For example, the encryption algorithms proposed in [4-6] have been crypt-analyzed in [7-9] by using chosen plaintext attack or other attack methods. In these research works, the main reason for low security is that the secret code streams used to encrypt the plain-image are only relied on the secret key but bear no relation to the plain-image. Aware of this deficiency, an encryption scheme which uses the size of plain-image as part of secret keys was proposed [10], but this method has defect in against time attack; Another scheme introduced the hash code of plain-image as part of the secret keys [11]. It has high security but needs an extra private secret channel to transmit the hash code which increases the burden of communication.

Recently, Zhang et al. proposed an image encryption method based on total shuffling scheme [12]. This method is characterized in that the secret code stream used in encryption is not only associated with the key, but also the plain image. However, the first secret code in [12] is independent of plain image, and that make it is not safe enough to resist the chosen plaintext attack, this is pointed out and crypt-analyzed in [13]. Eslami et al. suggested an improved algorithm [14] over these shortcomings described in [13]. Two major improvements of it are to use previous cipher image pixels to execute “add modulus and xor” operations instead of plain image pixels, and to enlarge the iteration times of chaotic system in every round. That made the image encryption scheme proposed in [12] was higher security against the chosen plaintext attacks but slower the encryption speed as the cost. Similar to the scheme of [12], some plain image related image encryption methods were proposed [15-16], in which the information of plain image was used to determine the parameters and iteration times of the chaotic map.

More recently, Ahmed A. Abd El-Latif et al. presented a plain-image related image encryption scheme called BES-w/r/b, where ‘BES’ meant block encryption scheme, ‘w’ represented the number of bits in each pixel, ‘r’ indicated the number of rounds, and ‘b’ was the key length (in bytes) [17]. In this scheme, the four one-dimensional chaotic systems are used to encrypt the plain image, block by block. The shortness of this scheme is that the encryption speed will be very slow when the value of ‘r’ is large. Interestingly, Ahmed A. Abd El-Latif et al.

Received June 17, 2013; Revised July 17, 2013; Accepted August 13, 2013

seemed to have discovered the deficiency, and only gave the simulation results for BES-32/2/16 (r=2). But even with r=2, the speed of encryption is less optimistic and slower compared to the other four encryption schemes tabulated in [17].

In 2012, we suggested a plain image related image encryption method with two levels of secret keys [18]. This scheme can withdraw the existing passive attacks effectively, but the encryption speed need to be further improved. On the basis of these researches, a new chaotic system based image encryption method is proposed in this paper. Section 2 describes the chaotic system and the detailed image encryption scheme for this method. Section 3 provides some simulation results by using MATLAB to demonstrate the feasibility of the proposed method. Section 4 discusses the security performance of the proposed scheme center on the secret key space, statistical properties of the cipher image, NPCI and UACI, information entropy, and resisting chosen plaintext attack, etc.

2. Proposed Encryption Scheme

2.1. Used Chaotic System

The triangular map of discrete form shown in (1) is employed in the proposed scheme, wherein, a and b are its parameters. When the values of a and b belong to the interval [3.57, 4.00], Eq. (1) has chaotic attractor, and the state values of x_n and y_n range in (0,1) [19].

$$\begin{cases} x_{n+1} = ax_n(1 - x_n) \\ y_{n+1} = bx_n y_n(1 - x_n y_n) \end{cases} \quad (1)$$

2.2. Basic Principle of Encryption

The basic principle of the proposed encryption scheme is shown in Figure 1. Firstly, the secret key is regarded as the initial values and the parameters of the chaotic map. Iterate the chaotic system to obtain a floating-point form of look-up table (size of 256 entries) and two pseudo-random numbers which will be used as the initial values of the chaotic system. Iterate the chaotic system to generate the secret code streams for the first block of the plain image and yield the first block of the cipher image; Secondly, after obtaining the cipher image block n-1, two new random numbers are generated by continuing iterating the chaotic map, and are transformed into integers to locate the pixel in block n of the plain image. The value of the pixel is used to search the look-up table for the entry with corresponding value, and the entry's content serves as the new initial values of the chaotic map which is iterated to generate the random numbers for encrypting the plain image block n; Thirdly, the random numbers generated in the previous step are employed to encrypt the plain image block n to produce the cipher image block n, which will be cycled by 2 rounds. Finally, the whole encryption process is cycled by 2 rounds from the plain image by the secret key transforming to the cipher image, and the decryption process is the reverse of the encryption one.

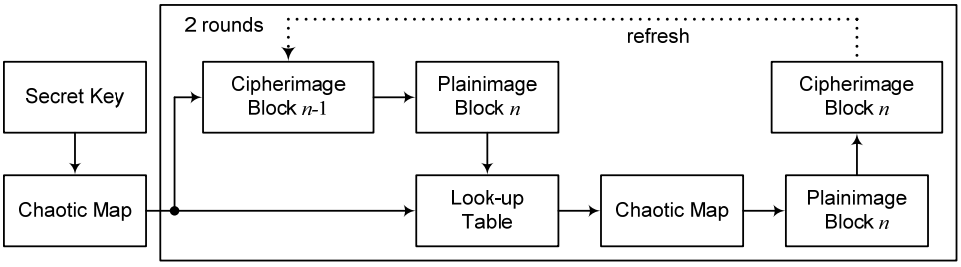

Figure 1. Basic principle diagram of encryption scheme

2.3. Encryption Scheme

Encryption scheme encrypts the plain image into the noise-like cipher image, and therein the plain image and the secret key are inputs, while the cipher image and the encryption time are outputs. The detailed steps are as follows:

(1) Given the secret key K={k_1, k_2, k_3, k_4}, 0<k_i<1, i=1, 2, 3, 4, the parameters of Eq. (1) are a=3.57+0.43×k_1, b=3.57+0.43×k_2, and its initial values are x_0=k_3, y_0=k_4. Then, iterate Eq. (1) to generate a sequence {(x_i,y_i),i=1,2,…,132}, wherein, {(x_i,y_i),i=1,2,…,128} is used as the look-up table (named by LT) of size 256, in which each entry is noted by e_i, i=0,1,2,…,255, satisfying e_{2i-2}=x_i, e_{2i-1}=y_i, i=1,2,…,128.

(2) Suppose that the grayscale plain image $\boldsymbol{P}$ is of size m=M×N. Divide $\boldsymbol{P}$ into u=$\lceil m/16 \rceil$ blocks, wherein $\lceil x \rceil$ represents the smallest integer value of not less than x (thereinafter the same meaning). Each block contains 16 pixels, and if the last block has less than 16 pixels, the block should be padded to 16 pixels with zeros. Denote each block by P_i, i=0,1,…,u-1, and P_i=$p_{i,0}p_{i,1}p_{i,2}$,…,$p_{i,15}$, then p_{ij} represents the value of the j-th pixel in the i-th block, i=0,1,…,u-1, j=0,1,2,…,15. Therefore, the plain image $\boldsymbol{P}$=P_0,P_1,…,P_{u-1}.

(3) Transform (x_{129},y_{129}) into two integers k_1 and l_1 by using k_1=$\lceil x_{129} \times 10^6 \rceil$ mod 16 and l_1=$\lceil y_{129} \times 10^8 \rceil$ mod 256, then locate the k_1-th pixel in the block P_0, whose value is p_{0,k_1}. Let $d_1 = (p_{0,k_1} + l_1)$ mod 256, then locate the d_1-th entry in the LT, whose value is e_{d_1}. And refresh x_{129} with x_{129}=(former x_{129}+e_{d_1}) mod 1.

(4) Use values of (x_{129},y_{129}) as the initial values of Eq. (1), and iterate the chaotic map to generate 9 pairs of values, denoted by ($x_{0,i}^f$,$y_{0,i}^f$), i=0,1,2,…,8, wherein the former eight pairs of values are arranged into a vector, denoted by h_0, h_1, h_2,..., h_{15}, satisfying h_0=$x_{0,0}^f$, h_1=$y_{0,0}^f$, h_2=$x_{0,1}^f$, …, h_{15}=$y_{0,7}^f$. Each h_i is converted into the integer g_i with formulation g_i=$\lceil h_i \times 10^8 \rceil$ mod 256, i=0,1,2,…,15, and the later are used to do "add and modulus" operations with the 15 pixels of block P_0 (except the k_1-th position of pixel p_{0,k_1}) by using the following formula.

$$\begin{cases} q_{0,0} = (p_{0,0} + g_0) \bmod 256 \\ q_{0,i} = (p_{0,i} + g_i + q_{0,i-1}) \bmod 256, 1 \le i \le k_1 - 1 \\ q_{0,k_1} = p_{0,k_1} \\ q_{0,(k_1+1) \bmod 16} = (p_{0,(k_1+1) \bmod 16} + g_{(k_1+1) \bmod 16} + q_{0,(k_1-1) \bmod 16}) \bmod 256 \\ q_{0,i} = (p_{0,i} + g_i + q_{0,i-1}) \bmod 256, \ \ k_1 + 2 \le i \le 15 \end{cases} \quad (2)$$

Therefore, the so-called block Q_0=$q_{0,0}q_{0,1}q_{0,2}\ldots q_{0,15}$ is obtained.

(5) Convert (x_{130}, y_{130}) into two integers k_2 and l_2 by using k_2=$\lceil x_{130} \times 10^6 \rceil$ mod 16 and l_2=$\lceil y_{130} \times 10^8 \rceil$ mod 256. If k_1=k_2, then k_2=(k_2+1) mod 16. Then locate the k_2-th pixel in the block Q_0, whose value is q_{0,k_2}. Let d_2=(q_{0,k_2}+l_2) mod 256, then locate the d_2-th entry in the LT, whose value is e_{d_2}. And refresh x_{130} with x_{130}=(former x_{130}+ e_{d_2}) mod 1.

(6) The values of (x_{130}, y_{130}) are used as the initial values of Eq. (1) and, iterate the chaotic map to generate 9 pairs of values, denoted by ($x_{0,i}^b$,$y_{0,i}^b$), i=0,1,2,…,8, wherein the former eight pairs of values are arranged into a vector, denoted by w_0, w_1, ..., w_{15}, satisfying w_0=$x_{0,0}^b$, w_1=$y_{0,0}^b$, w_2=$x_{0,1}^b$, …, w_{15}=$y_{0,7}^b$. Each w_i is converted into the integer v_i with formulation v_i=$\lceil w_i \times 10^8 \rceil$ mod 256, i=0,1,2,…,15, and the later are used to do "add and modulus" operations with the 15 pixels of block Q_0 (except the k_2-th position of pixel q_{0,k_2}) by using the following formula.

$$\begin{cases} r_{0,15} = (q_{0,15} + v_{15}) \bmod 256 \\ r_{0,i} = (q_{0,i} + v_i + r_{0,i+1}) \bmod 256, 14 \ge i \ge k_2 + 1 \\ r_{0,k_2} = q_{0,k_2} \\ r_{0,(k_2-1) \bmod 16} = (q_{0,(k_2-1) \bmod 16} + v_{(k_2-1) \bmod 16} + r_{0,(k_2+1) \bmod 16}) \bmod 256 \\ r_{0,i} = (q_{0,i} + v_i + r_{0,i+1}) \bmod 256, \ \ k_2 - 2 \ge i \ge 0 \end{cases} \quad (3)$$

Therefore, the so-called block R_0=$r_{0,0}$,$r_{0,1}$,$r_{0,2}$,...,$r_{0,15}$ is obtained.

(7) Derive integer l_3 from $y_{0,8}^f$ by using l_3=$\lceil y_{0,8}^f \times 10^8 \rceil$ mod 256, and let d_3=($q_{0,15}$+l_3) mod 256, locate the d_3-th entry in the LT, whose value is e_{d_3}. Refresh $x_{0,8}^f$ with $x_{0,8}^f$ =(former $x_{0,8}^f$ + e_{d_3}) mod 1, then use the values of ($x_{0,8}^f$, $y_{0,8}^f$) as the initial values of Eq. (1), and iterate the chaotic map once to generate a pair of state values as new (x_{129},y_{129}).

(8) Derive an integer l_4 from $y_{0,8}^b$ by using $l_4=\lceil y_{0,8}^b \times 10^8 \rceil$ mod 256, and let $d_4=(r_{0,0}+l_4)$ mod 256, then locate the d_4-th entry in the LT, whose value is e_{d_4}. Refresh $x_{0,8}^b$ with $x_{0,8}^b$=(former $x_{0,8}^b+ e_{d_4}$) mod 1, then the values of ($x_{0,8}^b$, $y_{0,8}^b$) are used as the initial values of Eq. (1), and iterate the chaotic map once to generate a pair of state values as new (x_{130}, y_{130}).

(9) Replace P_0 with P_1, and repeat steps (3-6) to obtain the encryption block R_1 of the plain image block P_1. Then repeat steps (7-8) to produce new iterative initial values (x_{129}, y_{129}) and (x_{130}, y_{130}).

(10) Repeat steps (7-9) for u-2 times, and replace P_0 with P_i (i=2,3,...,u-1) in step (9) every time, sequentially to produce the encryption block R_i (i=2,3,...,u-1) for block P_i.

(11) Let $\boldsymbol{R}$=R_{u-1},R_{u-2},...,R_1,R_0, substitute $\boldsymbol{P}$ with $\boldsymbol{R}$, also replace (x_{129}, y_{129}) and (x_{130}, y_{130}) with (x_{131}, y_{131}) and (x_{132}, y_{132}), respectively. Then repeat steps (3-10) to get the final cipher image $\boldsymbol{C}$=C_0,C_1,...,C_{u-1}.

Decryption scheme is the inverse of the encryption scheme, wherein the inputs are the cipher image and the identical secret key, and the outputs are the original plain image and the decryption time.

3. Simulate Results

A large number of experiments are carried successfully with the proposed scheme. The followings are the results of encrypting the images (Lena and Baboon) as examples, as shown in Figure 2. The secret keys used are {0.9767, 0.8834, 0.7834, 0.3401}, and the corresponding parameters and the initial values of Eq. (1) are a=3.989981, b=3.949862, x_0=0.7834, y_0=0.3401. From Figure 2, we can see that the cipher images are visually noise-like, and the decrypted images are identical to the original images.

Figure 2. The results of Lena and Baboon. (a) Plain image; (b) Cipher image of (a); (c) Decrypted image; (d) Plain image; (e) Cipher image of (d); (f) Decrypted image.

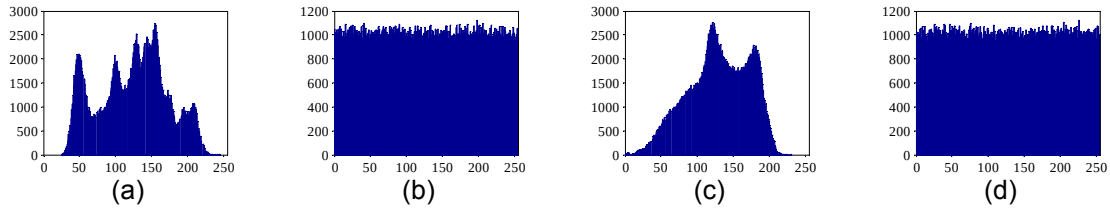

Figure 3. Histograms of Lena and Baboon. (a) Histogram of Figure 2a; (b) Histogram of Figure 2b; (c) Histogram of Figure 2d; (d) Histogram of Figure 2e.

4. Security Performance Analysis

4.1. Key Space

Secret keys {k_1, k_2, k_3, k_4} range in (0,1), where, k_1, k_2 and k_3 are correct to 14 decimal places, while k_4 is correct to 13 decimal places, so the size of the key space is approximately 10^{55} (equivalent to the key of 183-bit binary numbers). Meanwhile the look-up table of length 256 can be regarded as the equivalent key against differential attack, and each of the entries is correct to 14 decimal numbers, so the size of the equivalent key space is about 10^{3584}. Therefore the size of the key space is large enough to confront the brute-force attacks.

4.2. Histogram

Take the histograms of the images of Lena and Baboon as examples, shown in Figure 3. From Figure 3, it can be seen that the histogram of the cipher image is completely different from that of the plain image. Meanwhile the histogram of the cipher image is flat and close to the histogram of the noise-like image, which can resist the statistical attacks effectively.

4.3. Correlation Analysis

Take the images of Lena and Baboon as examples to calculate the horizontal, vertical and diagonal correlation coefficients [3], and tabulate them in Table 1. Here illustrate only horizontal correlations of Lena and its cipher image in Figure 4 to save space. T=10000 in both Table 1 and Figure 4. From Figure 4 and Table 1, it can be seen that the adjacent pixels in plain image are highly relevant, while the adjacent pixels in cipher image are nearly irrelevant.

Table 1. Correlation coefficients

	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena (Fig.2a-b)	0.9720	0.9846	0.9624	−0.0127	0.0024	0.0032
Baboon (Fig. 2d-e)	0.8685	0.7719	0.7235	−0.0090	−0.0069	0.0115

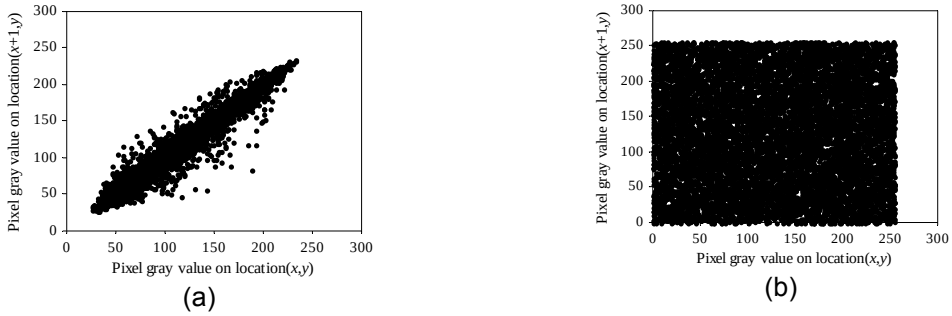

Figure 4. Horizontal correlations of Lena and its cipher image.
(a) Horizontal correlation of Figure 2a; (b) Horizontal correlation of Figure 2b.

4.4. Encryption Speed

The machine used is equipped with Intel I5 M460 processor, 2GB memory and MATLAB 7. 1000 pieces of images (all of size 512×512) were chosen to execute the proposed scheme. The average encryption or decryption speeds are 0.3776 and 0.3772 seconds separately. Under the same conditions, the average encryption and decryption speeds are about 1.5646 and 0.7433 seconds separately in [2], which are even faster than the conventional AES. So the proposed is much faster and can be used in the practical communication.

4.5. Information Entropy

As well known, the theoretical value of information entropy for 8-bit random image is exactly 8. However the entropy values of Lena and Baboon are about 7.4451 and 7.3583, respectively. The entropy values of their cipher images illustrated in Figure 2b and 2d are about 7.9992 and 7.9993, respectively, which are close to the theoretical value of 8. Therefore there is no information leakage in the cipher image.

4.6. Key Sensitivity

4.6.1. Theoretical Values of Sensitivity Indicators

The three indicators NPCR, UACI and NMSE are used to demonstrate the key sensitivity of the proposed scheme [3]. Supposing that the cipher images $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ (both of size $M\times N$) are obtained by encrypting same plain image with two secret keys of only 1-bit different, or encrypting two plain images of only 1-pixel different with the same key. Introduce a matrix $\boldsymbol{D}$ of size $M\times N$. If $C_1(i,j)=C_2(i,j)$, then $D(i,j)=0$, else $D(i,j)=1$. The NPCR, UACI and NMSE are defined as follows.

$$\mathrm{NPCR} = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} D(i,j)}{M\times N} \times 100\% \quad (4)$$

$$\mathrm{UACI} = \frac{1}{M\times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\frac{|C_1(i,j)-C_2(i,j)|}{255} \times 100\% \quad (5)$$

$$\mathrm{NMSE} = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[C_1(i,j)-C_2(i,j)]^2}{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} C_1^2(i,j)} \times 100\% \quad (6)$$

(1) Theoretical value of NPCR between two random noise images
As for two random images of 8-bit grayscale, the probability distribution is

$$D(i,j) = \begin{cases} 0, C_1(i,j) = C_2(i,j), p_0 = 1/256 \\ 1, C_1(i,j) \neq C_2(i,j), p_1 = 1 - 1/256 \end{cases} \quad (7)$$

Therefore the expected value of NPCR is $\mathrm{E[NPCR]} = \frac{\{M\times N\times[0\times p_0+1\times p_1]\}}{\{M\times N\}} = \frac{255}{256} \approx 99.6094\%$.

(2) Theoretical value of UACI between two random noise images
As for two random images of 8-bit grayscale, the expected valued of UACI is

$$\mathrm{E[UACI]} = \frac{1}{M\times N}\mathrm{E}\left[\sum_{i,j}\frac{|C_1(i,j)-C_2(i,j)|}{255}\right] \times 100\% = \frac{1}{M\times N}\frac{1}{255}\mathrm{E}\left[\sum_{i,j}|C_1(i,j) - C_2(i,j)|\right] \quad (8)$$

The values and frequencies of $C_1(i,j)$-$C_2(i,j)$ are tabulated in Table 2.

Table 2. Values and frequencies of $C_1(i,j)$-$C_2(i,j)$

Value	-255	-254	-253	…	-2	-1	0	1	2	…	253	254	255
Frequency	1	2	3	…	254	255	256	255	254	…	3	2	1

So $|C_1(i,j)$-$C_2(i,j)|$ is expected to be [2×(255×1+254×2+253×3+...+2×254+1×255) + 0×256] / (256×256) = 5592320/65536. Therefore, E[UACI] = 5592320/65536/255 = 257/768 ≈ 33.4635%.

(3) Theoretical value of NMSE between two random noise images
As for two random images $\boldsymbol{C_1}$ and $\boldsymbol{C_2}$ of 8-bit grayscale, the expected valued of UACI is

$$\mathrm{E[NMSE]} = \frac{\mathrm{E}[(\mathrm{C_1}-\mathrm{C_2})^2]}{\mathrm{E}[\mathrm{C_1^2}]} = \frac{\mathrm{E}[\mathrm{C_1^2}]+\mathrm{E}[\mathrm{C_2^2}]-2\mathrm{E}[\mathrm{C_1}\cdot\mathrm{C_2}]}{\mathrm{var}(\mathrm{C_1})+\{\mathrm{E}[\mathrm{C_1}]\}^2} = \frac{2\mathrm{E}[\mathrm{C_1^2}]-2\{\mathrm{E}[\mathrm{C_1}]\}^2}{\mathrm{var}(\mathrm{C_1})+\{\mathrm{E}[\mathrm{C_1}]\}^2}$$
$$= 2\frac{\mathrm{var}(\mathrm{C_1})}{\mathrm{var}(\mathrm{C_1})+\{\mathrm{E}[\mathrm{C_1}]\}^2} = \frac{2}{1+\frac{\{\mathrm{E}[\mathrm{C_1}]\}^2}{\mathrm{var}(\mathrm{C_1})}} = \frac{2}{1+\frac{[(255+0)/2]^2}{(255-0)^2/12}} = \frac{1}{2} = 50\%$$

(4) Theoretical value of NPCR between a random image and a deterministic image
As for 8-bit grayscale images, the expected value of NPCR is the same as the case with two random images. Therefore, $\mathrm{E[NPCR]} = \frac{M\times N\times[0\times p_0+1\times p_1]}{M\times N} = \frac{255}{256} \approx 99.6094\%$.

(5) Theoretical value of UACI between a random image and a deterministic image
The values of UACI are different for specific images. For examples, the value of UACI between Lena and random image is 28.6242%, while the value of UACI between Baboon and random image is 27.8472%.

(6) Theoretical value of NMSE between a random image and a deterministic image
The values of NMSE are different for specific images. For examples, the value of NMSE between Lena and random image is 35.6210%, while the value of NMSE between Baboon and random image is 33.2810%.

4.6.2. Key Sensitivity Analysis

Take the images Lena and Baboon of 8-bit grayscale as examples to calculate the NPCR, UACI and NMSE indicators with four situations:

For encryption: (1) Generate two set of cipher images from the same plain image with two set of secret keys. For the two set of secret keys, randomly select 1000 values range from 0 to 1 as k_1 for one set of the secret key, but keep keys {k_2, k_3, k_4} always are {0.7329, 0.5712, 0.8320}, we get 1000 secret keys as one set of them. Keep the value of keys {k_2, k_3, k_4}, but change each k_1 value with 10^{-14}, we get the other set of secret key, then calculate the average, minimum and maximum values of NPCR, UACI and NMSE between the obtained two set of cipher images; (2) Same as method 1, but keep keys {k_1, k_3, k_4} as {0.2789, 0.7904, 0.3321}, while make the 1000 k_2 different as the k_1 in method 1; (3) Same as method 1, but keep keys {k_1, k_2, k_4} as {0.7739, 0.3015, 0.4848}, while make the 1000 k_3 different as the k_1 in method 1; (4) Same as method 1, but keep keys {k_1, k_2, k_3} as {0.4908, 0.6583, 0.4989}, while make the 1000 k_4 changed with 10^{-13}. These results are tabulated in Table 3 and 4.

For decryption: (1) Generate one set of cipher images from a plain image with a set of secret keys. For the secret keys, randomly select 1000 values range from 0 to 1 as k_1, but keep keys {k_2, k_3, k_4} always are{0.7329, 0.5712, 0.8320} to get the set of secret keys. Keep the value of keys {k_2, k_3, k_4}, but change each k_1 value with 10^{-14}, we get the other set of secret keys. Then decrypt the set of cipher images with the correct secret keys, and the secret keys in the other set of secret keys. This way, we get two set of decrypted images. Then calculate the average, minimum and maximum values of NPCR, UACI and NMSE between the obtained two set of decrypted images; (2) Same as method 1, but keep keys {k_1, k_3, k_4} as {0.2789, 0.7904, 0.3321}, while make the 1000 k_2 different as the k_1 in method 1; (3) Same as method 1, but keep keys {k_1, k_2, k_4} as {0.7739, 0.3015, 0.4848}, while make the 1000 k_3 different as the k_1 in method 1; (4) Same as method 1, but keep keys {k_1, k_2, k_3} as {0.4908, 0.6583, 0.4989}, while make the 1000 k_4 changed with 10^{-13}. These results are tabulated in Table 5 and 6, where the values in bracket are theoretical values.

Table 3. Key sensitivity analysis of Lena image (for encryption)

	NPCR(99.6094%)			UACI(33.4635%)			NMSE(50%)		
	max	mean	min	max	mean	min	max	mean	min
k1	99.6559	99.6092	99.5731	33.6058	33.4638	33.3210	50.7289	50.2989	49.9027
k2	99.6502	99.6090	99.5701	33.6324	33.4614	33.2935	50.7511	50.2985	49.6817
k3	99.6536	99.6092	99.5693	33.5859	33.4617	33.3383	50.7271	50.2911	49.8652
k4	99.6521	99.6093	99.5682	33.6388	33.4636	33.3043	50.7431	50.2938	49.8573

Table 4. Key sensitivity analysis of Baboon image (for encryption)

	NPCR(99.6094%)			UACI(33.4635%)			NMSE(50%)		
	max	mean	min	max	mean	min	max	mean	min
k1	99.6441	99.6087	99.5644	33.6439	33.4643	33.3170	50.7530	50.2962	49.8735
k2	99.6498	99.6088	99.5476	33.5983	33.4606	33.3212	50.7171	50.2875	49.8189
k3	99.6456	99.6095	99.5701	33.6208	33.4632	33.3295	50.7571	50.2884	49.8407
k4	99.6517	99.6099	99.5686	33.5946	33.4635	33.3216	50.7684	50.2884	49.8407

Table 5. Key sensitivity analysis of Lena image (for decryption)

	NPCR(99.6094%)			UACI(28.6242 %)			NMSE(35.6210%)		
	max	mean	min	max	mean	min	max	mean	min
k1	99.6460	99.6095	99.5674	28.7380	28.6250	28.5041	36.0229	35.7467	35.4594
k2	99.6456	99.6090	99.5701	28.7455	28.6249	28.4778	35.9800	35.7477	35.4769
k3	99.6532	99.6103	99.5762	28.7515	28.6256	28.5164	36.0274	35.7497	35.4211
k4	99.6531	99.6088	99.5674	28.7514	28.6236	28.5075	36.0747	35.7484	35.4883

Table 6. Key sensitivity analysis of Baboon image (for decryption)

	NPCR(99.6094%)			UACI(27.8472%)			NMSE(33.2810%)		
	max	mean	min	max	mean	min	max	mean	min
k1	99.6517	99.6095	99.5708	28.0011	27.8744	27.7123	33.6832	33.4115	33.1236
k2	99.6426	99.6094	99.5693	27.9665	27.8482	27.7419	33.6330	33.4153	33.1394
k3	99.6609	99.6093	99.5716	27.9511	27.8462	27.7343	33.7258	33.4103	33.1756
k4	99.6437	99.6094	99.5724	27.9549	27.8476	27.7488	33.6582	33.4110	33.1457

From Tables 3-6, it can be deduced that the maximum relative error of NPCR is 0.06204%, the maximum relative error of UACI is 0.5527%, and the maximum relative error of NMSE is 1.5368%, which demonstrate that the proposed scheme has high key sensitivity.

4.7. Plain Image Sensitivity and Resisting Chosen Plaintext Attack

Take the images of Lena, Baboon, all-white and all-black of 8-bit grayscale (of size 512 × 512) as examples. Experiment on them for 10,000 times. In each experiment, randomly select one pixel from one plain image, and plus 1 to the same pixel to get the other plain image, encrypt the two plain images with the identical secret key to get two cipher images, then compare the calculated NPCR and UACI indicators. The results are tabulated in Table 7.

Table 7. NPCR and UACI indicators for chosen plaintext attack

	NPCR(99.6094%)			UACI(33.4635%)		
	max	mean	min	max	mean	min
Lena	99.6563	99.6096	99.5625	33.6176	33.4595	33.2848
Baboon	99.6521	99.6095	99.5590	33.6166	33.4709	33.3199
All-black	99.6590	99.6093	99.5647	33.6551	33.4629	33.2909
All-white	99.6490	99.6090	99.5590	33.6176	33.4483	33.2873

From Table 7, it can be calculated that the maximum relative error of NPCR is 0.05060%, and the maximum relative error of UACI is 0.57256%, which demonstrates that any slightly change in the plain images will make the produced cipher images completely different, and each pixel's information in plain image can spread all over the cipher image. Since the proposed method being the plain image related algorithm, i.e. different plain images correspond to different secret code streams, that makes the proposed scheme can resist the chosen plain image attack.

5. Conclusion

This paper suggested the plain image related image encryption method based on image blocks. On the one hand a part of the information of plain image is used to confuse the pixels within each block, on the other hand the plain image information is employed to make diffusion between the adjacent block, so as to achieve the purposed of encrypting original image. The entire process does not disrupt the pixel locations, and it just scrambles and diffuses the value of each pixel. Simulation results confirm that the proposed scheme has the characteristics of high speed, large key space, high key sensitivity, and well resisting differential attack and chosen plain image attack, etc. So the proposed scheme can be used in practical communications.

Acknowledgement

This work was fully supported by the Natural Science Foundations of Jiangxi Province (Grant Nos: 20122BAB201036 and 20114BAB211011).

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] Huang CK, Nien HH. Multi chaotic systems based pixel shuffle for image encryption. *Opt Commun*. 2009; 282(11): 2123-2127.
[5] Gao TG, Chen ZQ. Image encryption based on a new total shuffling algorithm. *Chaos Solitons Fractals*. 2008; 38(1): 213-220.
[6] Yu WW, Cao JD. Cryptography based on delayed chaotic neural networks. *Phys Lett A*. 2006; 356(4): 333-338.
[7] Solak E, Rhouma R, Belghith S. Cryptanalysis of a multi-chaotic systems based image cryptosystem. *Optics Communications*. 2010; 282(2): 232-236.
[8] Arroyo D, Li CQ, Li SJ, Alvarez G, Halang WA. Cryptanalysis of an image encryption scheme based on a new total shuffling algorithm. *Chaos Solitons Fractals*. 2009; 41(5): 2613-2616.
[9] Yang JY, Liao XF, Yu WW, Wong KW, Wei J. Cryptanalysis of a cryptographic scheme based on delayed chaotic neural networks. *Chaos Solitons Fractals*. 2009; 40(2): 821-825.

[10] Mazloom S, Eftekhari-Moghadam AM. Color image encryption based on coupled nonlinear chaotic map. *Chaos Solitons Fractals*. 2009; 42(3): 1745-1754.
[11] Yang HQ, Wong KW, Liao XF, Zhang W, Wei PC. A fast image encryption and authentication scheme based on chaotic maps. *Commun Nonlinear Sci Numer Simulat*. 2010; 15(11): 3507-3517.
[12] Zhang G, Liu Q. A novel image encryption method based on total shuffling scheme. *Opt Commun.* 2011; 284(12): 2775-2780.
[13] Wang X, He G. Cryptanalysis on a novel image encryption method based on total shuffling scheme. *Opt Commun*. 2011; 284(24): 5804-5807.
[14] Eslami Z, Bakhshandeh A. An improvement over an image encryption method based on total shuffling. *Opt Commun*. 2013; 286(1): 51-55.
[15] Ye G, Wong K-W. An efficient chaotic image encryption algorithm based on a generalized Arnold map. *Nonlinear Dynamics*. 2012; 69(4): 2079-2087.
[16] Ye R. A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism. *Opt Commun*. 2011; 284(22): 5290-5298.
[17] Adb El-Latif AA, Li L, Zhang T, Wang N, Song X, Niu X. Digital image encryption scheme based on multiple chaotic systems. *Sensing and Imaging: An International Journal*. 2012; 13(2): 67-88.
[18] 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.
[19] Horvat M, Esposti MD, Isola S, Prosen T, Bunimovich L. On ergodic and mixing properties of the triangle map. *Physica D*. 2009; 238(4): 395-415.