A self-adapting image encryption algorithm based on spatiotemporal chaos and ergodic matrix*

Luo Yu-Ling(罗玉玲)† and Du Ming-Hui(杜明辉)

School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, *China*

(Received 20 November 2012; revised manuscript received 15 January 2013)

To ensure the security of a digital image, a new self-adapting encryption algorithm based on the spatiotemporal chaos and ergodic matrix is proposed in this paper. First, the plain-image is divided into different blocks of the same size, and each block is sorted in ascending order to obtain the corresponding standard ergodic matrix. Then each block is encrypted by the spatiotemporal chaotic system and shuffled according to the standard ergodic matrix. Finally, all modules are rearranged to acquire the final encrypted image. In particular, the plain-image information is used in the initial conditions of the spatiotemporal chaos and the ergodic matrices, so different plain-images will be encrypted to obtain different cipher-images. Theoretical analysis and simulation results indicate that the performance and security of the proposed encryption scheme can encrypt the image effectively and resist various typical attacks.

Keywords: spatiotemporal chaos, ergodic matrix, standard map, image encryption

PACS: 05.45.–a, 05.45.Gg, 05.45.Ra **DOI:** 10.1088/1674-1056/22/8/080503

1. Introduction

With the rapid development of network technology and information technology, the digital image has become an important approach to acquiring information, and a part of people's lives. In order to ensure the information security, image encryption technology has aroused much interest in both research and application fields. According to Shannon's two fundamental principles of secure communication: confusion and diffusion,[1] traditional symmetric ciphers such as data encryption standard (DES) and advanced encryption standard (AES) are designed with good confusion and diffusion properties.[2] However, they are not suitable for image encryption due to some intrinsic properties of images such as bulk data volume and high pixel correlation. As a ubiquitous physical process in nature, chaos has become an immense research direction to cryptography, particularly its application in the field of digital image encryption due to its analogous cryptography characteristics, such as extreme sensitivity to initial conditions and system parameters, random-like and ergodicity.[3]

In 1998, Fridrich first applied a chaos-based encryption method to image encryption.[4] Inspired by Fridrich in Ref. [4], several chaos-based image encryption schemes have been proposed,[5–14] but most of them are one-dimensional or multiple one-dimensional chaotic systems. For example, Pareek *et al.*[6,7] have proposed two different encryption schemes based on one and more one-dimensional chaotic maps. In 2008 and 2009, Li *et al.*[8,9] analyzed and pointed out the weak keys and existing known-plaintext attack or chosen-plaintext attack against the algorithm. In 2010, Patidar *et al.* proposed a modified version of the original scheme in Ref. [11] to resist Rhouma *et al.*'s attack presented in Ref. [12]. However, Li *et al.*[13] found that the improved scheme is still insecure against a known/chosen-plaintext attack and has insufficient sensitivity to the change of plain-image. These encryption schemes encountered small key space, low speed, and low security. Therefore, to ensure the security of a digital image, the cycle of two or more times must be required in a permutation–diffusion process so that a tiny change in the plain-image can spread over the whole cipher-image, and it also is very vulnerable to known-plaintext attack or chosen-plaintext attack. To overcome these drawbacks, in 2011, Akhavan *et al.*[14] introduced an image algorithm based on the combination of a one-dimensional polynomial chaotic map and a piecewise nonlinear chaotic map, Sathishkumar and Bhoopathy[15] proposed a novel image encryption algorithm using pixel shuffling and base 64 encoding based chaotic block cipher, and Ye[16] presented a new chaos-based image encryption scheme with an efficient permutation–diffusion mechanism, in which permuting the position of image pixels incorporates with changing the gray values of image pixels to confuse the relationship between cipher-image and plain-image. Although these algorithms improve the security to some extent, the efficiency of practical application is not high. Then in 2011, two chaos-based bit-level permutation schemes for image encryption were first proposed,[17,18] but any permutation-only encryption scheme was broken by Ref. [19]. In addition, Wang *et*

*Project supported by the National Natural Science Foundation of China (Grant Nos. U0735004 and 60972133), the Natural Science Foundation of Guangdong Province, China (Grant No. 05006593), the Project Team for Natural Science Foundation of Guangdong Province, China (Grant No. 9351064101000003), Energy Technology Key Laboratory Project of Guangdong Province, China (Grant No. 2008A060301002), and the Fundamental Research Funds for the Central Universities, China (Grant No. X2dXD2116370).

†Corresponding author. E-mail: luo.yuling@mail.scut.edu.cn

© 2013 Chinese Physical Society and IOP Publishing Ltd

al.[20] proposed a fast image encryption algorithm with the architecture of combining permutation and diffusion. Meanwhile, Ye[21] presented a modeled image encryption scheme, in which a generalized circulant matrix is employed together with a dynamical chaotic system. Wang and Jin,[22] and Wang and Teng[23] proposed two new image encryption schemes based on the ergodicity of chaos trajectory.

Compared with simple chaotic maps, the spatiotemporal chaos greatly enhances the spatiotemporal complexity of nonlinear dynamics and mixture, which can effectively cope with the computer precision problem. Therefore, some image encryption schemes based on the spatiotemporal-chaos system were proposed,.[24–28] Besides, Yuen and Wong[29] proposed the chaos-based encryption for fractal image coding, and Wang *et al.*[30] proposed an image encryption according to the property of delayed fractional-order chaotic logistic system. For these encryption schemes, most of their keys are usually irrelevant to the plain-image information, so they are insecure enough in the sense of preventing known/chosen-plaintext attack.

For the above reasons, we propose a self-adapting image encryption algorithm based on the spatiotemporal chaotic system and standard ergodic matrix in this paper. Compared with the conventional permutation–diffusion process, the proposed scheme is a reverse process. The diffusion process is first implemented in order to make the plain-image pixels more random-like, and then the corresponding ergodic matrices are generated for the following permutation process. Particularly, the initial conditions of the spatiotemporal-chaos system are closely related to the plain-image, which is iterated to generate random matrices for diffusion. Then we use the index order matrices generated from sorting diffusion-image pixels to shuffle the diffusion-image. Finally, we rearrange all permutation-modules to acquire the final encrypted image. Theoretical analysis and simulation results show that the performance and security of the proposed encryption scheme can encrypt the image effectively and resist various typical attacks.

2. Spatiotemporal chaos and ergodic matrix

2.1. Spatiotemporal chaos system

A coupled map lattice (CML) is often used to construct the basic model of a spatiotemporal chaotic system.[24–28] Here, a CML is used as the model of a spatiotemporal chaos system, which was introduced by Kaneko.[31] Its dynamic formula is expressed as

$$\begin{aligned} x_{n+1}(i) &= (1-\varepsilon) f(x_n(i)) \\ &\quad + 0.5\varepsilon[f(x_n(i-1)) + f(x_n(i+1))], \end{aligned} \tag{1}$$

where n is the time index, i $(i = 1, 2, \ldots, L)$ the lattice site index, $\varepsilon \in (0,1)$[31] the coupling constant, and $x_n(i)$ the state variable for the i-th site at time n. The periodic boundary condition $x_n(i) = x_n(i+L)$ is assumed. It is a dynamical system with discrete-time, discrete-space, and continuous states, consisting of nonlinear maps located on the lattice sites. Each lattice map is coupled with other lattice maps in terms of certain coupling coefficients. Taking the logistic map as function $f(x_n(i))$, it is defined as

$$f(x_n(i)) = 1 - \mu x_n^2(i), \tag{2}$$

where μ is the parameter. When $\mu \in [1.4011551891, 2]$, the logistic system is in a chaotic state shown in Fig. 1(a). Therefore, system (1) becomes,

$$\begin{aligned} x_{n+1}(i) &= (1-\varepsilon)(1-\mu x_n^2(i)) \\ &\quad + \varepsilon/2((1-\mu x_n^2(i-1)) + (1-\mu x_n^2(i+1))). \end{aligned} \tag{3}$$

If we discretionarily choose $\mu = 1.99$, $\varepsilon = 0.1$, and initial values $x_0(i) \in (-1, 1)$, under the periodic boundary condition, we can gain the chaotic behavior of 30 lattices in 200 time-step iteration, demonstrated in Fig. 1(b).

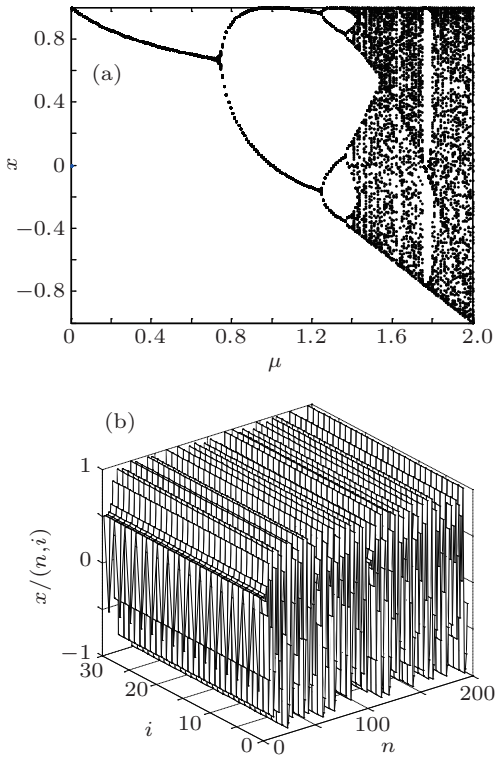


Fig. 1. (a) Bifurcation diagram of logistic map (2), and (b) chaotic behavior of the spatiotemporal system (3).

According to Golomb's three postulates for pseudo-random sequence,[24] ideal chaotic sequence should have such statistical characteristics as follows: its average value is zero, the auto-correlation is of delta function, and the mutual correlation is zero. It is easy to see from Figs. 2(a) and 2(b) that their correlation characteristics are ideal.

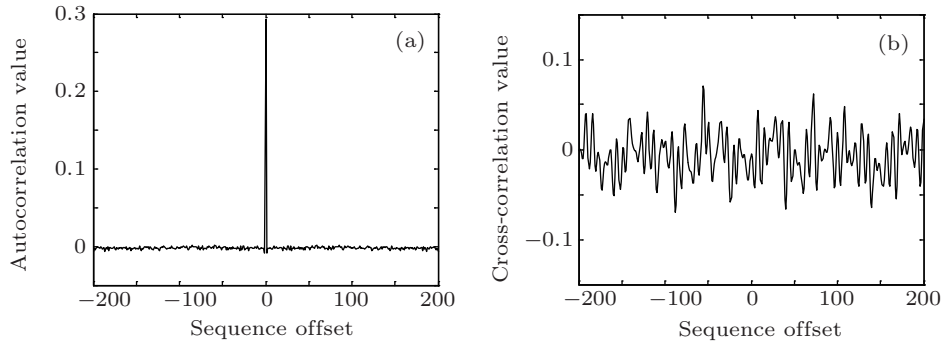


Fig. 2. Correlation characteristics of the spatiotemporal chaos of (a) auto-correlation and (b) cross-correlation.

2.2. Ergodic matrix

Definition 1[32] an $m \times n$ matrix R is defined as an ergodic matrix, if each element of matrix R is in the set of $\{1,2,\ldots,mn\}$, and $r(i,j) = r(i',j')$ if and only if $i = i'$, $j = j'$. We note that $r_{(i-1)n+j} = r(i,j)$. General form of main ergodic matrix is as follows:

$$\begin{pmatrix} 1 & 2 & 3 & \ldots & n \\ n+1 & n+2 & n+3 & \ldots & 2n \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ (m-1)n+1 & (m-1)n+2 & (m-1)n+3 & \ldots & mn \end{pmatrix}_{m\times n}. \tag{4}$$

16	3	13	14
11	12	6	9
7	8	10	4
5	15	1	2

(a)

6	7	10	11
5	8	9	12
4	3	14	13
1	2	15	16

(b)

Fig. 3. Two typical ergodic matrices: (a) Magic square ergodic matrix and (b) Hilbert ergodic matrix.

There are two typical ergodic matrices, i.e., Magic square ergodic matrix and Hilbert ergodic matrix, which are shown in Fig. 3, respectively.

The ergodic matrix is mainly applied to the shuffled process in transforming space geometry location for a digital image. For example, we can realize the ergodicity of a digital image matrix through an ergodic pattern represented by $R_{m\times n}$ as shown in Eq. (5).

$$Q_{4\times4} = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{pmatrix} \xrightarrow{R=\begin{pmatrix} 12 & 4 & 7 & 16 \\ 2 & 9 & 11 & 14 \\ 15 & 5 & 10 & 1 \\ 3 & 8 & 13 & 6 \end{pmatrix}} Q'_{4\times4} = \begin{pmatrix} q_{34} & q_{21} & q_{41} & q_{12} \\ q_{32} & q_{44} & q_{13} & q_{42} \\ q_{22} & q_{33} & q_{23} & q_{11} \\ q_{43} & q_{24} & q_{31} & q_{14} \end{pmatrix}. \tag{5}$$

As described in Eq. (5), the rule is that Q' is obtained by arranging Q in light of the matrix R. Here, assuming

$$R_{i,j} = \kappa (i \in [1,4], j \in [1,4]),$$

if $1 \leq \kappa \leq 4$, we place the element $q_{i,j}$ of Q at the location $(1,\kappa)$ of Q'; and if $4 \leq \kappa \leq 8$, we place the element $q_{i,j}$ of Q at the location $(2,\kappa-4)$ of Q'; and if $9 \leq \kappa \leq 12$, we place the element $q_{i,j}$ of Q at the location $(3,\kappa-8)$ of Q'; and if $13 \leq \kappa \leq 16$, we place the element $q_{i,j}$ of Q at the location $(4,\kappa-12)$ of Q', i.e., when $R_{3,4} = 1$, $q_{3,4}$ is placed at the location $(1,1)$ of Q'.

3. Image encryption algorithm

The overall architecture of the proposed diffusion–permutation image cryptosystem is shown in Fig. 4.

There are mainly two stages in the encryption process. First, we utilize a pseudorandom matrix generated by the spatiotemporal chaos to modify the pixel value, so that a tiny change in one pixel can spread over other pixels. Then diffusion-image pixels are shuffled sequentially according to standard ergodic matrices generated from its index order sequences. The two processes will be introduced in the following figure.

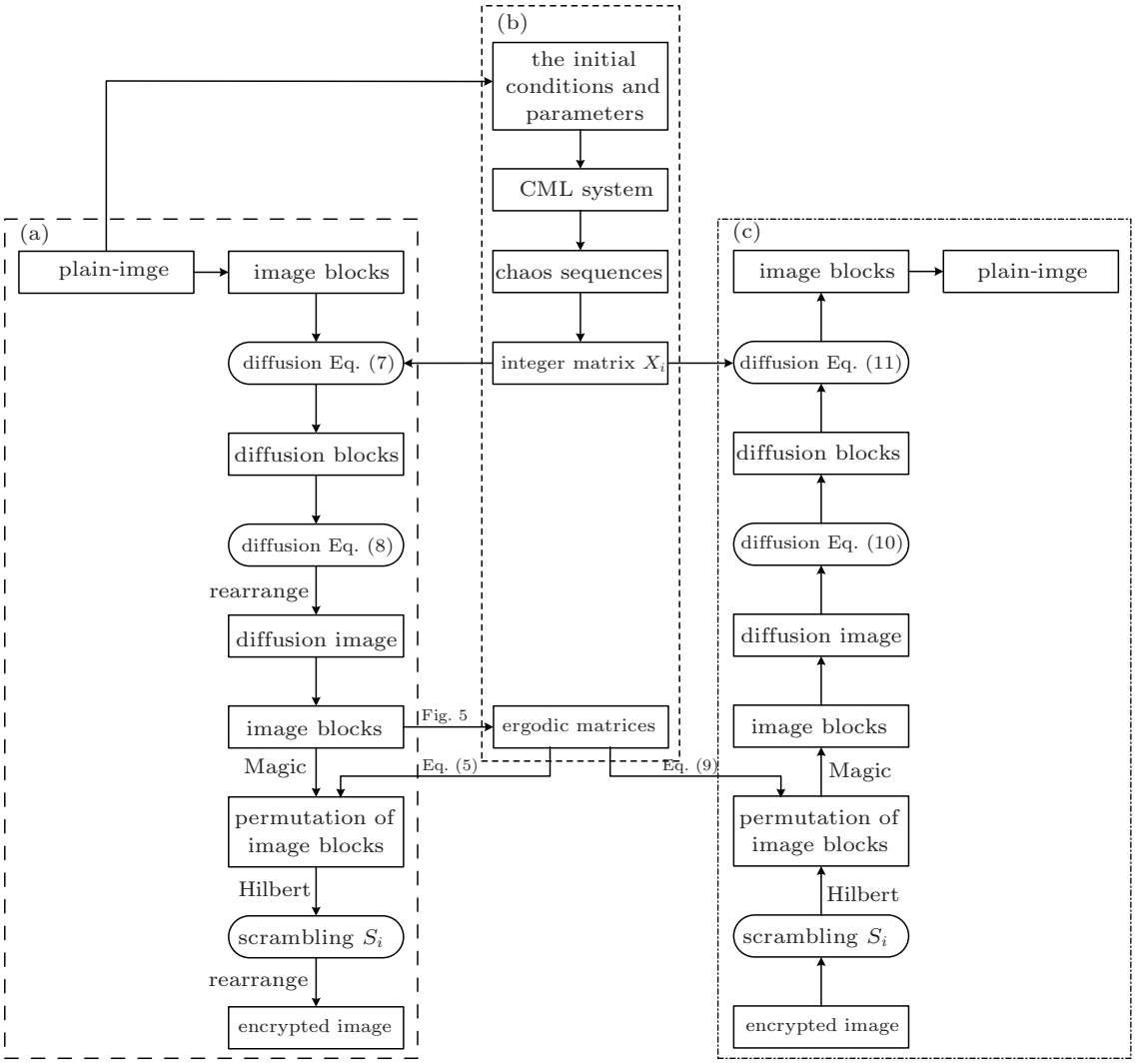


Fig. 4. A block diagram of the proposed scheme of (a) the encryption process, (b) the key space, and (c) the decryption process.

3.1. Diffusion of the plain-image

A good diffusion should have a key stream strongly related to the plain-image. When we encrypt different plain-images, the encryption scheme can generate different key streams. Without loss of generality, we assume that the size of plain-image I is $M \times N$, and the pixel value ranges from 0 to 255. In this paper, we take the mean value of pixels in each column as initial state of the spatiotemporal chaos. The diffusion process is represented as follows.

Step 1 Preprocessing: if the related-coefficient of the plain-image in vertical direction is larger than that in horizontal direction, transpose the plain-image I; otherwise keep unchanged.

Step 2 Calculate the mean value $\overline{x(j)}$ $(j = 1,2,\ldots,N)$ of pixels for each column of the image which is preprocessed in Step 1. Then place the columns $\{i, N/L+i, 2N/L+i, \ldots, N-N/L+i\}$ $(i = 1,2,\ldots,N/L)$ together as the i-th module I_i of the image, and let the row hold the line. So the plain-image is divided into N/L modules, where L is the max lattice number of the spatiotemporal chaos;

Step 3 Set $i = 1$;

Step 4 Set $x_0(jj) = \{[\overline{x(j)} - \text{floor}(\overline{x(j)})] + [\overline{x(j+1)} - \text{floor}(\overline{x(j+1)})]\}/2$,

$$(j = i, N/L+i, 2N/L+i, \ldots, N-N/L+i; jj = 1,2,\ldots,L);$$

Step 5 Iterate Eq. (3) N_0 times to avoid harmful effect of transitional procedure, and continue to iterate it M times to obtain an $M \times L$ matrix X. Transform each output number $x_n(jj) \in [0,1]$ into integer $x'_n(jj) \in [0,255]$ according to Eq. (6),

$$x'_n(jj) = \text{mod}\,(x_n(jj) \times 10^{10}, 256); \tag{6}$$

Step 6 Use the transformed integer matrix X' to diffuse the corresponding block I_i. The detailed encryption formula is shown in Eq. (7), where E_i is corresponding encrypted module matrix

$$E_i = I_i \oplus X' \oplus I_{i-1} \oplus E_{i-1} \quad \text{and} \quad E_0 = I_0 = 0; \tag{7}$$

Step 7 Let $i = i+1$, and return to Step 4 until $i = N/L$;

Step 8 The above diffusion process implies that each module only affects its latter modules, but has no influence on its front modules with a gray value change. As a remedy, we add a reverse diffusion process as a supplement to the above diffusion process. The operation is executed according to Eq. (8) as follows:

$$E_i = E_i \oplus E_{i+1} \quad (i = N/L, N/L-1, \ldots, 2, 1) \text{ and } E_{N/L+1} = 255. \tag{8}$$

After diffusion, all modules are rearranged to reconstruct the encrypted image E, from left to right.

3.2. Self-adapting permutation of the cipher-image

Before permutation, the cipher-image E is divided into $m \times n$ blocks of the same size, from left to right, and from top to bottom, so the size of each block is $M/m \times N/n$. The detailed permutation procedure is described in the following.

Step 1 Let $A = \{A_{1\times1}, A_{1\times2}, \ldots, A_{i\times j}, \ldots, A_{m\times n}\}$ $(i \in [1,m],\ j \in [1,n])$ denote the $m \times n$ blocks of the cipher-image E, then all elements of each block $A_{i\times j}$, from left to right, and from top to bottom, are sorted in the ascending order. Then we can obtain a new set $B = \{B_{1\times1}, B_{1\times2}, \ldots, B_{i\times j}, \ldots, B_{m\times n}\}$ $(i \in [1,m], j \in [1,n])$ and its corresponding index order set $R = \{R_{1\times1}, R_{1\times2}, \ldots, R_{i\times j}, \ldots R_{m\times n}\}$ $(i \in [1,m],\ j \in [1,n])$, that is, $B_{i\times j} = A_{i\times j}(R_{i\times j})$ as shown in Fig. 5.

$$A_{i\times j} = \begin{bmatrix} 120 & 100 & 20 & 221 \\ 250 & 250 & 97 & 120 \\ 102 & 121 & 35 & 132 \\ 114 & 201 & 201 & 33 \end{bmatrix} \xrightarrow{\text{sorting}} B_{i\times j} = \begin{bmatrix} 20 & 33 & 35 & 97 \\ 100 & 102 & 114 & 120 \\ 120 & 121 & 132 & 201 \\ 201 & 221 & 250 & 250 \end{bmatrix} \Rightarrow R_{i\times j} = \begin{bmatrix} 8 & 5 & 1 & 14 \\ 15 & 16 & 4 & 9 \\ 6 & 10 & 3 & 11 \\ 7 & 12 & 13 & 2 \end{bmatrix}$$

Fig. 5. Illustration of sorting image pixels' values and generating the corresponding index order matrix R.

Step 2 Use $R_{i,j}$ to shuffle the cipher-image blocks $A_{i',j'}$ according to the discipline shown in Eq. (5), respectively. Here, the congruent relationship between locations (i,j) and (i',j') is determined by using Magic square ergodic matrix MG, namely, assuming $MG_{ii,jj} = k$ $(ii \in [1,4], jj \in [1,4])$, then $i' = ii, j' = jj$ and if $1 \le k \le 4$, $i = 1, j = k$; if $5 \le k \le 8$, $i = 2, j = k-4$; if $9 \le k \le 12$, $i = 3, j = k-8$; if $13 \le k \le 16$, $i = 4, j = k-12$. In other words, we use $R_{4\times4}$ to shuffle $A_{1\times1}$, $R_{1\times3}$ to do $A_{1\times2}$, etc. as shown in Fig. 3(a). Therefore, we obtain the corresponding scrambled blocks $S_{1\times1}, S_{1\times2}, \ldots, S_{m\times n}$ $(i \in [1,m], j \in [1,n])$.

Step 3 Shuffle the scrambled blocks $S_{1\times1}, S_{1\times2}, \ldots, S_{m\times n}$ in accordance with Hilbert ergodic matrix HB shown in Fig. 3(b), i.e., assuming $HB_{ii,jj} = k$ $(ii \in [1,4], jj \in [1,4])$, if $1 \le k \le 4$, place $S_{1,k}$ at the location of coordinates (ii, jj); if $5 \le k \le 8$, place $S_{2,k-4}$ at the location of coordinates (ii, jj); if $9 \le k \le 12$, place $S_{3,k-8}$ at the location of coordinates (ii, jj); if $13 \le k \le 16$, place $S_{4,k-12}$ at the location of coordinates (ii, jj). For instance, we place $S_{2\times2}$ at the location of coordinates $(1,1)$, $S_{2\times3}$ at the location of coordinates $(1,2)$, etc. Then we obtain the final scrambled image S.

The decryption procedure is depicted in Fig. 4(c), which is similar to that of the encryption procedure illustrated above but in the reversed order. Attention should be paid to the following steps.

(i) The decrypted scrambling image can be obtained by using the inverse process of Hilbert ergodic matrix HB, i.e., assuming $HB_{ii,jj} = k$ $(ii \in [1,4], jj \in [1,4])$, if $1 \le k \le 4$, place $S_{ii,jj}$ at the location of coordinates $(1,k)$; if $5 \le k \le 8$, place $S_{ii,jj}$ at the location of coordinates $(2,k-4)$; if $9 \le k \le 12$, place $S_{ii,jj}$ at the location of coordinates $(3,k-8)$; if $13 \le k \le 16$, place $S_{ii,jj}$ at the location of coordinates $(4,k-12)$.

(ii) The decrypted image blocks of permutation can be obtained by using the inverse of Eq. (5) in the encryption process as follows:

$$Q_{4\times4} = \begin{pmatrix} q_{34} & q_{21} & q_{41} & q_{12} \\ q_{32} & q_{44} & q_{13} & q_{42} \\ q_{22} & q_{33} & q_{23} & q_{11} \\ q_{43} & q_{24} & q_{31} & q_{14} \end{pmatrix} \xrightarrow{R=\begin{pmatrix} 12 & 4 & 7 & 16 \\ 2 & 9 & 11 & 14 \\ 15 & 5 & 10 & 1 \\ 3 & 8 & 13 & 6 \end{pmatrix}} Q'_{4\times4} = \begin{pmatrix} q_{11} & q_{12} & q_{13} & q_{14} \\ q_{21} & q_{22} & q_{23} & q_{24} \\ q_{31} & q_{32} & q_{33} & q_{34} \\ q_{41} & q_{42} & q_{43} & q_{44} \end{pmatrix}. \tag{9}$$

(iii) The decrypted image and image blocks of diffusion can be obtained by using the inverse of Eqs. (8) and (7) in the encryption process as follows:

$$E_i = E_i \oplus E_{i+1} \quad (i = 1, 2, \ldots, N/L-1, N/L) \text{ and } E_{N/L+1} = 255, \tag{10}$$

$$I_i = E_i \oplus X' \oplus I_{i-1} \oplus E_{i-1} \text{ and } E_0 = I_0 = 0. \tag{11}$$

4. Experimental results

The plain-image 'Lena' with a size of 256×256 is shown in Fig. 6(a) and the corresponding histogram is shown in

Fig. 6(b). If $L = 32$, $\mu = 1.99$, $\varepsilon = 0.6$, and $x_0(j) = ((g_j - \text{floor}(g_j)) + (g_{j+1} - \text{floor}(g_{j+1})))/2$, where g_j $(j = 1,2,\ldots,256)$ is the mean value of pixels of each column and $g_{N+1} = 0.123$ is assured, we first encrypt the transposed image according to the detailed illustration in Subsection 3.1, then shuffle the diffusion-image after dividing it into 16 $(m = n = 4)$ blocks according to the detailed illustration in Subsection 3.2. The final encrypted image is shown in Fig. 6(c), and Fig. 6(d) is the histogram of the final encrypted image. We can observe that the histogram of the encrypted image is fairly uniform and is significantly different from that of the plain-image.

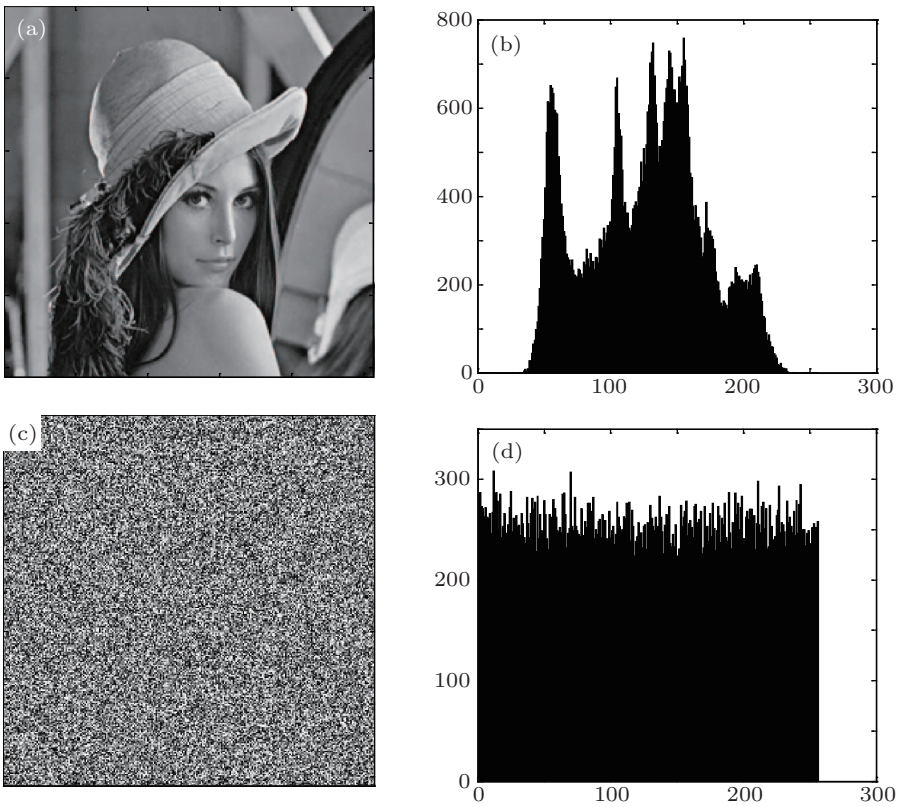


Fig. 6. Experimental results of image encryption, showing (a) plain-image, (b) histogram of the plain-image, (c) encrypted image, and (d) histogram of the encrypted image.

5. Performance and security analysis

In this section, the proposed image encryption scheme is analyzed in detail. We have several tests to check the security of the cryptosystem. The results show that the scheme has higher security, bigger key space, and can resist various typical attacks, such as statistical analysis, differential attack, brute-force attack, etc.

5.1. Key space and key sensitivity analysis

A good encryption scheme should be sensitive to secret keys, and have a large enough key space to make any brute-force attacks infeasible.[5] As shown in Fig. 4(b) in this paper, parameters ε, μ, initial values $x_0(j)$ $(j = 1,2,\ldots,N)$ and ergodic matrices can be used as the keys. In particular, from Subsection 3.1, we can see that initial values $x_0(j)$ $(j = 1,2,\ldots,N)$ of chaos system are obtained by calculating $x_0(j) = \{[\overline{x(j)} - \text{floor}(\overline{x(j)})] + [\overline{x(j+1)} - \text{floor}(\overline{x(j+1)})]\}/2$, where $\overline{x(j)}$ is the mean value of pixels for each column of the image. Moreover, ergodic matrices are obtained by using the rule shown in Fig. 5 for the diffused image blocks. So the initial values and ergodic matrices are relevant to the original plain-image data. In addition, parameters $\varepsilon \in (0,1)$ and $\mu \in [1.4011551891,2]$ are determined by the chaos system which is described in Subsection 2.1. If the precision is 10^{-15}, the key space size is $10^{15\times(2+N)}$. From Table 1, it can be seen that the key space is large enough. In addition, several key sensitivity tests are performed. Figure 7(a) shows the decrypted image of 'Lena' with similar encryption keys, figures 7(b) and 7(c) show the decrypted images with an error parameter $\mu = 1.99 + 1e-15$ and $\varepsilon = 0.6 + (1e-15)$, respectively, and figure 7(d) shows the difference between them. Moreover, the rates of change between two encrypted images, which are obtained by encrypting the same image with a slight difference in parameter, are listed in Table 1.

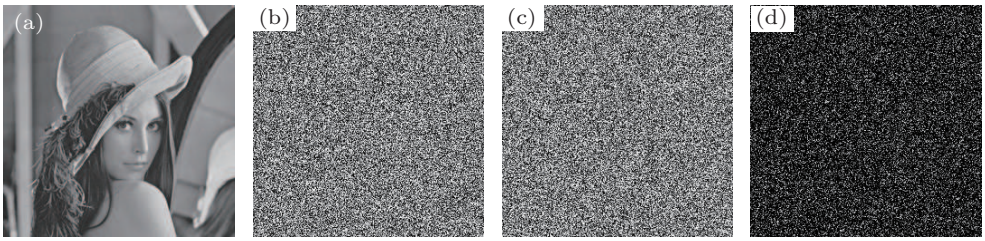


Fig. 7. Key sensitivity analyses, showing (a) decrypted image with correct keys, (b) decrypted image with wrong key $\mu = 1.99 + 1e-15$, (c) decrypted image with wrong key $\varepsilon = 0.6 + (1e-15)$, and (d) difference image between panels (b) and (c).

From Fig. 7 and Table 1, we can see that the proposed algorithm is strongly sensitive to the key.

Table 1. Results of test for key sensitivity.

Change for keys	Rate of change between two encrypted images (%)
$\mu = 1.99 \pm 10^{-15}$, $\varepsilon = 0.6$, $N_0 = 100$	99.62
$\mu = 1.99$, $\varepsilon = 0.6 \pm 10^{-16}$, $N_0 = 100$	99.59
$\mu = 1.99$, $\varepsilon = 0.6$, $N_0 = 100 \pm 1$	99.58
$\mu = 1.99 \pm 10^{-16}$, $\varepsilon = 0.6 \pm 10^{-16}$, $N_0 = 100 \pm 1$	99.63

5.2. Statistical analysis

As is well known, there is high correlation among adjacent pixels for an ordinary image in either horizontal, or vertical or diagonal direction. So, we must reduce the correlation between two adjacent pixels in the cipher-image to withstand statistical attack. According to Shannon's theory, diffusion and confusion can be employed in the cryptosystem to frustrate the powerful attacks based on statistical analysis.[1] In the proposed cryptosystem, a CML map is used to generate a chaotic matrix with pseudorandom properties. Then this chaotic matrix is used to change the image pixel values sequentially. Thus, the diffusion effect is introduced. Also some standard matrices are used to shuffle the plain-image, which can be considered as a confusion process.

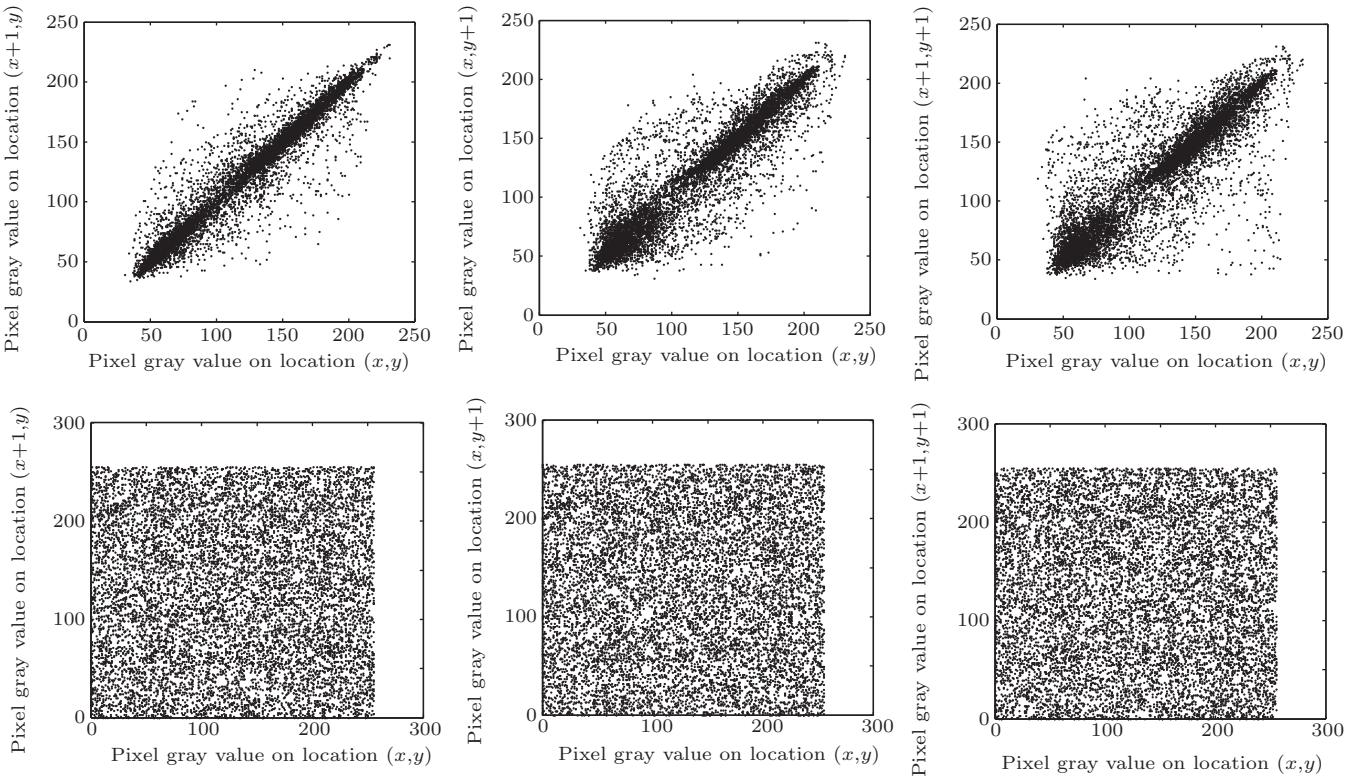


Fig. 8. Distributions of two adjacent pixels in the plain-image and in the cipher-image.

The operations are carried out to test the correlation between two adjacent pixels in vertical, horizontal, and diagonal directions of the cipher-image, respectively. First, we randomly select 10^4 pairs of two adjacent pixels in three directions, and plot the corresponding distributions. The results are shown in Fig. 8. Then, we calculate the correlation coefficients of each pair using the following formula:

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)D(y)}}, \tag{12}$$

where x and y are gray-scale values of two adjacent pixels, and

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2,$$
$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)).$$

The calculation results are shown in Table 2.

Table 2. Correlation coefficients of two adjacent pixels in two images.

	Horizontal	Vertical	Diagonal
Plain-image	0.9693	0.9700	0.9105
Cipher-image	–0.0047	–0.0014	–0.0087

From Fig. 8 and Table 2, we can see that the proposed algorithm is close to zero correlation.

5.3. Differential attack

If a tiny change in the plain-image causes a significant difference in cipher-image, differential attack would become ineffective.[29] The number of pixels change rate (NPCR) and united average changing intensity (UACI) are usually used to measure this capability quantitatively, which are defined as follows:[5]

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%, \tag{13}$$
$$\mathrm{UACI} = \frac{1}{M \times N}\left[\sum_{i,j}\frac{|E_1(i,j) - E_2(i,j)|}{255}\right] \times 100\%, \tag{14}$$

where M and N are the width and height of the cipher-image, $E_1(i,j)$ and $E_2(i,j)$ are the two cipher-images whose corresponding plain-images have only one-pixel difference, respectively. The difference array $D(i,j)$ is determined by the following rule: when $E_1(i,j) \neq E_2(i,j)$, then $D(i,j) = 1$; otherwise it is 0. The values of the quantitative measures of NPCR and UACI are calculated at different locations, respectively, and they are shown in Table 3. Compared with the other results cited from different published papers in Table 4, our simulation results show that the encryption scheme is more sensitive to the plain-image and can effectively resist differential attacks.

Table 3. Results of NPCR and UACI at different locations.

Position	(1,1)	(50,100)	(50,150)	(50,200)
Pixel value change	160→159	134→135	210→209	146→145
NPCR/%	99.61	99.59	99.63	99.63
UACI/%	33.61	33.62	33.56	33.54
Position	(100,50)	(100,100)	(100,150)	(100,200)
Pixel value change	100→101	104→105	179→178	58→59
NPCR/%	99.59	99.59	99.63	99.59
UACI/%	33.61	33.60	33.64	33.60
Position	(150,50)	(150,100)	(150,150)	(150,200)
Pixel value change	99→100	113→112	131→130	139→138
NPCR/%	99.58	99.61	99.61	99.61
UACI/%	33.41	33.55	33.55	33.54
Position	(200,50)	(200,100)	(200,150)	(200,200)
Pixel value change	49→50	53→52	167→168	127→128
NPCR/%	99.62	99.63	99.63	99.62
UACI/%	33.70	33.58	33.70	33.69

Table 4. Comparison of several performances.

	Key space	Entropy	NPCR/%	UACI/%
Our algorithm	$10^{15\times(2+N)}$	7.9968	99.62	33.60
Ref. [5]	10^{38}	–	< 50.4	< 25.3
Ref. [7]	–	–	99.61	33.46
Ref. [14]	2^{180}	7.9978	–	–
Ref. [16]	–	–	< 99.81	< 46.37
Ref. [17]	10^{42}	7.9993	< 85.00	< 30.00
Ref. [18]	–	7.9880	–	–
Ref. [20]	–	–	44.27	14.87
Ref. [22]	10^{66}	7.9983	99.62	33.49
Ref. [24]	$10^{70\ mn}$	7.9965	–	–
Ref. [25]	$10^{14\times16}$	–	99.30	–
Ref. [26]	$10^{14\times(2+L)}$	7.9954	–	–
Ref. [27]	2^{192}	7.9970	–	–
Ref. [28]	–	–	< 99.61	< 14.47
Ref. [29]	2^{143}	–	–	–
Ref. [30]	–	7.9702	–	–

5.4. Known-plaintext and chosen-plaintext attacks

According to Kirchhoff's principle,[7,26,27] known-plaintext attack is that the opponent possesses a string of plaintext and corresponding cipher-text, while chosen-plaintext is the opponent has obtained temporary access to the encryption machinery, so he can choose a plaintext string to construct corresponding cipher-text string. They are the two most powerful classical typical attacks. In this proposed cryptosystem, the initial conditions of chaotic system and block permutation matrixes are different if we choose a different plain-image. Moreover, the diffusion-image is not only related to the former plain-image but also related to the latter encrypted image, that is, $E_i = I_i \oplus X \oplus I_{i-1} \oplus E_{i-1} \oplus E_{i+1}$. Therefore, the proposed cryptosystem can effectively resist these two attacks and then can resist other conventional attacks.

5.5. Information entropy analysis

Apart from several common security analyses mentioned above, information entropy is also a test to measure the security level of a cryptosystem. According to Shannon's theory,[1] the entropy $H(m)$ of a message source m can be calculated as follows:

$$H(m) = \sum_{i=0}^{2^N-1} p(m_i) \log \frac{1}{p(m_i)}, \tag{15}$$

where $p(m_i)$ is the probability of symbol m_i, N is the number of bits for each symbol m_i. Ideally, for an encrypted image with 256 gray levels, its information entropy should be equal to 8 bits. By calculating, the information entropy of the final encrypted image is 7.9968 bits, and it is very close to the theoretical value 8. This means that the information leakage may be negligible in the encryption process and the encrypted scheme is secure against the entropy attack.[27]

6. Conclusions

In this paper, a block-based self-adapting image encryption algorithm is presented, which contains diffusion and permutation processes. The basic idea is that the plain-image is divided into different blocks successively, and the corresponding blocks are encrypted by the spatiotemporal chaotic system and then they are shuffled by virtue of the ergodicity of standard ergodic matrices. In particular, both the initial conditions of the spatiotemporal chaos for diffusion and the standard ergodic matrices for permutation are extremely related to the plain-image. In addition, after diffusion and permutation, all modules are rearranged to generate the final encrypted image. Theoretical analysis and simulation results indicate that the performance and security of the proposed encryption scheme can encrypt the image effectively and resist various typical attacks. In the future, better image processing schemes combined with compression and encryption will be focused on.

References

[1] Shannon C E 1949 *Bell System Technology Journal* **28** 656
[2] Schneier B 1995 *Cryptography: Theory and Practice* (Boca Raton (FL): CRC Press)
[3] Brown R and Chua L O 1996 *Int. J. Bifur. Chaos* **6** 219
[4] Fridrich J 1998 *Int. J. Bifur. Chaos* **8** 1259
[5] Chen G R, Mao Y B and Chui C K 2004 *Chaos, Solitons and Fractals* **21** 749
[6] Pareek N K, Patidar V and Sud K K 2006 *Image and Vision Computing* **24** 926
[7] Pareek N K, Patidar V and Sud K K 2005 *Commun. Ninlinear Sci. Numer. Simul.* **10** 715
[8] Li C Q, Li S J, Asim M, Nunez J, Alvarez G and Chen G R 2009 *Image and Vision Computing* **27** 1371
[9] Li C Q, Li S J, Alvarez G, Chen G R and Lo Kwok-Tung 2008 *Chaos, Solitons and Fractals* **37** 299
[10] Patidar V, Pareek N K and Sud K K 2010 *Commun. Nonlinear Sci. Numer. Simul.* **14** 2755
[11] Patidar V, Pareek N K and Sud K K 2009 *Commun. Nonlinear Sci. Numer. Simul.* **14** 3056
[12] Rhouma R, Solak E and Belghith S 2010 *Commun. Nonlinear Sci. Numer. Simul.* **15** 1887
[13] Li C Q, Li S J and Lo K T 2011 *Commun. Nonlinear Sci. Numer. Simulat.* **16** 837
[14] Akhavan A, Samsudin A and Akhshani A 2011 *Journal of the Franklin Institute* **348** 1797
[15] Sathishkumar G A and Bhoopathy Bagan K 2011 *Wseas Transactions on Computers* **10** 169
[16] Ye R S 2011 *Opt. Commun.* **284** 5290
[17] Zhu Z L, Zhang W, Wong K W and Yu H 2011 *Inf. Sci.* **181** 1171
[18] Fu C, Lin B B, Miao Y S, Liu X and Chen J J 2011 *Opt. Commun.* **284** 5415
[19] Li C Q and Lo K T 2011 *Signal Processing* **91** 949
[20] Wang Y, Wong K W, Liao X F and Chen G R 2011 *Appl. Soft Comput.* **11** 514
[21] Ye G D 2011 *The Imaging Science Journal* **59** 183
[22] Wang X Y and Jin C Q 2012 *Opt. Commun.* **285** 412
[23] Wang X Y and Teng L 2012 *Chin. Phys. B* **21** 020504
[24] Sun F Y, Lü Z W and Liu S T 2010 *Opt. Commun.* **283** 2066
[25] Sun F Y and Lü Z W 2011 *Chin. Phys. B* **20** 040506
[26] Wang X Y and Teng L 2011 *Nonlinear Dynamics* **67** 365
[27] Wang X Y, Zhang N, Ren X L and Zhang Y L 2011 *Chin. Phys. B* **20** 020507
[28] Banerjee S, Rondoni L, Mukhopadhyay S and Misra A P 2011 *Opt. Commun.* **284** 2278
[29] Yuen C H and Wong K W 2012 *Chin. Phys. B* **21** 010502
[30] Wang Z, Huang X, Li N and Song X N 2012 *Chin. Phys. B* **21** 050506
[31] Kaneko K 1985 *Prog. Theor. Phys.* **74** 1033
[32] Cheng G, Zhao X Y and Li J L 2005 *Journal of Software* **16** 1975