Information Sciences 297 (2015) 80–94

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

2D Sine Logistic modulation map for image encryption

Zhongyun Hua, Yicong Zhou *, Chi-Man Pun, C.L. Philip Chen

Department of Computer and Information Science, University of Macau, Macau 999078, China

ARTICLE INFO

Article history:
Received 22 April 2014
Received in revised form 28 August 2014
Accepted 6 November 2014
Available online 15 November 2014

Keywords:
2D Sine Logistic modulation map
Chaotic magic transform
Image encryption

ABSTRACT

Because of the excellent properties of unpredictability, ergodicity and sensitivity to their parameters and initial values, chaotic maps are widely used in security applications. In this paper, we introduce a new two-dimensional Sine Logistic modulation map (2D-SLMM) which is derived from the Logistic and Sine maps. Compared with existing chaotic maps, it has the wider chaotic range, better ergodicity, hyperchaotic property and relatively low implementation cost. To investigate its applications, we propose a chaotic magic transform (CMT) to efficiently change the image pixel positions. Combining 2D-SLMM with CMT, we further introduce a new image encryption algorithm. Simulation results and security analysis demonstrate that the proposed algorithm is able to protect images with low time complexity and a high security level as well as to resist various attacks.

1. Introduction

As traditional dynamical systems, chaotic maps have the properties of unpredictability, ergodicity, and sensitivity to their parameter(s) and initial value(s) [31]. They can generate different random sequences with different settings of parameters or initial values. Thus chaotic maps attract many researchers' attentions and have been widely used in different applications [13,21,24,33,35]. Especially in security applications, chaotic maps show excellent performance. For example, a symmetric image encryption scheme using the Arnold cat map was proposed by Zhu et al. [40], a symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice was proposed by Zhang and Wang [34], and a new 1D chaotic system for image encryption was proposed by Zhou et al. [36].

Existing chaotic maps can be classified into two categories: one-dimensional (1D) chaotic maps and high-dimensional (HD) chaotic maps. 1D chaotic maps usually contain one variable and a few parameters. Examples include the Logistic, Gaussian, Sine and Tent maps [12]. Their structures and chaotic orbits are rather simple. With the development of chaotic signal estimation technologies, when little information is extracted, their chaotic orbits may be estimated [2,17] and their parameters or/and initial values may be predicted [28]. These weaknesses limit their applications in many security areas. For example, when 1D chaotic maps are used in image encryption, several encryption algorithms were reported to be insecure [2,20]. A Logistic map-based image encryption algorithm proposed in [23] was proved to be insecure [1,15]. On the other hand, HD chaotic maps have at least two variables, e.g., the Hénon map [12], Lorenz system [26] and Chee-Lee system [4]. Compared with 1D chaotic maps, HD chaotic maps usually have more complex structures and better chaotic performance. These make their chaotic orbits much harder to predict. However, their hardware implementations are relatively complex and expensive. Therefore, developing a chaotic map with excellent chaotic performance and a low implementation cost becomes significative.

* Corresponding author. Tel.: +853 88228458; fax: +853 88222426.
E-mail address: yicongzhou@umac.mo (Y. Zhou).

http://dx.doi.org/10.1016/j.ins.2014.11.018
0020-0255/

This paper proposes a two-dimensional (2D) chaotic map, called the 2D Sine Logistic modulation map (2D-SLMM). It is derived from two 1D chaotic maps, the Sine and Logistic maps. Performance analysis is provided to show that 2D-SLMM has the wider chaotic range, better ergodicity and hyperchaotic properties than existing chaotic maps. To show its performance in security applications, a chaotic magic transform (CMT) is introduced for image encryption. Using chaotic sequences generated by 2D-SLMM, CMT is able to quickly shuffle image pixels in both the row and column directions at the same time. Integrating 2D-SLMM and CMT, we also propose a new CMT-based image encryption algorithm (CMT-IEA). Experimental results and security analysis are given to show that the proposed CMT-IEA can encrypt different types of digital images with a high level of security and low time complexity.

The rest of this paper is organized as follows. Section 2 will introduce 2D-SLMM and analyze its chaotic behaviors. In Section 3, CMT for image encryption will be introduced. Section 4 will propose CMT-IEA using 2D-SLMM and CMT. Section 5 will provide simulation results and time complexity analysis. Section 6 will analyze the security issues of CMT-IEA and Section 7 will reach a conclusion.

2. 2D Sine Logistic modulation map

This section introduces the 2D Sine Logistic modulation map (2D-SLMM) and investigates its chaotic behaviors.

2.1. Mathematic definition

The Logistic and Sine maps are two commonly used 1D chaotic maps. They are defined by Eqs. (1) and (2), respectively.

$$x_{i+1} = a x_i (1 - x_i) \tag{1}$$

$$x_{i+1} = u \sin(\pi x_i) \tag{2}$$

where a and u are parameters. $a \in [0, 4]$ and $u \in [0, 1]$.

Both maps are nonlinear transforms to generate next iteration values with a simple structure. Thus, their orbits are easy to be predicted using chaotic signal estimation technologies [2,17]. To address this problem, we propose 2D-SLMM. It is defined by Eq. (3).

$$\begin{cases} x_{i+1} = \alpha(\sin(\pi y_i) + \beta) x_i (1 - x_i) \\ y_{i+1} = \alpha(\sin(\pi x_{i+1}) + \beta) y_i (1 - y_i) \end{cases} \tag{3}$$

where α and β are control parameters. $\alpha \in [0, 1]$ and $\beta \in [0, 3]$.

2D-SLMM is derived from the Logistic and Sine maps. A combination of the Sine map and parameter β is used to modulate the output of the Logistic map to enhance its nonlinearity and randomness. The result is then extended from one-dimension to two-dimension to obtain 2D-SLMM. Two output values of 2D-SLMM x_{i+1} and y_{i+1} intertwine each other. Thus, its orbits and iteration values are difficult to predict. When parameter β is close to 3, 2D-SLMM shows good chaotic performance. For simplicity, we set $\beta = 3$ in the rest of this paper.

2.2. Performance evaluation

The proposed 2D-SLMM has good chaotic behaviors. Here, we use the trajectory, Lyapunov exponent [18,19], Lyapunov dimension [14] and Kolmogorov entropy [7] to evaluate its chaotic performance.

2.2.1. Trajectory

Fig. 1 compares the trajectories of 2D-SLMM with the 2D Logistic map defined in Eq. (4) [31]. Their parameters are set to the values that ensure both maps to have excellent chaotic behaviors. Their initial values are set to the same. As can be seen, the trajectory of 2D-SLMM distributes in much larger regions in the phase plane than that of the 2D Logistic map. This means that 2D-SLMM is able to generate more random outputs and has much better ergodicity property.

$$\begin{cases} x_{i+1} = r(3y_i + 1) x_i (1 - x_i) \\ y_{i+1} = r(3x_{i+1} + 1) y_i (1 - y_i) \end{cases} \tag{4}$$

2.2.2. Lyapunov exponent and Lyapunov dimension

Chaotic behaviors of a dynamical system can be evaluated by the Lyapunov exponent (LE) and Lyapunov dimension (LD). For two extremely close trajectories of a dynamical system in the phase plane, a positive LE value denotes that they exponentially separate in each unit time and will be totally different over the time. A HD chaotic map has at least two LE values and the maximum LE (MLE) value determines its predictability. It has chaotic behaviors when its MLE value is positive, and has hyperchaotic behaviors when it has more than one positive LE values. A HD chaotic map with hyperchaotic behaviors generally has high complexity and its trajectories are extremely difficult to predict. On the other hand, LD is a measure of the geometric scaling properties of a dynamical system. It is a basic property to reflect a dynamical system's complexity [14].

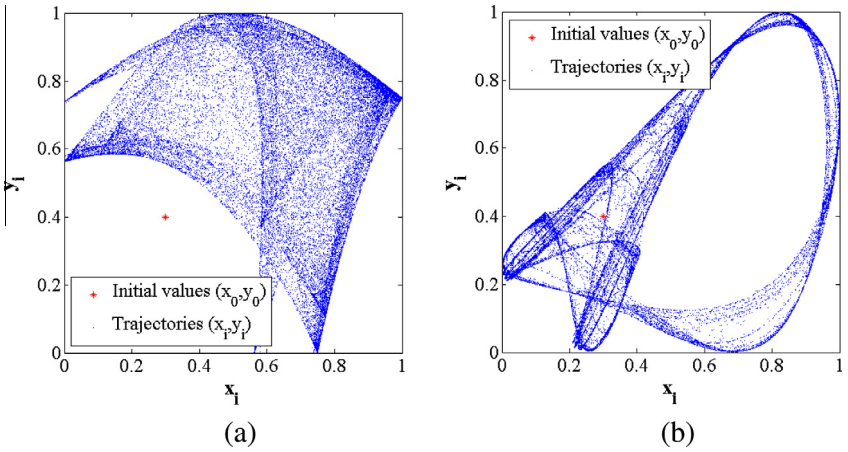

Fig. 1. Trajectories of (a) the proposed 2D-SLMM with parameters $\alpha = 1, \beta = 3$ and (b) the 2D Logistic map with parameter $r = 1.19$.

In our experiments, we use the algorithms in [6,27] to calculate the LE and LD values. A 2D chaotic map usually has two LE values. Fig. 2(a) and (b) plot two LE values, λ_1 and λ_2, of 2D-SLMM and the 2D Logistic map along with their parameter settings. Fig. 2(c) plots their LD distributions where we shift parameter r of the 2D Logistic map to provide a better visual comparison. As can be seen from Fig. 2(a), when $\alpha \in [0.87, 1]$ (approximately), 2D-SLMM has chaotic behaviors because its MLE value λ_1 is positive, and when $\alpha \in [0.905, 1]$ (approximately), 2D-SLMM has hyperchaotic behaviors because its two LE values λ_1 and λ_2 are positive. Both values become bigger when α is close to 1. From Fig. 2(b), we can observe that the 2D Logistic map has chaotic behaviors when its parameter $r \in [1.11, 1.15] \bigcup [1.18, 1.19]$ (approximately), and it does not have hyperchaotic behaviors. Comparing the LD values in Fig. 2(c), we can see that the LD values of 2D-SLMM are bigger than these of the 2D Logistic map. Therefore, 2D-SLMM has the wider chaotic range and more complex trajectories than the 2D Logistic map. Its outputs are difficult to predict because of its hyperchaotic property.

2.2.3. Kolmogorov entropy

Kolmogorov entropy (KE) is a test to calculate how much extra information needs to forecast the trajectory of a dynamical system in each unit time. A dynamical system with a positive KE value will be chaotic and a bigger positive KE value means better unpredictability.

In our experiments, we use the method reported in [11] to calculate the KE values of four testing chaotic maps: 2D-SLMM, the Logistic, Sine and 2D Logistic maps. 12,000 continuous points are selected from the trajectory of each chaotic map to calculate their KE values. Fig. 3 plots their KE values along with their parameters α, a, u and r. To provide a better visual comparison, we linearly scale down parameter a of the Logistic map and shift parameter r of the 2D Logistic map into the range of $[0, 1]$. As can be seen in Fig. 3, 2D-SLMM has larger positive KE values than other chaotic maps in most of the parameter settings. This means that the trajectory of 2D-SLMM has better unpredictability. For the chaos-based cryptographic systems, their security performance is determined by the characteristics of chaotic maps. Thus, 2D-SLMM is more suitable for cryptographic systems than the Logistic, Sine and 2D Logistic maps.

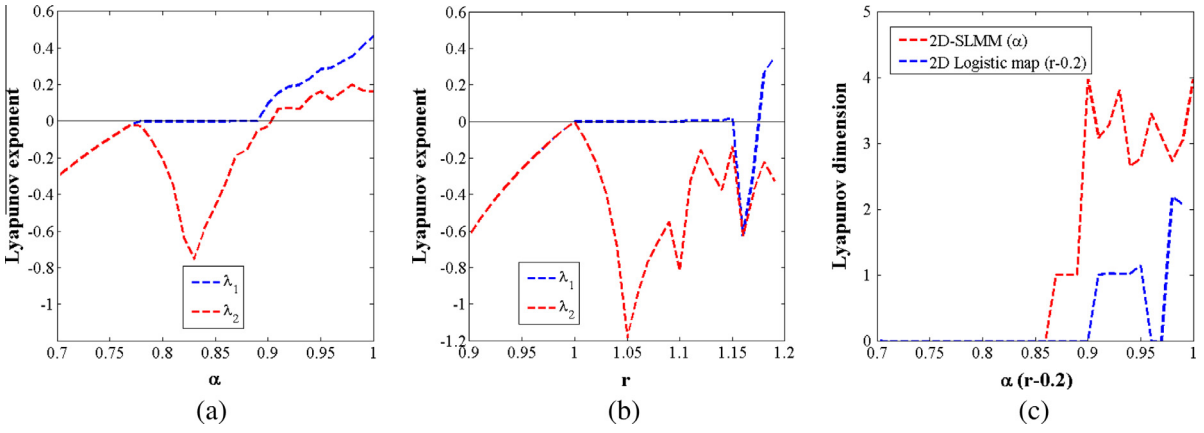

Fig. 2. The LE distributions of (a) 2D-SLMM ($\beta = 3$), and (b) the 2D Logistic map; (c) the LD distributions of 2D-SLMM ($\beta = 3$) and the 2D Logistic map.

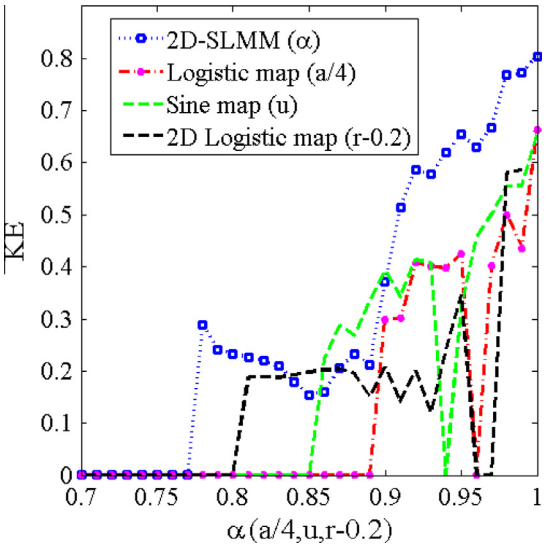

Fig. 3. The KE values of 2D-SLMM, the Logistic, Sine and 2D Logistic maps.

In summary, the iteration values (x_i, y_i) of 2D-SLMM fall into most regions of its 2D phase plane. The evaluation and comparison results of LE, LD and KE show that 2D-SLMM has more complex chaotic behaviors and better unpredictability than existing chaotic maps.

3. Chaotic magic transform

3.1. Definition

Digital images usually have high information redundancy because their neighboring pixels have high correlations. To break these correlations, this section proposes a new chaotic magic transform (CMT) to randomly change image pixel positions.

First, we generate a shuffled index matrix I. Let S be a 2D chaotic matrix with a size of $M \times N$ generated by 2D-SLMM, $\mathbb{S}$ be a 2D matrix generated from S by sorting its data within each column. The shuffled index matrix I is defined by,

$$I(i,j) = k \quad \text{for } \mathbb{S}(i,j) = S(k,j) \tag{5}$$

where i, j, k are integers, $1 \leqslant i, k \leqslant M$ and $1 \leqslant j \leqslant N$

Let P be an original image with a size of $M \times N$ and T be the shuffled image. The chaotic magic transform (CMT) is defined by,

$$T = \mathcal{F}(P, I) \tag{6}$$

where $\mathcal{F}$ is the CMT function as described in Algorithm 1. CMT is able to change pixel positions within the original image P according to the chaotic matrix generated by 2D-SLMM. It randomly connects pixels in different rows and columns into circles, and then shift them within the circles.

Algorithm 1. The chaotic magic transform.

Input: the original image P and chaotic matrix S with the size of $M \times N$
1: Sort each column of S to obtain the sorted matrix $\mathbb{S}$;
2: Generate shuffled index matrix I using Eq. (5);
3: **for** $i = 1$ to M **do**
4: Connect pixels in P with locations $(I_{i,1}, 1), (I_{i,2}, 2), (I_{i,3}, 3), \ldots, (I_{i,N}, N)$ into a circle;
5: Shift these pixels i positions to the left;
6: **end for**
Output: The shuffled image T.

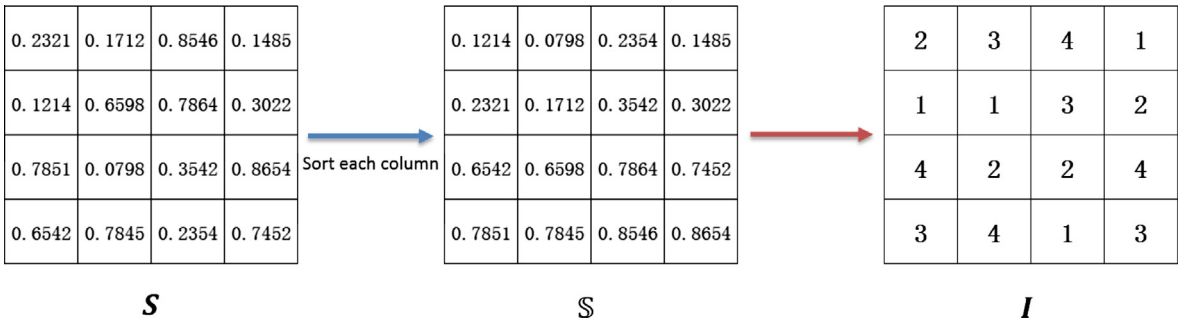

Fig. 4. An example of the generation of index matrix *I*.

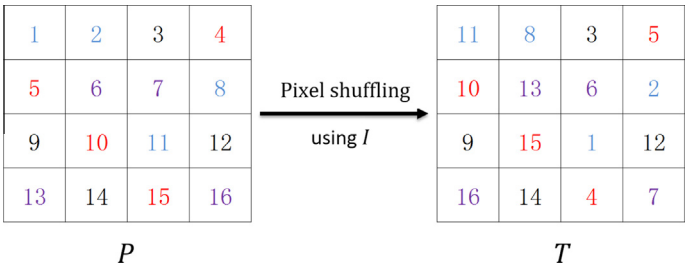

Fig. 5. An example of the pixel shuffling processes in CMT.

Figs. 4 and 5 show an illustrative example of CMT. Fig. 4 shows the generation of the index matrix I from a chaotic matrix while Fig. 5 shows the pixel shuffling processes according to the index matrix I. As can be seen in Fig. 4, by sorting each column of chaotic matrix S with an ascending order, the sorted matrix $\mathbb{S}$ can be obtained. The index matrix I indicates the locations where the data in $\mathbb{S}$ are permutated from. For example, $I_{1,1} = 2$ means that $\mathbb{S}_{1,1}$ is permuted from the location $(2, 1)$ in S. In Fig. 5, P is the original image matrix and T is the shuffled result of CMT. Because the index matrix I has four rows, the shuffling processes can be divided into four steps as follows.

Step 1: For $(2, 3, 4, 1)$ in the first row in I, we connect the pixels in red $(P_{2,1}, P_{3,2}, P_{4,3}, P_{1,4})$ in matrix P, namely $(5, 10, 15, 4)$, into a circle and shift them 1 pixel position to the left. After shifting, in the CMT result T, $T_{2,1} = P_{3,2}, T_{3,2} = P_{4,3}, T_{4,3} = P_{1,4}$ and $T_{1,4} = P_{2,1}$, also see the red numbers in matrix T.

Step 2: For $(1, 1, 3, 2)$ in the second row of I, we connect the pixels in blue $(P_{1,1}, P_{1,2}, P_{3,3}, P_{2,4})$ in P, namely $(1, 2, 11, 8)$, into a circle and shift them 2 pixel positions to the left. Then, $T_{1,1} = P_{3,3}, T_{1,2} = P_{2,4}, T_{3,3} = P_{1,1}$ and $T_{2,4} = P_{1,2}$, see the blue numbers in T.

Step 3: For $(4, 2, 2, 4)$ in the third row of I, we connect the pixels in purple $(P_{4,1}, P_{2,2}, P_{2,3}, P_{4,4})$ in P, namely $(13, 6, 7, 16)$, into a circle and shift them 3 pixel positions to the left. Thus, $T_{4,1} = P_{4,4}, T_{2,2} = P_{4,1}, T_{2,3} = P_{2,2}, T_{4,4} = P_{2,3}$, see the purple numbers in T.

Step 4: For $(3, 4, 1, 3)$ in the last row of I, we connect the pixels in black $(P_{3,1}, P_{4,2}, P_{1,3}, P_{3,4})$ in P, namely $(9, 14, 3, 12)$, into a circle and shift them 4 pixel positions to the left. The shifting results are $T_{3,1} = P_{3,1}, T_{4,2} = P_{4,2}, T_{1,3} = P_{1,3}$ and $T_{3,4} = P_{3,4}$, see the black numbers in T.

After all shifting operations, we can obtain the CMT result T.

3.2. Discussion

Different from many image scrambling algorithms that perform pixel shuffling row by row and column by column, CMT changes pixel positions simultaneously in both the horizontal and vertical directions based on the pseudo-random numbers generated by the chaotic map. After applying CMT to an image only one time, a pixel may be permuted to any position within an image. By this way, CMT can achieve the following advantages:

(1) A pixel can be quickly separated with its neighboring pixels.
(2) Without knowing the detailed conditions of the chaotic map, the CMT results are extremely difficult to predict, achieving a high level of security.

4. New CMT-based image encryption algorithm

Existing image encryption methods can be classified into two categories. The first category treats a digital image as a bit stream and applies data encryption algorithms for image encryption. These data encryption algorithms include the well-known Digital Encryption Standard (DES) [9], Advanced Encryption Standard (AES) [8], Block cipher [5] and many others. Because most of data encryption algorithms are stream/block ciphers and a digital image usually has a nature of high information redundancy, using stream/block ciphers to encrypt a digital image may not reach a good encryption performance [31]. The other category performs image encryption considering the special properties of digital images [21,37]. The algorithms include chaos-based image encryption algorithms [10,35,36], wavelet transform-based image encryption algorithms [3,22], recursive-sequence-based image encryption algorithms [38,39], and so on.

Among these image encryption technologies, chaos-based image encryption algorithms show excellent encryption performance. Using the proposed 2D-SLMM and CMT, this section proposes a new CMT-based image encryption algorithm, called CMT-IEA. It uses 2D-SLMM to generate pseudo-random sequences for CMT to change image pixel positions and for pixel substitution to change image pixel values.

Algorithm 2. The proposed CMT-IEA.

Input: The security key $K = (x_0, y_0, \alpha, H, G_1, G_2)$ and the plaintext image P with the size of $M \times N$.

1: Transform the binary sequences x_0, y_0, α, H into decimal numbers using Eq. (7) and G_1, G_2 into integers;
2: Obtain two groups of initial conditions $(x_{01}, y_{01}, \alpha_1)$ and $(x_{02}, y_{02}, \alpha_2)$ using Eq. (8);
3: Generate two chaotic matrices S_1 and S_2 (with the size of $M \times N$) using 2D-SLMM with two groups of initial conditions in Step 2;
4: **for** $i = 1$ to 2 **do**
5: Apply CMT to the plaintext image P using the chaotic matrix S_i;
6: Do row substitution to the image using chaotic matrix S_i;
7: Do column substitution to the image using chaotic matrix S_i;
8: **end for**

Output: The encrypted image C.

The flowchart of CMT-IEA is shown in Fig. 6. The plaintext image P is the original image and the ciphertext image C is the encrypted image. The security key is used to produce initial values and parameters of 2D-SLMM. CMT is to achieve the confusion property by randomly shuffling all pixel positions. The pixel substitution operations are to achieve the diffusion property by randomly changing all pixel values. To obtain random-like encryption results while avoiding the cases that 2D-SLMM

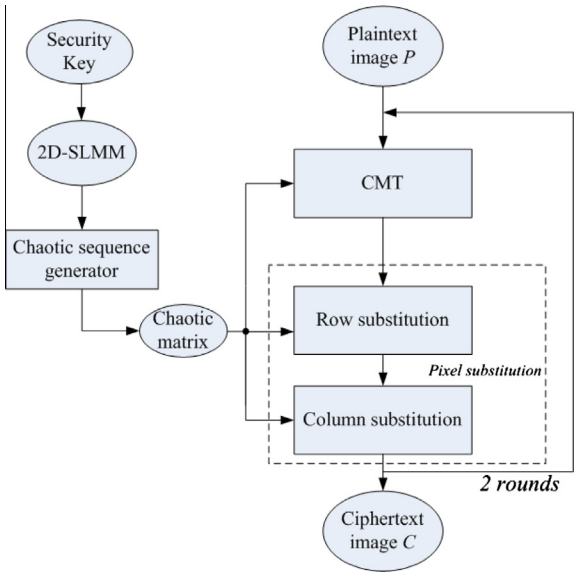

Fig. 6. The flowchart of the proposed CMT-IEA.

may lose its chaotic behaviors in some parameter settings, the proposed CMT-IEA uses two rounds of CMT and pixel substitution operations. The decryption process simply reverses the encryption operations of CMT-IEA. A pseudo-code implementation of CMT-IEA is shown in Algorithm 2. The encryption process is represented by $C = Enc(P, K)$ while the decryption process is denoted as $D = Dec(C, K)$, where K is the security key.

Here, we show an example of CMT-IEA described in Algorithm 2. In this example, $K = (0.9380, 0.7006, 0.4902, 0.1318, 1144200, 12173572)$, then two groups of initial conditions can be generated as $(0.4980, 0.2606, 0.9502)$ and $(0.7276, 0.4902, 0.9798)$. 2D-SLMM then generates two chaotic matrices as follows,

$$S_1 = \begin{pmatrix} 1.4995 & 1.2021 & 0.8105 & 1.2058 \\ 1.2624 & 1.6507 & 1.4355 & 1.5654 \\ 1.1080 & 1.0024 & 1.1424 & 1.0177 \\ 1.5315 & 1.6962 & 1.4705 & 1.5193 \end{pmatrix}, \quad S_2 = \begin{pmatrix} 1.6693 & 1.4124 & 0.8790 & 1.5755 \\ 0.9398 & 1.3284 & 1.5478 & 0.9657 \\ 1.6721 & 1.5142 & 1.2438 & 1.4072 \\ 0.8877 & 0.8262 & 1.2074 & 1.5392 \end{pmatrix}$$

The results in each operation are shown in Fig. 7. As can be seen, P is the original image and after two rounds of CMT and substitution with chaotic matrices S_1 and S_2, the encrypted result C can be obtained.

4.1. Generation of initial conditions

The security key of CMT-IEA is a sequence with a length of 256 bits. Its structure is shown in Fig. 8. It contains information of initial values and parameters of 2D-SLMM and can be divided into 6 parts, x_0, y_0, α, H, G_1, and G_2. (x_0, y_0) are initial values and α is control parameter. H, G_1 and G_2 are designed to change the initial values and parameters to enlarge the security key space.

x_0, y_0, α and H are decimal numbers which are generated by a 52-bit string $\{b_1, b_2, \ldots, b_{52}\}$ using the IEEE 754 format [25], as shown in Eq. (7).

$$x = \frac{\sum_{i=1}^{52} b_i 2^{52-i}}{2^{52}} \tag{7}$$

G_1 and G_2 are two integer coefficients generated by a 24-bit string $\{b_1, b_2, \ldots, b_{24}\}$. The initial values and control parameters of 2D-SLMM for generating two chaotic matrices are defined by Eq. (8)

$$\begin{cases} x_{0i} = (x_0 + G_i H) \bmod 1 \\ y_{0i} = (y_0 + G_i H) \bmod 1 \\ \alpha_i = 0.9 + ((\alpha + G_i H) \bmod 0.1) \end{cases} \tag{8}$$

where the round number i is 1 or 2.

In Eq. (8), the generated initial values will fall into the range of $[0, 1]$ and the control parameter α will be limited within $[0.9, 1]$. Thus, 2D-SLMM has good chaotic performance under these settings.

In the proposed CMT-IEA, users have the flexibility of manually selecting a binary sequence with 256 bits or randomly generating a binary stream to produce the security key. In our simulations and comparisons, we randomly generate binary

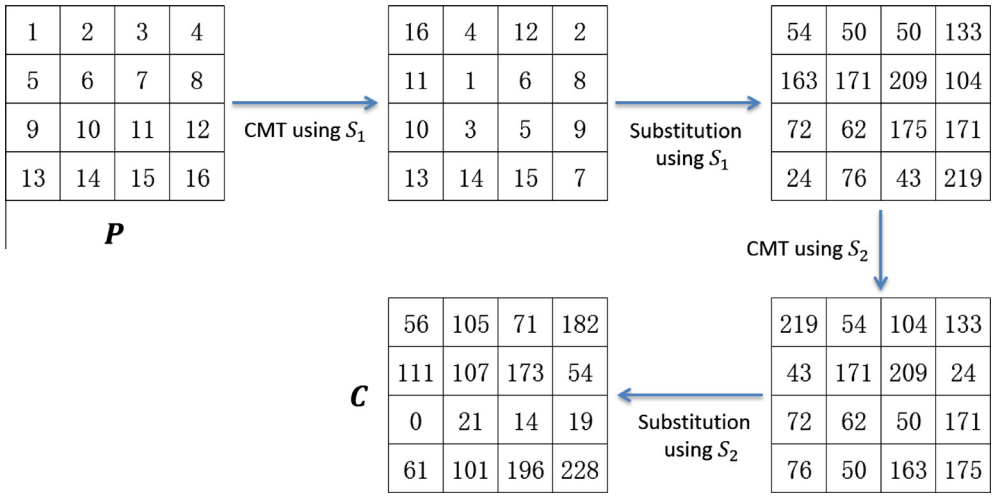

Fig. 7. An example of the proposed CMT-IEA.

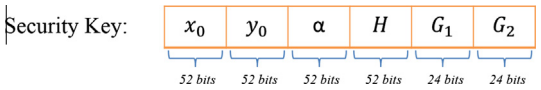

Fig. 8. The security key structure.

streams with a length of 256 bits as the security keys that will be returned along with the encrypted results for image decryption.

4.2. Pixel substitution

An encryption algorithm with a good diffusion property has the ability of resisting the chosen-plaintext attack. The diffusion property indicates that encrypting two plaintexts with a tiny difference using the same security key will yield totally different encryption results. To achieve the diffusion property, CMT-IEA performs pixel substitution in two steps: the row substitution and column substitution.

Suppose the CMT result T and chaotic matrix S are both with a size of $M \times N$. The mathematic definition of the pixel substitution is described in Eq. (9).

$$C_i = \begin{cases} (T_i + T_R + \lfloor S_i \times 2^{32} \rfloor) \bmod F & \text{If } i = 1 \\ (T_i + C_{i-1} + \lfloor S_i \times 2^{32} \rfloor) \bmod F & \text{If } i \in [2, R] \end{cases} \tag{9}$$

where F is the number of allowed pixel values in the plaintext image. For example, $F = 256$ if the image pixels are represented by 8-bit decimals. And $\lfloor \cdot \rfloor$ is the floor operation. When performing the row substitution, $R = N$ and Eq. (9) is used for each row of T; When doing the column substitution, $R = M$ and Eq. (9) is used for each column of T.

Fig. 9 shows the results of directly applying pixel substitution only one time to a data matrix and to an image. After one-time pixel substitution, the data matrix is dramatically changed into another randomly distributed data matrix (see Fig. 9(a)) and the image becomes a random-like image (see Fig. 9(b)). From Fig. 9(c), we can observe the distribution of the encrypted image in Fig. 9(b), the numbers of pixels in each grayscale level are almost equal. Attackers have difficult to obtain information of the original image using any statistic analysis method.

The proposed CMT-IEA can also be used for data encryption. The data sequence can be rearranged into 2D data matrices with a fixed size of blocks, e.g. 512×512 blocks. CMT-IEA is then used to encrypt the data block by block.

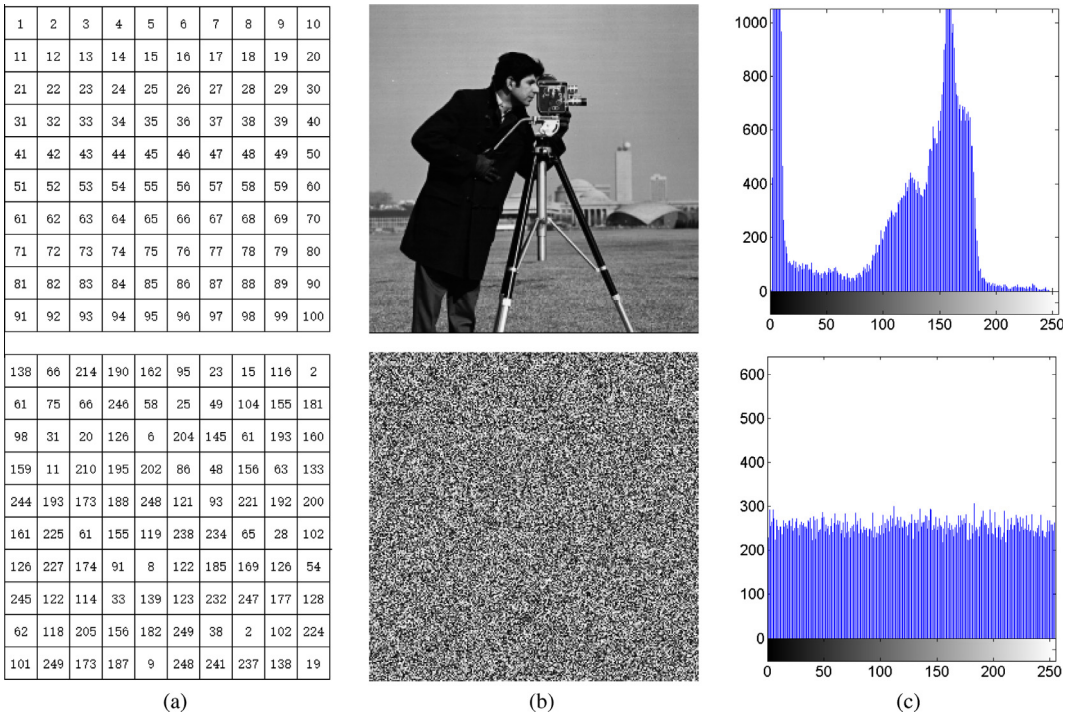

Fig. 9. Pixel substitution. The first row shows the original data matrix and image while the second row shows their corresponding one-time pixel substitution results. (a) A 10×10 data matrix and its substitution result; (b) the original and substituted images; and (c) the histograms of images in (b).

5. Simulation results and time complexity analysis

For a good image encryption algorithm, it should have the ability of transforming different types of plaintext images into random-like ciphertext images. This section provides several simulation results of CMT-IEA and analyzes its time complexity.

5.1. Simulation results

Fig. 10 shows the simulation results of using CMT-IEA to encrypt different types of images. The all-zero and all-one images are two types of extreme cases of binary images where all pixels in images have same values. Their simulation results demonstrate the excellent encryption performance of the proposed CMT-IEA because it is able to transform all-pixel-the-same images into noise-like encrypted images in which all pixel values are randomly and uniformly distributed. As can be seen in Fig. 10, the histograms of all plaintext images have specific patterns. But all ciphertext images are random-like images and their histograms distribute uniformly. They do not contain any information of the plaintext images. This shows that CMT-IEA has the good encryption performance for different types of images.

5.2. Time complexity analysis

CMT-IEA has a high encryption speed. By using the images from the USC-SIPI 'Miscellaneous' dataset under the MATLAB implementation, the CMT-IEA's encryption/decryption time for an image with a size of 256×256 is 0.05312 ± 0.010043 s. Therefore, the CMT-IEA's encryption/decryption speed is about 9.413 Mb/s (Megabits per second). Table 1 compares the encryption speeds of several image encryption algorithms for different sizes of images. The proposed CMT-IEA has the fastest encryption speed. Thus it has the lowest time complexity.

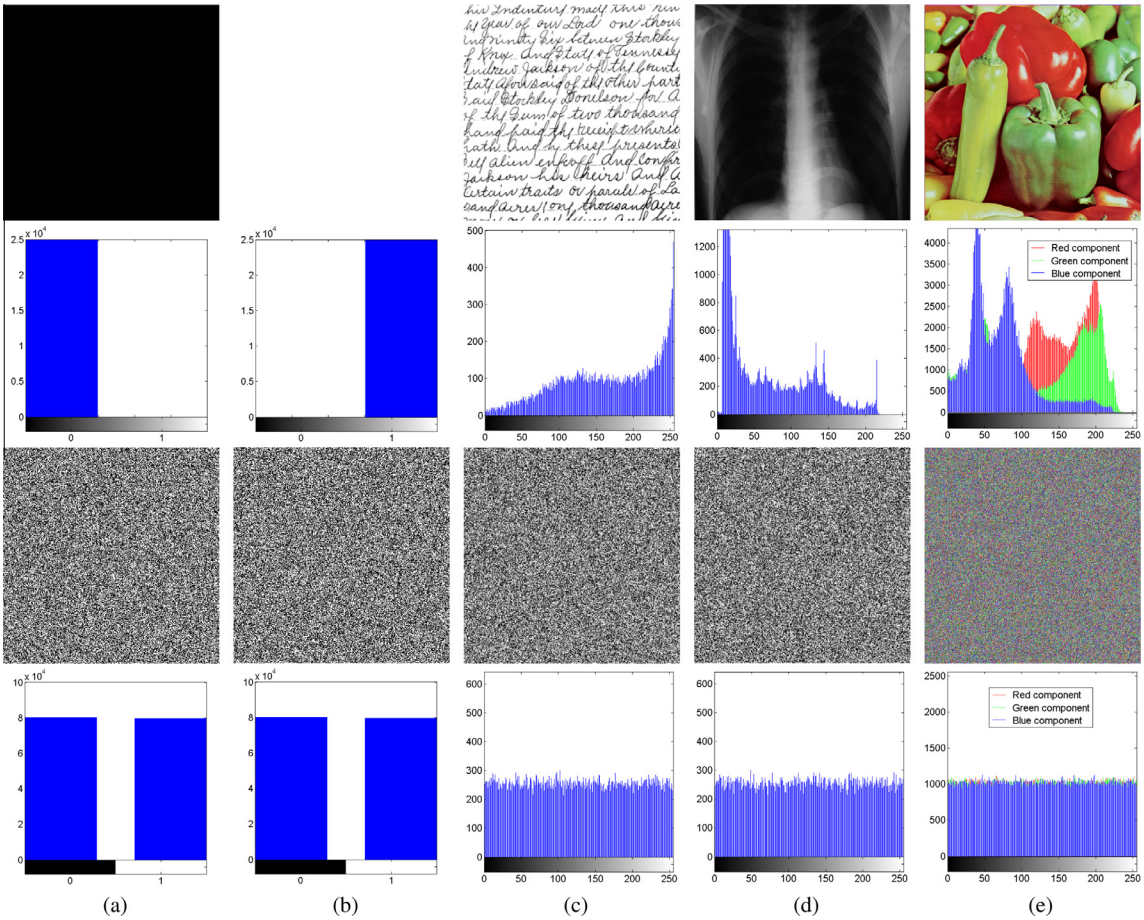

Fig. 10. Simulation results of different kinds of images. The first and second rows show plaintext images and their histograms. The third and fourth rows show the ciphertext images and their histograms. (a) All-zero image; (b) all-one image; (c) text image; (d) medical image; and (e) color image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Encryption speeds (second) of several image encryption algorithms for different sizes of images.

Image size	64 × 64	128 × 128	256 × 256	512 × 512	1024 × 1024
Wu's [31]	0.2503	1.3412	5.6544	27.1702	109.9320
Zhou's [36]	0.0174	0.0549	0.1967	0.6547	3.2415
Wu's [30]	0.0161	0.0582	0.2368	0.8587	3.5037
Liao's [16]	0.0546	0.1415	0.5630	2.2597	9.0046
CMT-IEA	0.0042	0.0130	0.0538	0.2338	1.1494

6. Security analysis

To demonstrate the security performance of CMT-IEA, different kinds of security analysis methods are being utilized in this section. Most of test images are selected from the USC-SIPI 'Miscellaneous' image dataset.

6.1. Security key analysis

A good encryption algorithm should have a sufficiently large security key space and be extremely sensitive to its security key changes. The proposed CMT-IEA has a security key with a length of 256 bits and thus its key space is 2^{256}. This is large enough to resist the brute-force attack according to the computation ability of current computers.

The key sensitivity can be described in two sides,

(1) The security key should be sensitive in the encryption process. Using two encryption keys with a tiny difference to encrypt a plaintext image, the ciphertext images are totally different.
(2) The security key should be sensitive in the decryption process. Using two decryption keys with a tiny difference to recover a ciphertext image, the recovered images are totally different.

Fig. 11 shows the key sensitivity analysis results. K_2 and K_3 are two different keys derived from K_1 with one bit difference. When a plaintext image (Fig. 11(a)) is encrypted using K_1 and K_2, the encryption results in Fig. 11(b) and (c) are totally different. Their difference is shown in Fig. 11(d). A ciphertext image (Fig. 11(b)) can be only completely decrypted by the correct key as shown in Fig. 11(e). When the ciphertext image is decrypted by two keys with one bit difference from K_1, the decryption results in Fig. 11(f) and (g) are also totally different and unrecognizable. Fig. 11(h) shows their difference. Therefore, the proposed CMT-IEA is sensitive with its security keys in both the encryption and decryption processes.

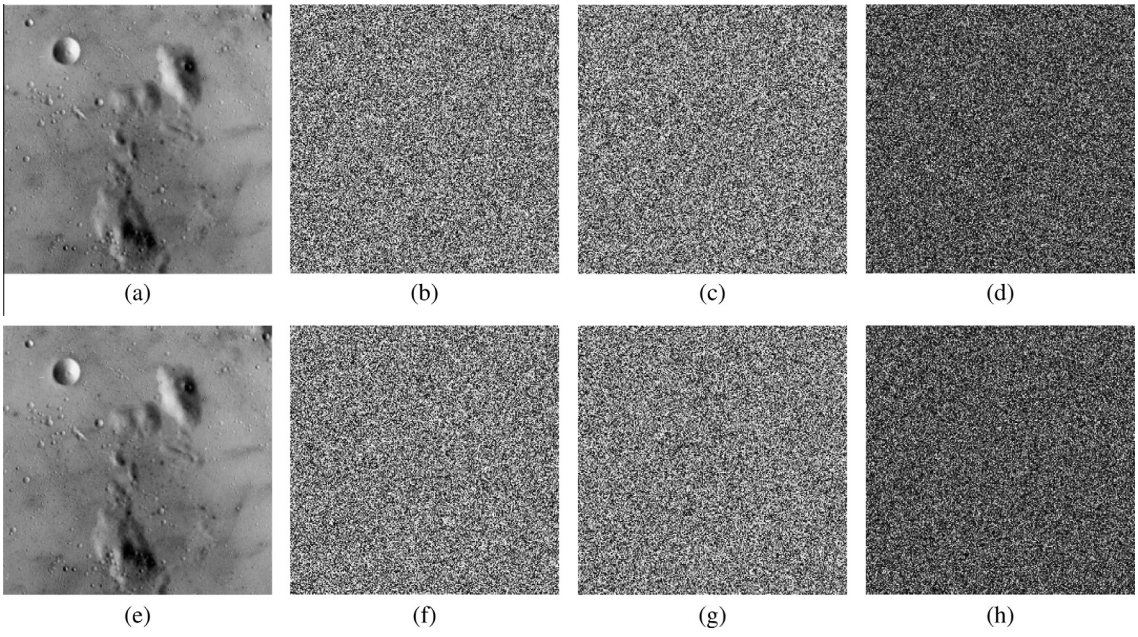

Fig. 11. Key sensitivity analysis. (a) Plaintext image P; (b) ciphertext image $C_1 = Enc(P, K_1)$; (c) ciphertext image $C_2 = Enc(P, K_2)$; (d) difference of ciphertext images: $|C_1 - C_2|$; (e) decrypted image $D_1 = Dec(C_1, K_1)$; (f) decrypted image $D_2 = Dec(C_1, K_2)$; (g) decrypted image $D_3 = Dec(C_1, K_3)$; and (h) difference of decrypted images: $|D_2 - D_3|$.

6.2. Pixel correlation

An image generally has high data redundancy and thus its pixels have high correlations with their neighboring pixels. A good image encryption algorithm should have the ability of breaking these correlations. Mathematically, data correlation is defined by Eq. (10)

$$Corr = \frac{E[(X - \mu_X)(Y - \mu_Y]}{\sigma_X \sigma_Y} \tag{10}$$

where X and Y are two data sequences. μ is the mean value and σ is the standard deviation. If two sequences X and Y have high correlations, their correlation value is close to 1. Otherwise, it is close to 0.

In our test, we randomly select 2,000 pixels and their corresponding adjacent pixels along with the horizontal, vertical and diagonal directions. Fig. 12 plots the distributions of the pixel sequence pairs, X and Y of the plaintext image and its ciphertext image generated by the proposed CMT-IEA. As can be seen, for the plaintext image, most points are located on or nearby the diagonal line of the coordinate system, while for the ciphtertext image, all points randomly distribute in the entire data range. This means that adjacent pixels in the plaintext image have equal values or close to each other while adjacent pixels in the ciphertext image dramatically change. Table 2 shows the quantitative results of the adjacent pixel correlation test. The results of the plaintext image are close to 1 while the ciphertext image's results are close to 0. These further verify that the encrypted image by CMT-IEA has extremely low correlation.

6.3. Local Shannon entropy

Different from the histogram analysis that gives a straightforward result to show how uniformly pixels of an image distribute, the local Shannon entropy (LSE) [32] can provide a qualitative standard to evaluate the randomness of an image.

Firstly, randomly choosing k non-overlapping image blocks with T_B pixels in the test image, and the test score can be gotten by calculating the mean Shannon entropy values of these image blocks by Eq. (11)

$$\overline{H_{k,T_B}}(S) = \sum_{i=1}^{k} \frac{H(S_i)}{k} \tag{11}$$

where $S_1, S_2, \ldots, S_k$ are k chosen image blocks. The image will pass the LSE test if $\overline{H_{k,T_B}}(S)$ falls into the interval of $(h^*_{left}, h^*_{right})$ defined by Eq. (12)

$$\begin{cases} h^*_{left} = \mu_{H(x)} - \Phi(\alpha/2)^{-1}\sigma_{H(x)}/\sqrt{k} \\ h^*_{right} = \mu_{H(x)} + \Phi(\alpha/2)^{-1}\sigma_{H(x)}/\sqrt{k} \end{cases} \tag{12}$$

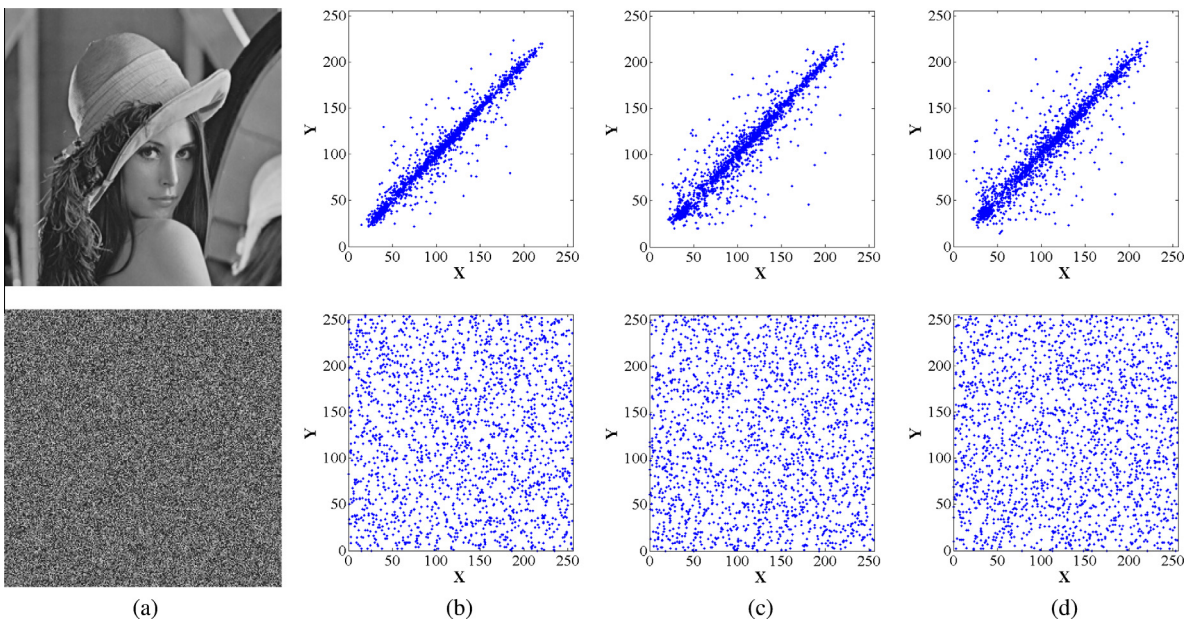

Fig. 12. Distributions of adjacent pixel sequence pairs of (a) the plaintext image and its ciphertext image along with the (b) horizontal, (c) vertical, and (d) diagonal directions, respectively.

Table 2
Pixel correlations of the Lena image and its encrypted image.

Images	Horizontal	Vertical	Diagonal
Original image	0.965935	0.936620	0.915342
Encrypted result	0.002383	−0.008576	0.040242

where $\Phi^{-1}(\cdot)$ is the inverse cumulative distribution function (CDF) of the standard Normal distribution $\mathbb{N}(0,1)$. $\mu_{H(x)}$ and $\sigma_{H(x)}$ are the mean and standard deviation of the LSE values of k non-overlapping image blocks in an ideally random image. The Shannon entropy $H(x)$ is defined by Eq. (13)

$$H(X) = -\sum_{i=1}^{n} Pr(x_i)\log_2 Pr(x_i) \tag{13}$$

where X represents a collection of pixels, x_i is the i-th possible value in X and $Pr(x_i)$ is the probability of x_i.

In each test, test images are encrypted by different image encryption algorithms, and then measured by LSE. Several tests are done and the best test results are reported as the experiment results. Table 3 shows the LSE results of the Wu's algorithm [31], Zhou's algorithm [35], Wu's algorithm [30], Liao's algorithm [16] and the proposed CMT-IEA. As can be seen, 26 out of 28 images encrypted by CMT-IEA pass the LSE test. Its pass rate is much higher than those of other four encryption algorithms. This means that the ciphertext images of the proposed CMT-IEA have good randomness.

6.4. Differential attack

The differential attack, also called the chosen-plaintext attack, is a well-known and effective method to break an encrypted result. An encryption algorithm with a good diffusion property can resist the differential attack. The number of pixel change rate (NPCR) and unified averaged changed intensity (UACI) can be used to evaluate the diffusion property of an image encryption algorithm. Mathematically, the NPCR and UACI between two images C_1 and C_2 are defined as Eqs. (14) and (15) [4].

Table 3
The Local Shannon entropy test of the ciphertext images generated by different image encryption algorithms. $\alpha = 0.001, k = 30, T_B = 1936$. (The results marked in bold have passed the test).

File name	Wu's [31]	Zhou's [35]	Wu's [30]	Liao's [16]	CMT-IEA
5.1.09	**7.901985**	**7.903354**	7.903764	7.904191	**7.902127**
5.1.10	**7.902731**	**7.902443**	**7.901801**	**7.902371**	**7.903402**
5.1.11	**7.902446**	**7.902756**	**7.903306**	7.900799	**7.902687**
5.1.12	**7.902556**	**7.901526**	7.904478	**7.903374**	**7.901906**
5.1.13	**7.902688**	7.904563	7.904657	7.904566	**7.902825**
5.1.14	7.903474	**7.902954**	**7.902874**	**7.903111**	**7.902340**
5.2.08	7.903953	**7.902356**	**7.903218**	**7.901762**	**7.903327**
5.2.09	**7.902233**	7.899853	**7.903089**	7.905854	**7.901765**
5.2.10	7.900714	**7.902654**	**7.902077**	**7.902768**	**7.902748**
5.3.01	**7.902727**	**7.902647**	**7.902108**	7.901040	**7.901772**
5.3.02	**7.903182**	7.910474	7.904169	7.900981	**7.903328**
7.1.01	**7.902173**	**7.902634**	**7.901965**	**7.902145**	7.901305
7.1.02	7.900879	**7.901634**	7.904970	**7.902157**	**7.901578**
7.1.03	**7.902543**	7.905423	7.891503	7.900645	**7.903099**
7.1.04	7.901126	**7.902125**	**7.903399**	7.904141	**7.902607**
7.1.05	7.903579	7.883653	7.901301	7.900027	7.905305
7.1.06	**7.901930**	**7.902356**	**7.903367**	**7.901736**	**7.902695**
7.1.07	**7.903000**	**7.902364**	7.899556	7.900802	**7.902896**
7.1.08	**7.903197**	7.904456	7.883531	7.900944	**7.901632**
7.1.09	**7.902308**	**7.903012**	**7.903201**	7.905658	**7.903173**
7.1.10	7.899542	**7.901598**	**7.901542**	7.893848	**7.901524**
7.2.01	**7.902772**	**7.901989**	7.904945	7.904525	**7.902454**
boat.512	**7.901908**	**7.901879**	**7.903091**	7.900712	**7.903088**
elaine.512	7.901122	**7.902989**	**7.901859**	**7.902030**	**7.901720**
gray21.512	7.900170	7.905107	**7.901832**	**7.902149**	**7.902688**
numbers.512	7.903615	7.892351	**7.902144**	7.903579	**7.901657**
ruler.512	**7.903265**	**7.903001**	**7.901937**	7.901428	**7.903052**
testpat.1k	**7.902806**	**7.901681**	7.903856	**7.903343**	**7.902752**
Mean	7.902308	7.901923	7.903764	7.902167	7.902488
Pass rate	18/28	20/28	17/28	11/28	26/28

$h^*_{left}/h^*_{right} = 7.901515698/7.903422936$.

$$\mathrm{NPCR}(C_1, C_2) = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{D(i,j)}{L} \times 100\% \tag{14}$$

$$\mathrm{UACI}(C_1, C_2) = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|C_1(i,j) - C_2(i,j)|}{T \times L} \times 100\% \tag{15}$$

$$D(i,j) = \begin{cases} 0, & \text{if } C_1(i,j) = C_2(i,j) \\ 1, & \text{if } C_1(i,j) \neq C_2(i,j) \end{cases} \tag{16}$$

where L is the number of pixels and T is the largest allowed pixel value in the images.

Recently, new criteria for the NPCR and UACI tests have been developed [29]. For the NPCR test, an encryption algorithm passes the NPCR test if its NPCR score is greater than a one-side hypothesis test under the significance level α defined by Eq. (17)

$$\mathcal{N}_\alpha^* = \frac{L - \Phi^{-1}(\alpha)\sqrt{L/T}}{L+1} \tag{17}$$

where $\Phi^{-1}(\cdot)$ is the inverse CDF of the standard Normal distribution $\mathbb{N}(0,1)$.

For the UACI test, an encryption algorithm can pass the UACI test if its actual UACI score falls into the interval of $(\mathcal{U}_\alpha^{*-}, \mathcal{U}_\alpha^{*-})$ defined by Eq. (18).

$$\begin{cases} \mathcal{U}_\alpha^{*-} = \mu_\mathcal{U} - \Phi^{-1}(\alpha/2)\sigma_\mathcal{U} \\ \mathcal{U}_\alpha^{*+} = \mu_\mathcal{U} + \Phi^{-1}(\alpha/2)\sigma_\mathcal{U} \end{cases} \tag{18}$$

$$\mu_\mathcal{U} = \frac{L+2}{3L+3} \tag{19}$$

$$\sigma_\mathcal{U} = \frac{(L+2)(L^2+2L+3)}{18(L+1)^2LT} \tag{20}$$

Among all 28 test images in the USC-SIPI 'Miscellaneous' image dataset, there are 6 images with the size of 256×256, 18 images with the size of 512×512, and 4 images with the size of 1024×1024. For each test image, we change one bit of a randomly selected pixel to generate a new plaintext image, and then encrypt both plaintext images using five encryption algorithms with the same security key. To provide a fair comparison, for Zhou's [35] and Wu's [31] algorithms, we directly use their NPCR and UACI test scores presented in [35,31]; and for other algorithms, we implement them in MATLAB R2010a and calculate the NPCR and UACI scores. Tables 4 and 5 show the NPCR and UACI results of five algorithms under the significance level $\alpha = 0.05$. We can see that most images encrypted by the proposed CMT-IEA can pass the NPCR and UACI tests. These prove that the encrypted images by CMT-IEA have a good diffusion property and can withstand the differential attack.

6.5. Robustness to noise and data loss

When a digital image is transmitted through networks or stored in the physical media, it is easily contaminated by noise or may have the data loss. An image encryption algorithm should have the robustness to resist noise and the data loss. In the proposed CMT-IEA, the encryption and decryption processes are asymmetric. In the encryption process, one pixel change in the plaintext image will spread over all pixels in the ciphertext image. However, in the decryption procedure, the change of one pixel in the ciphertext image can effect only few pixels in the recovered result. Thus, CMT-IEA has the ability to decrypt the ciphertext image with noise or the data loss. Fig. 13 shows the robustness analysis results of CMT-IEA against noise and the data loss. As can be seen, when the ciphertext images are with different types of noises or different sizes of the data loss, the decryption process of CMT-IEA can still recover the original image. Although the recovered images are with some noises, we can still recognize most of the image information.

Table 4
The NPCR results of different image encryption algorithms (significance level $\alpha = 0.05$).

Image size	256×256 $\geqslant 99.5693$	512×512 $\geqslant 99.5893$	1024×1024 $\geqslant 99.5994$	Pass rate
Wu's [31]	6/6	18/18	4/4	28/28
Zhou's [35]	6/6	17/18	4/4	27/28
Wu's [30]	6/6	17/18	3/4	26/28
Liao's [16]	0/6	0/18	0/4	0/28
CMT-IEA	6/6	18/18	4/4	28/28

Table 5
The UACI results of different image encryption algorithms (significance level $\alpha = 0.05$).

Image size	256×256 33.2255/33.7016	512×512 33.3730/33.5541	1024×1024 33.4183/33.5088	Pass rate
Wu's [31]	5/6	18/18	4/4	27/28
Zhou's [35]	1/6	4/18	2/4	7/28
Wu's [30]	6/6	15/18	4/4	25/28
Liao's [16]	0/6	0/18	0/4	0/28
CMT-IEA	6/6	17/18	4/4	27/28

Fig. 13. Robustness analysis results of noise and the data loss. (a) The ciphertext image C and its decrypted image $D = Dec(C, K)$; (b) the ciphertext image C_2 with 0.1% speckle noise and its decrypted result $D_2 = Dec(C_2, K)$; (c) the ciphertext image C_3 with 2% salt&pepper noise and its decrypted result $D_3 = Dec(C_3, K)$; (d) the ciphertext image C_4 with 0.1526% data loss and its decrypted result $D_4 = Dec(C_4, K)$; and (e) the ciphertext image C_5 with 1.3733% data loss and its decrypted result $D_5 = Dec(C_5, K)$.

7. Conclusion

This paper has proposed a new 2D chaotic map, 2D-SLMM. It was derived from the Sine and Logistic maps. Several assessment methods, including the trajectory, Lyapunov exponent and Kolmogorov entropy, have been used to evaluate the chaotic performance of 2D-SLMM. Analysis and evaluation results have shown that 2D-SLMM has the wider chaotic range, better ergodicity and hyperchaotic property, and that it has better chaotic performance than existing chaotic maps.

To demonstrate the performance of 2D-SLMM in security applications, a chaotic magic transform (CMT) has been introduced. It can quickly shuffle neighboring pixels within an image. Using 2D-SLMM and CMT, we have proposed a new image encryption algorithm. The experimental results have shown that the proposed algorithm can protect different types of images with a high security level and low time complexity.

Acknowledgements

The authors wound like to thank the editor and anonymous reviews for their valued comments and suggestions which helped to improve the manuscript. This work was supported in part by the Macau Science and Technology Development Fund under Grant FDCT/017/2012/A1 and by the Research Committee at University of Macau under Grants MYRG2014-00003-FST, MRG017/ZYC/2014/FST, MYRG113(Y1-L3)-FST12-ZYC and MRG001/ZYC/2013/FST.

References

[1] D. Arroyo, J. Diaz, F.B. Rodriguez, Cryptanalysis of a one round chaos-based substitution permutation network, Signal Process. 93 (2013) 1358–1364.
[2] D. Arroyo, R. Rhouma, G. Alvarez, S. Li, V. Fernandez, On the security of a new image encryption scheme based on chaotic map lattices, Chaos: Interdiscip. J. Nonlinear Sci. 18 (2008).
[3] G. Bhatnagar, Q.M.J. Wu, B. Raman, Discrete fractional wavelet transform and its application to multiple encryption, Inform. Sci. 223 (2013) 297–316.
[4] G. Chen, Y. Mao, C.K. Chui, A symmetric image encryption scheme based on 3D chaotic cat maps, Chaos Solitons Fractals 21 (2004) 749–761.
[5] J. Daemen, V. Rijmen, The Block Cipher Rijndael, 2000.
[6] J.P. Eckmann, D. Ruelle, Ergodic theory of chaos and strange attractors, Rev. Mod. Phys. 57 (1985) 617.
[7] P. Faure, H. Korn, A new method to estimate the Kolmogorov entropy from recurrence plots: its application to neuronal signals, Physica D 122 (1998) 265–279.

[8] FIPS PUB 197, Advanced Encryption Standard (AES), 2001.
[9] FIPS PUB 46, Data Encryption Standard (DES), 1999.
[10] M. Francois, T. Grosges, D. Barchiesi, R. Erra, A new image encryption scheme based on a chaotic function, Signal Process.: Image Commun. 27 (2012) 249–259.
[11] P. Grassberger, I. Procaccia, Estimation of the Kolmogorov entropy from a chaotic signal, Phys. Rev. A 28 (1983) 2591–2593.
[12] R.C. Hilborn, Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers, second ed., Oxford University Press, USA, 2001.
[13] A. Kanso, H. Yahyaoui, M. Almulla, Keyed hash function based on a chaotic map, Inform. Sci. 186 (2012) 249–264.
[14] N.V. Kuznetsov, T.N. Mokaev, P.A. Vasilyev, Numerical justification of Leonov conjecture on Lyapunov dimension of Rossler attractor, Commun. Nonlinear Sci. Numer. Simul. 19 (2014) 1027–1034.
[15] C. Li, L. Zhang, R. Ou, K.W. Wong, S. Shu, Breaking a novel colour image encryption algorithm based on chaos, Nonlinear Dyn. 70 (2012) 2383–2388.
[16] X. Liao, S. Lai, Q. Zhou, A novel image encryption algorithm based on self-adaptive wave transmission, Signal Process. 90 (2010) 2714–2722.
[17] C. Ling, X. Wu, S. Sun, A general efficient method for chaotic signal estimation, IEEE Trans. Signal Process. 47 (1999) 1424–1428.
[18] C. Shen, S. Yu, J. Lu, G. Chen, Designing hyperchaotic systems with any desired number of positive Lyapunov exponents via a simple model, IEEE Trans. Circuits Syst. I Regul. Pap. 61 (2014) 2380–2389.
[19] I.I. Shevchenko, Lyapunov exponents in resonance multiplets, Phys. Lett. A 378 (2014) 34–42.
[20] A. Skrobek, Cryptanalysis of chaotic stream cipher, Phys. Lett. A 363 (2007) 84–90.
[21] E. Solak, C. Cokal, Algebraic break of image ciphers based on discretized chaotic map lattices, Inform. Sci. 181 (2011) 227–233.
[22] S. Tedmori, N. Al-Najdawi, Image cryptographic algorithm based on the Haar wavelet transform, Inform. Sci. 269 (2014) 21–34.
[23] X. Wang, L. Teng, X. Qin, A novel colour image encryption algorithm based on chaos, Signal Process. 92 (2012) 1101–1108.
[24] X. Wang, W. Zhang, W. Guo, J. Zhang, Secure chaotic system with application to chaotic ciphers, Inform. Sci. 221 (2013) 555–570.
[25] Wikipedia, Double-precision Floating-point Format – Wikipedia, the Free Encyclopedia, 2013 (online; accessed 10.12.13).
[26] Wikipedia, Lorenz System – Wikipedia, the Free Encyclopedia, 2013 (online; accessed 01.01.14).
[27] A. Wolf, J.B. Swift, H.L. Swinney, J.A. Vastano, Determining Lyapunov exponents from a time series, Physica D 16 (1985) 285–317.
[28] X. Wu, H. Hu, B. Zhang, Parameter estimation only from the symbolic sequences generated by chaos system, Chaos Solitons Fractals 22 (2004) 359–366.
[29] Y. Wu, J.P. Noonan, S. Agaian, NPCR and UACI randomness tests for image encryption, Cyber J. Multidiscip. (2011) 31–38.
[30] Y. Wu, J.P. Noonan, S. Agaian, A wheel-switch chaotic system for image encryption, in: 2011 International Conference on System Science and Engineering (ICSSE), 2011, pp. 23–27.
[31] Y. Wu, G. Yang, H. Jin, J.P. Noonan, Image encryption using the two-dimensional logistic chaotic map, J. Electron. Imaging 21 (2012). 013014-1–013014-15.
[32] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J.P. Noonan, P. Natarajan, Local Shannon entropy measure with statistical tests for image randomness, Inform. Sci. 222 (2013) 323–342.
[33] W. Xu, Z. Geng, Q. Zhu, X. Gu, A piecewise linear chaotic map and sequential quadratic programming based robust hybrid particle swarm optimization, Inform. Sci. 218 (2013) 85–102.
[34] Y.Q. Zhang, X.Y. Wang, A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice, Inform. Sci. 273 (2014) 329–351.
[35] Y. Zhou, L. Bao, C.L.P. Chen, Image encryption using a new parametric switching chaotic system, Signal Process. 93 (2013) 3039–3052.
[36] Y. Zhou, L. Bao, C.L.P. Chen, A new 1D chaotic system for image encryption, Signal Process. 97 (2014) 172–182.
[37] Y. Zhou, W. Cao, C.L. Philip Chen, Image encryption using binary bitplane, Signal Process. 100 (2014) 197–207.
[38] Y. Zhou, K. Panetta, S. Agaian, C.L.P. Chen, Image encryption using P-Fibonacci transform and decomposition, Opt. Commun. 285 (2012) 594–608.
[39] Y. Zhou, K. Panetta, S. Agaian, C.L.P. Chen, (n, k, p)-Gray code for image systems, IEEE Trans. Cybernet. 43 (2013) 515–529.
[40] Z.l. Zhu, W. Zhang, K.w. Wong, H. Yu, A chaos-based symmetric image encryption scheme using a bit-level permutation, Inform. Sci. 181 (2011) 1171–1186.