A fast image encryption algorithm based on only blocks in cipher text*

Wang Xing-Yuan(王兴元)† and Wang Qian(王 倩)

Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, *China*

(Received 12 July 2013; revised manuscript received 5 September 2013; published online 24 January 2014)

In this paper, a fast image encryption algorithm is proposed, in which the shuffling and diffusion is performed simultaneously. The cipher-text image is divided into blocks and each block has $k \times k$ pixels, while the pixels of the plain-text are scanned one by one. Four logistic maps are used to generate the encryption key stream and the new place in the cipher image of plain image pixels, including the row and column of the block which the pixel belongs to and the place where the pixel would be placed in the block. After encrypting each pixel, the initial conditions of logistic maps would be changed according to the encrypted pixel's value; after encrypting each row of plain image, the initial condition would also be changed by the skew tent map. At last, it is illustrated that this algorithm has a faster speed, big key space, and better properties in withstanding differential attacks, statistical analysis, known plaintext, and chosen plaintext attacks.

Keywords: image encryption, shuffle and diffuse simultaneously, block cipher, logistic map

PACS: 05.45.Gg, 05.45.–a, 05.45.Vx **DOI:** 10.1088/1674-1056/23/3/030503

1. Introduction

With the development of internet and multimedia technology, encryption of digital images has become an emergency research in recent years. Because of its properties of bulky data and strong correlations among adjacent pixels, the traditional cryptographies such as RSA,[1,2] DES[3,4] are unsuited to image encryption.

Chaotic systems are characterized by sensitive dependence on initial conditions, pseudo-randomness, ergodicity, and reproduction. Their pseudorandom sequences have good features of randomness, non-relevance, and complexity and are quite suitable in image encryption. Chaos has been widely used in physics,[5] biology,[6] medical technology,[7] synchronous control,[8] complex networks,[9] and so on. According to the classification of chaotic systems, the chaotic encryption schemes which have been proposed, could be divided into analog chaotic cryptosystems utilizing continuous dynamical systems[10] and digital chaotic cryptosystems utilizing discrete dynamical systems.[11] Lots of image encryption algorithms which are digital chaotic cryptosystems have been proposed in recent years, including stream cipher[12] and block cipher.[13–18] It tends to be a promising direction that chaotic maps are used in image encryption.

The stream cipher encryption algorithm proposed by Sun *et al.* in Ref. [12] has been proven to be unable to resist a chosen plaintext attack in Ref. [19].

Among those algorithms used in block cipher, some algorithms have three phases: Seyedzadeh *et al.*[14] proposed an image algorithm which is different from the traditional image encryption that has two stages, shuffling and diffusion. It has three stages, forward encryption, reverse encryption, and masking; Kanso *et al.*[17] raised an algorithm which adds a masking stage on the basis of traditional image encryption stages. These two algorithms can satisfy the security of resisting differential attacks, but both of them cannot resist known and chosen plaintext attacks and the image pixels must be scanned three times in each encryption round.

Some, as traditional algorithms, have two phases: Gao *et al.*[18] used logistic and hyper-chaotic maps for shuffling and diffusion respectively. This algorithm cannot resist differential attacks nor known and chosen plaintext attacks and the image pixels must be scanned two times in each encryption round.

Some only have one phase: Wang *et al.*[16] proposed an image encryption algorithm which combines the shuffling and diffusion stages into one stage. This algorithm is based on the nearest-neighboring coupled-map lattices, tent map and S-box. The plain image is divided into blocks which would be shuffled as a unit; an algorithm also doing shuffling and diffusion simultaneously was proposed in Ref. [15], but the pixel of the image is as the unit in shuffling. Behnis *et al.*[13] proposed an image encryption algorithm in which the shuffling stage was abandoned. Although the algorithm in Ref. [16] is one phase, the preprocessing stage must be performed before the encryption and the image is scanned once in the preprocessing. Actually, the image is scanned twice in one encryption round. The encryption must be performed at least twice

*Project supported by the National Natural Science Foundation of China (Grant Nos. 61370145, 61173183, and 60973152), the Doctoral Program Foundation of Institution of Higher Education of China (Grant No. 20070141014), the Program for Liaoning Excellent Talents in University, China (Grant No. LR2012003), the National Natural Science Foundation of Liaoning Province, China (Grant No. 20082165), and the Fundamental Research Funds for the Central Universities, China (Grant No. DUT12JB06).

†Corresponding author. E-mail: wangxy@dlut.edu.cn; wq605053@163.com

© 2014 Chinese Physical Society and IOP Publishing Ltd

in order to resist differential attacks. Moreover, the algorithm employs S-box and it is not described how to use it in detail. Therefore, there would be two cases: if they use the S-box proposed in the example, the use of S-box becomes unnecessary; if they use the S-box constructed just at the encryption time, the time consumed by the encryption algorithm is related with the construction algorithm of the S-box. In order to obtain good security, the algorithms in Ref. [16] and in Ref. [13] must be processed in more than three rounds.

Generally speaking, most of the proposed image encryption algorithms are not perfect. Based on the analysis of the above image encryption algorithms, we propose an algorithm in which the shuffling and diffusion stage is combined and we only have to scan the image once in one encryption round. In this algorithm, the plain image pixels are scanned one by one and the pixels are packed into different groups in the cipher image. In order to resist known and chosen plaintext attacks, the chaotic maps' states are altered according to the encrypted pixel, and in order to get a good diffusion effect and security, the algorithm must be performed more than two times, and it is illustrated that this algorithm has better properties of speed and better security than other algorithms.

The rest of the paper consists of as follows: In Section 2, the encryption and decryption algorithm will be stated; in Section 3, the security and speed analysis of the algorithm is illustrated; the conclusion is given in Section 4.

2. Encryption and decryption algorithm

The proposed algorithm is a new image encryption algorithm based on iterating chaotic maps. The pseudorandom sequences generated by a group of chaotic maps are used to encrypt the image and a special chaotic map is used to alter the initial conditions of the group of chaotic maps. This group consists of four chaotic maps and chaotic maps with good properties must be chosen. For simplicity, a logistic map is chosen,[15,20] and the group of logistic maps can be presented as

$$\begin{cases} r_{n+1} = \mu_r r_n(1-r_n), \\ c_{n+1} = \mu_c c_n(1-c_n), \\ p_{n+1} = \mu_p p_n(1-p_n), \\ k_{n+1} = \mu_k k_n(1-k_n). \end{cases} \tag{1}$$

where r_n, c_n, p_n, $k_n \in (0,1)$, u_r, u_c, u_p, $u_k \in (0,4]$, and when u_r, u_c, u_p, $u_k \in (3.8,4]$, the pseudorandom sequences between 0 and 1 are almost equally distributed.

The skew tent map is selected as the special chaotic map and it can be presented by

$$x_{n+1} = g(x_n,b) = \begin{cases} \dfrac{x_n}{b}, & 0 \le x \le b, \\ \dfrac{1-x_n}{1-b}, & b < x \le 1, \end{cases} \tag{2}$$

where $x_n \in (0,1)$ and $b \in (0,1)$.

A 256-bit external key is needed and presented as

$$K = \{k_l\} \quad (l = 1,2,\ldots,32).$$

The plain-text image matrix is denoted as

$$P = \{p_{i,j}\} \quad (i = 1,2,\ldots,M;\ \ j = 1,2,\ldots,N),$$

where M, N are the height and width of the image matrix, respectively, and the cipher-text image is denoted as

$$C = \{c_{i,j}\} \quad (i = 1,2,\ldots,M;\ \ j = 1,2,\ldots,N),$$

and is initialized to a zero matrix, t is a common divisor of M and N. The cipher matrix is divided into blocks as

$$B = \{b_{r,c}\} \quad (r = 1,2,\ldots,M/t;\ \ c = 1,2,\ldots,N/t),$$

and each block has $t \times t$ pixels.

The encryption algorithm is stated as follows, and its flow chart is shown in Fig. 1.

Step 1 Calculate the initial conditions and parameters of chaotic maps by

$$\text{sum} = \sum_{l=1}^{32} k_l, \tag{3}$$

$$b_1 = \{(k_4 \oplus k_5 + k_6 + \text{sum})/2^8\}, \tag{4}$$

$$b_2 = \{(k_7 \oplus k_8 + k_9 + \text{sum})/2^8\}, \tag{5}$$

$$b_3 = \{(k_2 \oplus k_3 + k_4 + \text{sum})/2^8\}, \tag{6}$$

$$b_4 = \{(k_5 \oplus k_6 + k_7 + \text{sum})/2^8\}, \tag{7}$$

$$b_5 = \{(k_8 \oplus k_9 + k_{10} + \text{sum})/2^8\}, \tag{8}$$

$$b_6 = \{(k_{11} \oplus k_{12} + k_{13} + \text{sum})/2^8\}, \tag{9}$$

$$\mu_r = \{(k_{10} \oplus k_{11} + k_{12} + \text{sum})/2^8\}/5 + 3.8, \tag{10}$$

$$\mu_c = \{(k_{13} \oplus k_{14} + k_{15} + \text{sum})/2^8\}/5 + 3.8, \tag{11}$$

$$\mu_p = \{(k_{16} \oplus k_{17} + k_{18} + \text{sum})/2^8\}/5 + 3.8, \tag{12}$$

$$\mu_k = 4.0, \tag{13}$$

$$r_1 = \{(k_{19} \oplus k_{20} + k_{21} + \text{sum})/2^8\}, \tag{14}$$

$$r_2 = \{(k_{22} \oplus k_{23} + k_{24} + \text{sum})/2^8\}, \tag{15}$$

$$r_3 = \{(k_{14} \oplus k_{15} + k_{16} + \text{sum})/2^8\}, \tag{16}$$

$$c_1 = ((k_{25} \oplus k_{26}) + (k_{27} \oplus k_{28}) + \text{sum}) \bmod 256, \tag{17}$$

$$c_2 = ((k_{29} \oplus k_{30}) + (k_{31} \oplus k_{32}) + \text{sum}) \bmod 256, \tag{18}$$

$$c_3 = ((k_{21} \oplus k_{22}) + (k_{23} \oplus k_{24}) + \text{sum}) \bmod 256, \tag{19}$$

where $\{x\}$ gets the decimal part of x. With Eqs. (3)–(19), the initial conditions and parameters for the first three rounds have been obtained. If the image needs to be encrypted in more than 3 rounds, the initial conditions and parameters can be gotten in a similar way. At last, set $R = [r_1, r_2, r_3]$, $B_1 = [b_1, b_3, b_5]$, $B_2 = [b_2, b_4, b_6]$, $C_0 = [c_1, c_2, c_3]$, and $n_r = 1$.

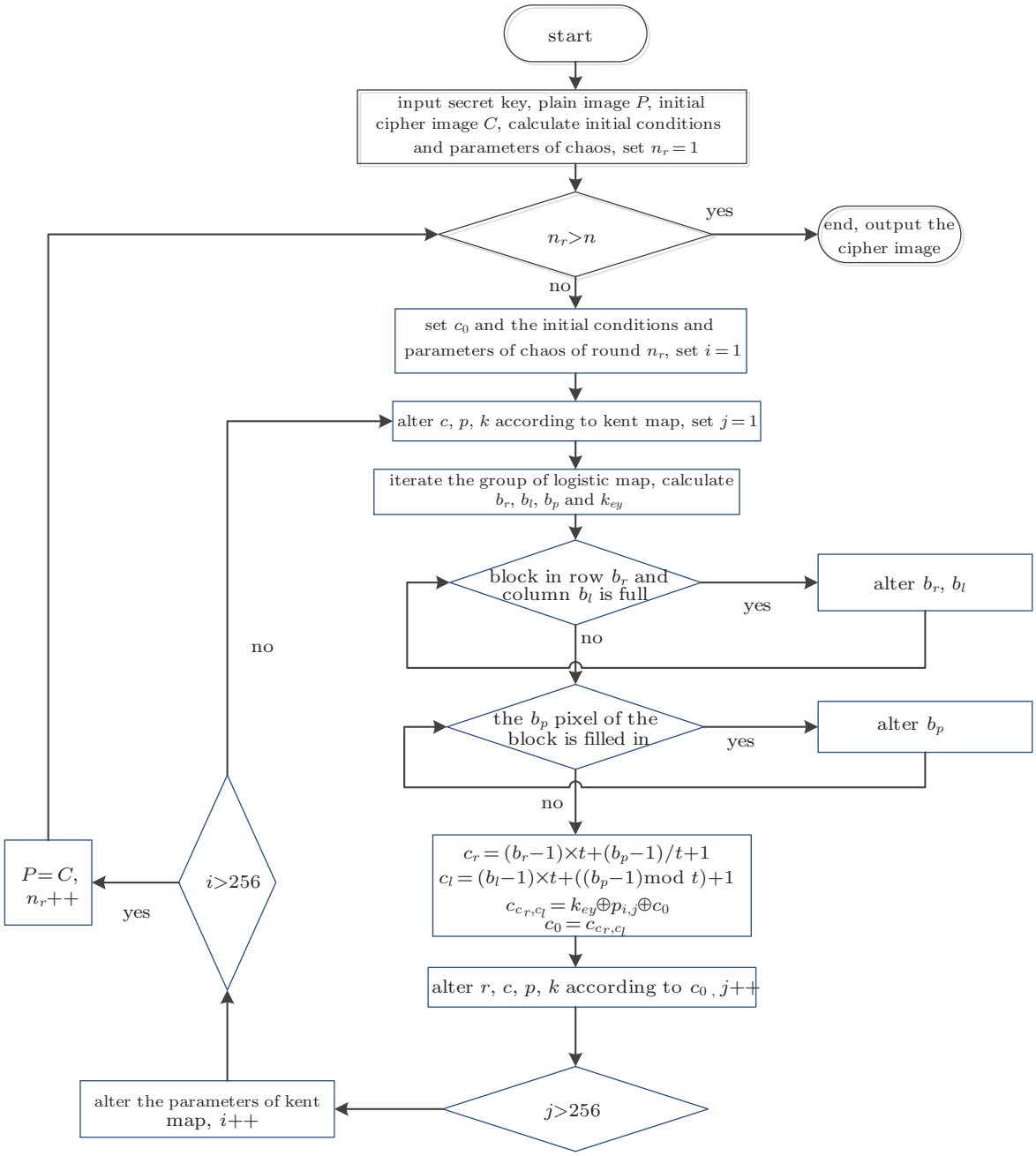


Fig. 1. Flow chart of encryption algorithm.

Step 2 Set c_0 and initial conditions and parameters of the chaotic maps by

$$r = R(n_r), \tag{20}$$

$$c_0 = C_0(n_r), \tag{21}$$

$$b_1 = B_1(n_r), \tag{22}$$

$$b_2 = B_2(n_r). \tag{23}$$

Then two matrixes are initiated by

$$\text{status} = \text{ones}(M/t, N/t) \times t \times t,$$

$$\text{statusn} = \text{ones}(M/t, N/t, t \times t),$$

which are used to judge if the block is full and if the pixel is filled, where ones() is a function used to initiate a matrix where the elements are all equal to the one in MATLAB. At last, set $i = 1$.

Step 3 Alter the initial conditions of the group of logistic map by doing

$$c = g(r, b_1), \tag{24}$$

$$p = g(c, b_2), \tag{25}$$

$$k = \{r + c + p\}, \tag{26}$$

and set $j = 1$.

Step 4 Iterate the logistic maps presented as

$$\begin{cases} r = f(r,\mu_r), \\ c = f(c,\mu_c), \\ p = f(p,\mu_p), \\ k = f(k,\mu_k), \end{cases} \tag{27}$$

and calculate b_r, b_l, b_p, k_{ey} by

$$b_r = \{[r \times 10^{14}] \bmod M/t\} + 1, \tag{28}$$

$$b_c = \{[c \times 10^{14}] \bmod N/t\} + 1, \tag{29}$$

$$b_p = \{[p \times 10^{14}] \bmod (t \times t)\} + 1, \tag{30}$$

$$k_{ey} = [k \times 10^{14}] \bmod 256, \tag{31}$$

where b_r, b_l are the indexes of the row and column of a block, respectively; the pixel $p_{i,j}$ would be filled in b_p of the block; k_{ey} is the encryption key of the pixel $p_{i,j}$.

Step 5 Judge if the block b_{b_r,b_l} is full and whether the place b_p of the block is filled and according to the judgment deal with b_r, b_l, b_p by the following method.

When status$(b_r,b_c) < 1$, the block b_{b_r,b_l} is full and the prior block of b_{b_r,b_l} is selected as the destination block. Repeat this judgment and selection until status$(b_r,b_c) \geq 1$. Next, when statusn$(b_r,b_c,b_p) < 1$, the prior place is selected as the destination place. Repeat this judgment and selection until statusn$(b_r,b_c,b_p) \geq 1$.

Then do

$$\text{status}(b_r,b_c) = \text{status}(b_r,b_c) - 1, \tag{32}$$

$$\text{statusn}(b_r,b_c,b_p) = \text{statusn}(b_r,b_c,b_p) - 1. \tag{33}$$

The place of $p_{i,j}$ in the cipher image is obtained and its place in the index of the row and column in block b_{b_r,b_l} can be calculated by

$$c_r = (b_r - 1) \times t + (b_p - 1)/t + 1, \tag{34}$$

$$c_l = (b_l - 1) \times t + ((b_p - 1) \bmod t) + 1. \tag{35}$$

Step 6 Encrypt the pixel $p_{i,j}$ using

$$c_{c_r,c_l} = k_{ey} \oplus p_{i,j} \oplus c_0. \tag{36}$$

Step 7 Alter c_0 and r, c, p, k accordingly by

$$c_0 = c_{c_r,c_l}, \tag{37}$$

$$r = \left\{ r + \frac{c_0}{1000} \right\}, \tag{38}$$

$$c = \left\{ c + \frac{c_0}{100} \right\}, \tag{39}$$

$$p = \left\{ p + \frac{c_0}{10} \right\}. \tag{40}$$

Step 8 Do $j = j + 1$, and repeat steps 4–8 until $j > 256$.

Step 9 Alter the parameters of the skew tent maps by

$$b_1 = \left\{ b_1 + \frac{c_0}{100} \right\}, \tag{41}$$

$$b_2 = \left\{ b_2 + \frac{c_0}{10} \right\}. \tag{42}$$

Step 10 Do $i = i + 1$, and repeat steps 3–10 until $i > 256$.

Step 11 Do $P = C$ and $n_r = n_r + 1$, and repeat steps 2–11 until $n_r > n$, where n is the number of rounds of encryption, and in our experiment $n = 3$.

The decryption is similar to the encryption algorithm except

(i) At the end of step 1, set $n_r = n$ instead of $n_r = 1$.

(ii) The two matrixes status and statusn are used to judge if the pixel of the block is decoded.

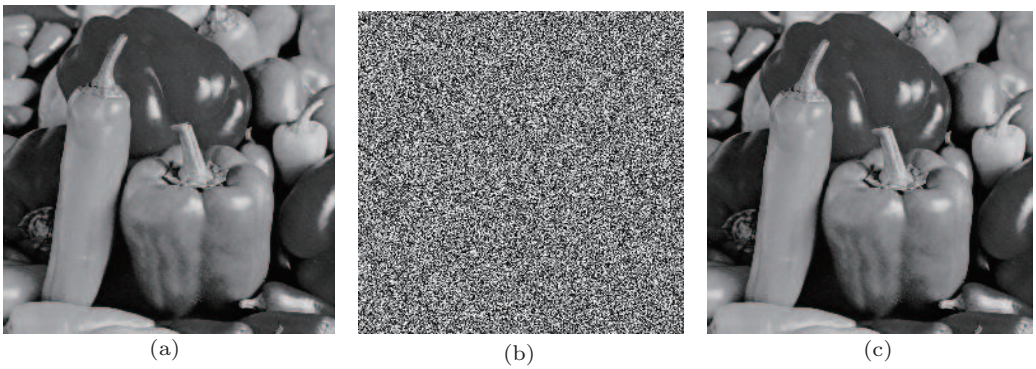
(iii) In step 4, are the place of the pixel which is going to be decoded and k_{ey} is the decoding key.

(iv) The order of steps 6 and 7 are changed. However, in step 6, decoding the pixel c_{c_r,c_l} and equation (36) is changed by

$$p_{i,j} = k_{ey} \oplus c_{c_r,c_l} \oplus c_0. \tag{43}$$

(v) Step 11, do $C = P$ and $n_r = n_r - 1$, and repeat steps 2–11 until $n_r < 1$.

Our experimental result is shown in Fig. 2. Figure 2(a) shows the plain-text image of peppers and its cipher-text image and decoding image are shown in Figs. 2(b) and 2(c), respectively.



(a) (b) (c)

Fig. 2. Experimental result of the algorithm proposed. (a) Peppers. (b) Cipher image. (c) Decoding image.

3. Security analysis and speed analysis

3.1. Statistical analysis

It is well known that the statistical property of a cipher image is very important and an ideal image algorithm should be robust against any statistic attacks. A histogram and the correlation of two adjacent pixels are two important indicators of statistical analysis.

Histogram A histogram is plotted according to the pixel values of images and it counts the number of pixels which have the same pixel value. A good image algorithm should make the histogram of cipher image as flat as possible. The histograms of peppers and its cipher images are shown in Fig. 3. Figure 3(a) is a histogram of peppers and figures 3(b), 3(c), and 3(d) are the histograms of the cipher images encrypted with one round, two rounds, and three rounds respectively.

Correlation of two adjacent pixels Generally speaking, the two adjacent pixels of a plain image would come near to each other and a good image encryption algorithm could smash the relation between them. 1000 pairs of adjacent pixels from the plain image of peppers and its cipher image encrypted with one round are selected randomly in the horizontal direction, vertical direction, and diagonal direction, respectively and the correlation of them is plotted out. The results are shown in Fig. 4. Moreover, the correlation coefficients r_{xy} of each pair are calculated using the following equations:[16]

$$\mathrm{cov}(x,y) = E\{(x-E(x))(y-E(y))\}, \tag{44}$$

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{45}$$

where x and y are the values of the two adjacent pixels in the image

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

and

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

The correlation coefficients of the plain image and cipher images are shown in Table 1 and Fig. 4, from which, it can be displayed that there is no detectable correlations between the plain image and its corresponding cipher images.

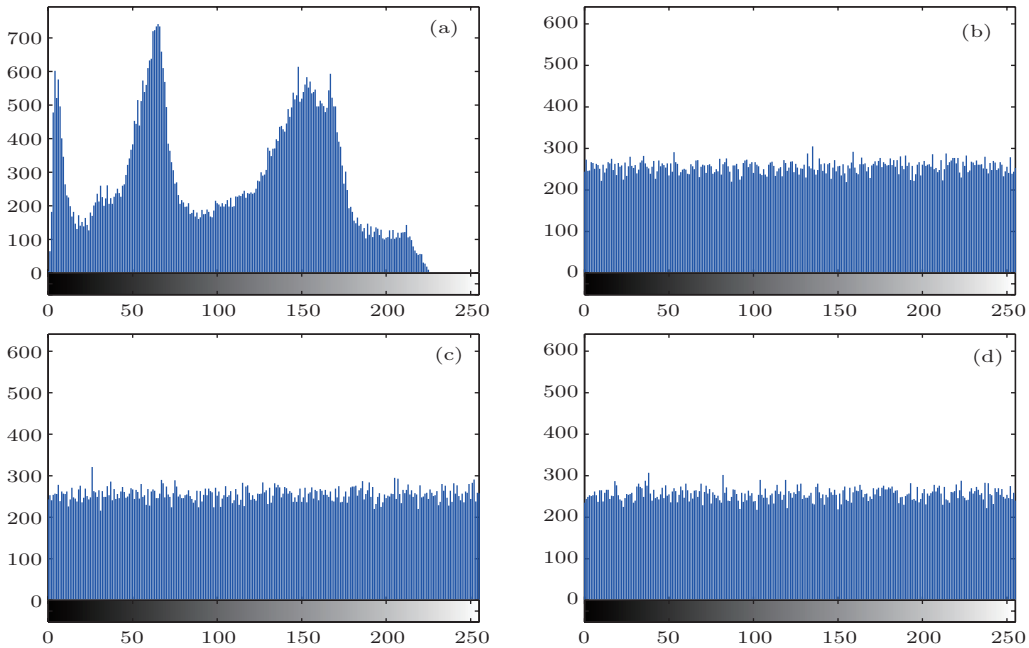


Fig. 3. (color online) Histograms of plain image and cipher images. (a) Histogram of peppers. (b) Histogram of encryption with one round. (c) Histogram of encryption with two rounds. (d) Histogram of encryption with three rounds.

Table 1. Correlation coefficients of two adjacent pixels in the plain and cipher images.

Name	Plain image			Cipher image (one round)			Cipher image (two rounds)			Cipher image (three rounds)		
	Horizontal	Diagonal	Vertical	Horizontal	Diagonal	Vertical	Horizontal	Diagonal	Vertical	Horizontal	Diagonal	Vertical
Lena	0.9502	0.9303	0.9768	0.0040	0.0397	0.0238	0.0081	0.0262	0.0085	0.0014	0.0171	0.0054
Moon	0.8870	0.8942	0.9344	0.0509	0.0112	0.0296	0.0035	0.0638	0.0338	0.0061	0.0419	0.0268
Boat	0.9299	0.8832	0.9388	0.0036	0.0149	0.0283	0.0155	0.0147	0.0341	0.0079	0.0200	0.0063
Brain	0.9525	0.9298	0.9693	0.0687	0.0096	0.0157	0.0026	0.0299	0.0536	0.0612	0.0303	0.0266
Finger	0.9096	0.8338	0.9510	0.0643	0.0172	0.0573	0.0084	0.0346	0.0364	0.0343	0.0122	0.0273

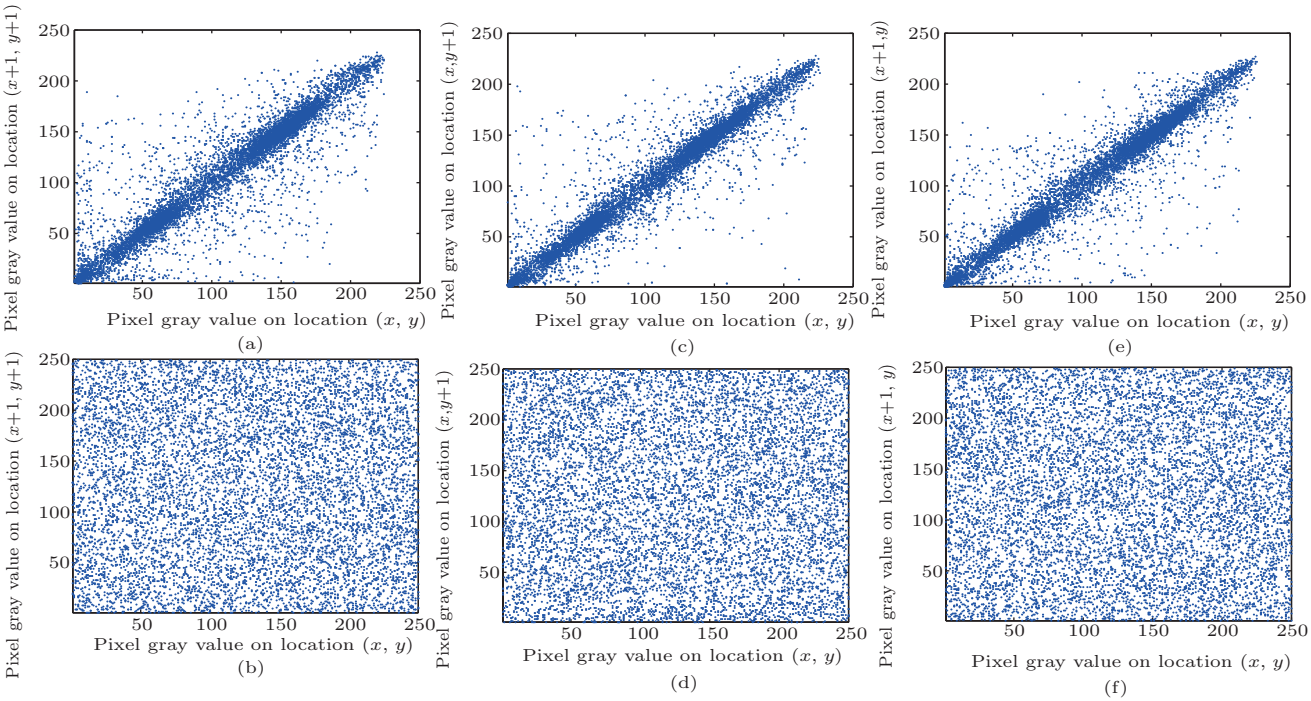


Fig. 4. (color online) Correlations of two adjacent pixels. (a) Diagonal direction in plain image, (b) diagonal direction in cipher image, (c) horizontal direction in plain image, (d) horizontal direction in cipher image, (e) vertical direction in plain image, and (f) vertical direction in cipher image.

3.2. Key space analysis

It is general knowledge that a good encryption algorithm for images should have a big key space and be sensitive to its key. In this paper, an external 256-bit secret key is designed and all initial conditions and parameters are generated by the secret key.

Key space In the proposed algorithm, a 256-bit external key is needed, so that the key space is 2^{256} and it is big enough to stand up to a brute force attack.

Sensitivity to secret key 1) An external key "116, 3, 163, 12, 213, 82, 4, 130, 56, 112, 27, 101, 127, 90, 110, 19, 216, 19, 1, 157, 149, 24, 223, 68, 1, 30, 41, 65, 23, 64, 19, 16" in which each 8-bit is presented by decimalism is used to encrypt the plain image "boat.bmp", and a slightly changed key "116, 3, 163, 12, 213, 82, 4, 130, 56, 112, 27, 101, 127, 90, 110, 19, 216, 19, 1, 157, 149, 24, 223, 68, 1, 30, 41, 65, 23, 64, 19, 17" and the correct key are used to decrypt the cipher image respectively. The results are shown in Fig. 5. 2) The different rate (DR) of the cipher images which are encrypted by the correct key and the slightly changed key respectively is calculated by

$$\mathrm{DR} = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%,$$

where W, H are the width and height of image; D is a matrix. The plain image is encrypted by two different external keys and the cipher images are presented as C_1 and C_2. When $C_1(i,j) = C_2(i,j)$, otherwise $D(i,j) = 0$. We compare our algorithm with some other algorithms in Refs. [13], [15], [16], and [18]. The results are shown in Table 2. From the table, it is known that our algorithm has good properties in sensitivity to the secret key.

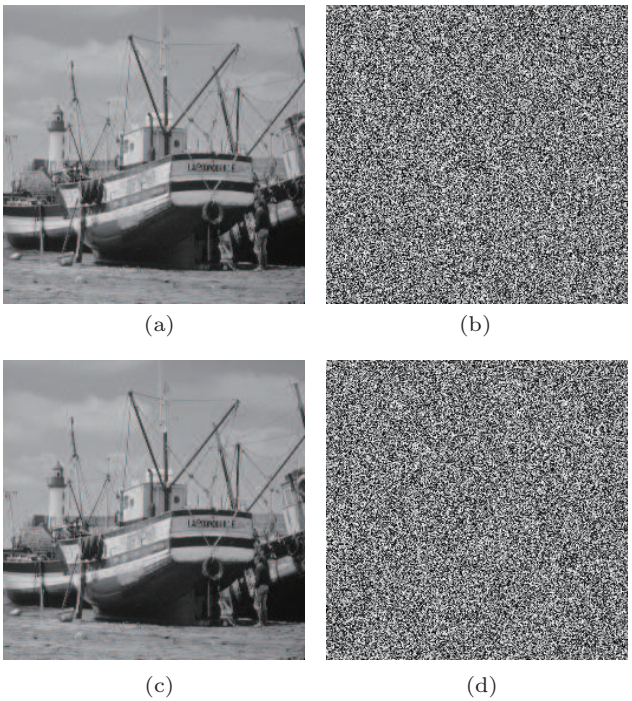


Fig. 5. (color online) The plain image, encoding image, decoding correctly image, and decoding by a slightly changed key. (a) Boat, (b) encoding image, (c) decoding correctly, (d) decoding with a slightly changed key.

Table 2. Different rates of the cipher images encrypted by correct key and the slightly changed key.

Name	Our algorithm	Ref. [13]	Ref. [15]	Ref. [16]	Ref. [18]
House	0.9963	0.9962	0.9962	0.9957	0.9755
Zone	0.9962	0.9961	0.9960	0.9962	0.9890
Plane	0.9960	0.9958	0.9962	0.9961	0.9866
Bird	0.9962	0.9960	0.9955	0.9960	0.9882
Finger	0.9960	0.9961	0.9965	0.9960	0.9033

3.3. Information entropy

The information entropy can be calculated by

$$H(m) = \sum_{i=0}^{M-1} p(m_i) \log \frac{1}{p(m_i)}, \tag{46}$$

in which m is a set of symbols; M is the total number of symbols $m_i \in m$; $p(m_i)$ is the probability of m_i. In the experiment 256 gray level images are used, so the theoretic $H(m)$ should be 8. The information entropy of cipher images is shown in Table 3. From Table 3, it can be known that the information entropies are close to eight, so that the algorithm proposed has good properties in information entropy.

Table 3. Information entropies of cipher images.

Name	Our algorithm	Ref. [13]	Ref. [15]	Ref. [16]	Ref. [18]
House	7.9975	7.9973	7.9973	7.9969	7.9969
Zone	7.9972	7.9968	7.9972	7.9971	7.9968
Lena	7.9974	7.9970	7.9975	7.9975	7.9972
Bird	7.9974	7.9968	7.9967	7.9977	7.9970
Finger	7.9970	7.9974	7.9968	7.9962	7.9952

3.4. Resisting differential attack analysis

In order to resist differential attacks, a petty change in the plain image must cause an amount of difference of pixels in cipher images. Two quantitative measures, which are the number of pixels change rate (NPCR) and the unified average changing intensity (UACI), are used to measure the influence of a one-pixel change on the cipher images. A cipher image and its plain image are presented as C_1, I_1 while another cipher image and its plain image are presented as C_2, I_2 where there is only one pixel value which is different between I_1 and I_2. A matrix D is created and when $C_1(i,j) = C_2(i,j)$, then $D(i,j) = 0$; otherwise $D(i,j) = 1$. NPCR and UACI are calculated by[17]

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%, \tag{47}$$

$$\text{UACI} = \frac{1}{W \times H} \left(\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255} \right) \times 100\%, \tag{48}$$

where W and H are the width and height of the image, respectively.

In the experiment, a pixel from the plain image is randomly taken and the pixel value is added 1, then we get a new plain image. Then the new cipher image is compared with the old cipher image to calculate NPCR and UACI. For each sample image, the last pixel and 999 randomly pixels are selected to change to get 1000 new plain images. After comparing each pair of new cipher image and old cipher image, the average value, the minimum value, and the maximum value of NPCRs and UACIs are calculated respectively. Besides, the experimental results are compared with the results of algorithms proposed in Refs. [13], [15], [16], and [18] and the results are shown in Table 4. A conclusion can be inferred that the proposed algorithm has good properties in resisting differential attacks.

Table 4. Comparison with other algorithms in resisting differential attacks.

Algorithms on peppers		One round			Two rounds			Three rounds		
		Minimum	Maximum	Average	Minimum	Maximum	Average	Minimum	Maximum	Average
Our results	NPCR	1.5259×10^{-5}	0.9786	0.4629	0.9954	0.9965	0.9960	0.9956	0.9968	0.9961
	UACI	1.4960×10^{-6}	0.3275	0.1554	0.3320	0.3366	0.3344	0.3335	0.3381	0.3356
Ref. [13]	NPCR	1.5259×10^{-5}	0.9955	0.5090	0.9955	0.9968	0.9961	0.9955	0.9966	0.9961
	UACI	7.4798×10^{-6}	0.3356	0.1710	0.3314	0.3361	0.3338	0.3324	0.3364	0.3346
Ref. [15]	NPCR	1.5259×10^{-5}	0.9784	0.4620	0.1187	0.9966	0.9868	0.9944	0.9967	0.9961
	UACI	4.8469×10^{-6}	0.3307	0.1554	0.0398	0.3367	0.3314	0.3323	0.3364	0.3347
Ref. [16]	NPCR	1.5259×10^{-5}	0.9761	0.4812	0.9956	0.9964	0.9961	0.9954	0.9969	0.9960
	UACI	5.9838×10^{-8}	0.3247	0.1471	0.3333	0.3342	0.3339	0.3329	0.3348	0.3340
Ref. [18]	NPCR	1.5259×10^{-5}	1.5259×10^{-5}	1.5259×10^{-5}	1.5259×10^{-5}	1.5259×10^{-5}	1.5259×10^{-5}	1.5259×10^{-5}	1.5259×10^{-5}	1.5259×10^{-5}
	UACI	1.1968×10^{-7}	1.9747×10^{-7}	1.5012×10^{-7}	7.5898×10^{-8}	1.1968×10^{-6}	2.3246×10^{-7}	5.9838×10^{-8}	1.2566×10^{-6}	4.1887×10^{-7}

3.5. Resisting known-plaintext and chosen-plaintext attacks analysis

According to our algorithm, the changes of the iteration condition of the logistic maps and parameters of the skew tent map are influenced by the encrypted pixel value and the encryption of a pixel is also influenced by its prior encrypted pixel value. With different images, the pixel of the same place in the plain image will be moved to different places in the cipher image and the key generated by the logistic map is different. Therefore, the proposed algorithm can well re-

sist the known- and chosen-plaintext attacks (as described in Refs. [21]– [23]).

3.6. Speed analysis

In the proposed algorithm, the shuffling and diffusion stages are combined for only one stage, so in each encryption round the image is only scanned once. This method can reduce the time of visiting memory and improve encryption speed. The encryption algorithm in Ref. [15], which has been found, performs the same idea in their encryption algorithm. Some comparisons with other algorithms on performance time are made. In our experiment, we measure the algorithms' performance time of their best security. For example, our algorithm has good security when it is performed more than two times, so we measure its performance time including two rounds' performance. Our experimental environment is MATLAB R2012a, a notebook computer with Intel Core i3-2330M CPU@ 2.20 GHz 2.20 GHz, 2.0G RAM and Window 7 OS. From experimental results shown in Table 5, it is known that our algorithm has good properties in speed.

Table 5. Comparisons of the proposed algorithm with other algorithms on speed (unit: s).

Name	Our algorithm (two rounds)	Ref. [13] (two rounds)	Ref. [15] (three rounds)	Ref. [16] (two rounds)	Ref. [18] (one round)
Boat	2.1760	2.7450	4.8900	30.3120	2.1560
Lena	2.1630	2.3700	4.9390	28.8460	2.1970
Bird	2.1840	2.2460	4.9890	30.4060	2.2010
Finger	2.1760	2.1240	4.9890	30.7110	2.2090
Moon	2.1790	2.5420	4.8980	29.2420	2.1020

4. Conclusion

In this paper, an image encryption algorithm based on multiple chaotic systems is proposed and the security of the algorithm is analyzed. The cipher image is divided into blocks while the plain image is scanned from left to right and from top to bottom. An idea that the encryption shuffling stage and diffusion stage should be combined into one stage is presented. That is, performing shuffling and diffusion simultaneously is employed. A group of logistic maps are used to generate the new place and the encryption key for each plain image pixel, and a skew tent map and encrypted cipher pixel values are used to alter the initial conditions and parameters of chaotic maps. In addition, the comparisons with other algorithms proposed in Refs. [13], [15], [16], and [18] are made for proving that the proposed algorithm has better properties in security and speed.

Although the proposed algorithm has good properties in security and speed, there is still lots of work to do. We may find an image encryption algorithm which would have high security and could consume little time. In addition, the precision of the decimal in different computers or different OS could affect the decryption of the image. It is important because the proposed algorithm is used in communication on the internet, so the next work would be solving the accordance between different environments.

References

[1] Chen C S, Wang T, Kou Y Z, Chen X C and Li X X 2013 *J. Syst. Softw.* **86** 100
[2] Guo W, Hu J W and Wei G 2013 *J. Comput.* **8** 33
[3] Coppersmith D 1994 *IBM J. Res. Dev.* **38** 243
[4] Han S J, Oh H S and Park J 1996 *Spread Spectrum Techniques and Applications Proceedings*, September 22–25, 1996, Mainz, Germany, p. 1310
[5] Weidenmueller H A and Mitchell G E 2009 *Rev. Mod. Phys.* **81** 539
[6] Stollenwerk N, Aguiar M, Ballesteros S, Boto J, Kooi B and Mateus L 2011 *Interface Focus* **2** 156
[7] Zhu Z L, Zhai K, Wang B L, Liu H J and Jiang H Y 2009 *International Congress on Image and Signal Processing*, October 17–19, 2009, Tianjin, China, p. 322
[8] Hegazi A S, Ahmed E and Matouk A E 2012 *Commun. Nonlinear Sci. Numer. Simul.* **18** 1193
[9] Lu L, Luan L, Meng L and Li C R 2012 *Nonlinear Dyn.* **70** 89
[10] Volos C K, Kyprianidis I M and Stouboulos I N 2013 *Signal Process.* **93** 1328
[11] Baptista M S 1998 *Phys. Lett. A* **240** 50
[12] Sun F Y and Lu Z W 2011 *Chin. Phys. B* **20** 040506
[13] Behnis S, Akhshani A, Ahadpour S, Mahnodi H and Akhavan A 2007 *Phys. Lett. A* **366** 391
[14] Seyedzadeh S M and Mirzakuchaki S 2012 *Signal Process.* **92** 1202
[15] Wang X Y, Zhao J F and Liu H J 2012 *Opt. Commun.* **285** 562
[16] Wang Y, Wong K W, Liao X F and Chen G R 2011 *Appl. Soft. Comput.* **11** 514
[17] Chen G, Mao Y and Chui C K 2004 *Chaos Soliton. Fract.* **21** 749
[18] Gao T G and Chen Z Q 2008 *Phys. Lett. A* **372** 394
[19] Wang X Y and Liu L T 2013 *Chin. Phys. B* **22** 050503
[20] Haeri M and Tavazoei M S 2007 *Appl. Math. Comput.* **187** 1076
[21] Wang Y, Liao X, Xiang T, Wong K W and D G Yang 2007 *Phys. Lett. A* **363** 277
[22] Wei J, Liao X F, Wong K W and Zhou T 2007 *Commun. Nonlinear Sci. Numer. Simul.* **12** 814
[23] Pei W J, Wang K, Zou L H, Song A G and He Z Y 2005 *Phys. Lett. A* **343** 432