Signal Processing 149 (2018) 148–161

Contents lists available at ScienceDirect

Signal Processing

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

2D Logistic-Sine-coupling map for image encryption

Zhongyun Hua, Fan Jin, Binxuan Xu, Hejiao Huang*

School of Computer Science and Technology, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen 518055, China

ARTICLE INFO

Article history:
Received 20 November 2017
Revised 4 March 2018
Accepted 16 March 2018
Available online 16 March 2018

Keywords:
Chaos
Cryptography
Image encryption
Multimedia security

ABSTRACT

Image encryption is a straightforward strategy to protect digital images by transforming images into unrecognized ones. The chaos theory is a widely used technology for image encryption as it has many significant properties such as ergodicity and initial state sensitivity. When chaotic systems are used in image encryption, their chaos performance highly determines the security level. This paper presents a two-dimensional (2D) Logistic-Sine-coupling map (LSCM). Performance estimations demonstrate that it has better ergodicity, more complex behavior and larger chaotic range than several newly developed 2D chaotic maps. Utilizing the proposed 2D-LSCM, we further propose a 2D-LSCM-based image encryption algorithm (LSCM-IEA), which adopts the classical confusion-diffusion structure. A permutation algorithm is designed to permutate image pixels to different rows and columns while a diffusion algorithm is developed to spread few changes of plain-image to the whole encrypted result. We compare the efficiency of LSCM-IEA with several advanced algorithms and the results show that it has higher encryption efficiency. To show the superiority of LSCM-IEA, we also analyze the security of LSCM-IEA in terms of key security, ability of defending differential attack, local Shannon entropy and contrast analysis. The analysis results demonstrate that LSCM-IEA has better security performance than several existing algorithms.

1. Introduction

With the rapid development of digital technology, more and more multimedia information is generated and spread in the Internet [1]. Among all these multimedia information, digital image is an information format that can carry information with visualized way. For these digital images transmitted in networks, many of them are private images. For example, the personal medical images are usually private images, as they contain the information of personal healthy conditions. If these private images are obtained by some unauthorized ways, serious security disasters may happen. Thus, it is important to protect these private images [2–4] and image encryption is one efficient technology to protect them [5–8].

One strategy of encrypting image is to treat an image as a binary data stream and then use the developed data encryption algorithms to encrypt the data stream. These algorithms include the well-known data encryption standard [9], advanced encryption standard [10]. However, image data has many unique characteristics such as large data volume, high correlation and strong redundancy [11,12]. Treating an image as a binary stream will miss these characteristics, and thus may make the encryption inefficient. To address this issue, many image encryption schemes considering image features have been proposed using various technologies, such as the chaos theory [13–16], DNA coding [17,18], quantum theory [19,20], compressive sensing [21,22] and some mathematics models [23,24]. Among these technologies, chaos theory is the most popular one. This is because chaotic behavior has many unique properties that are similar with the principles of image encryption [25–27]. Specifically, the ergodicity and initial state sensitivity of chaos theory correspond to the confusion and diffusion properties of encryption [28]. Some examples of chaos-based encryption schemes are as follows. In [29], Zhou et al. first proposed a new chaotic system that can use existing chaotic maps as seed maps to generate new chaotic maps, and then used one newly generated chaotic map to design an image encryption algorithm. In [30], Pak and Huang proposed a new color image encryption algorithm using the combination of Logistic, Sine and Chebyshev maps. In [21], Zhou et al. proposed a new image security scheme using hyperchaotic system and compressive sensing technology. This scheme can perform image encryption and image compression simultaneously.

For these chaos-based image encryption algorithms, their security is determined by the structure of encryption algorithms and the chaos performance of their used chaotic maps. On one hand, if the designed encryption structures are not secure enough, the encryption algorithms can be successfully broken using different security attacks [31–33]. On the other hand, with the fast

* Corresponding author.
E-mail addresses: huazyum@gmail.com, huazhongyun@hit.edu.cn (Z. Hua), huanghejiao@hit.edu.cn, hjhuang@aliyun.com (H. Huang).

https://doi.org/10.1016/j.sigpro.2018.03.010
0165-1684/© 2018 Published by Elsevier B.V.

development of discerning chaos methodology, researchers found that some existing chaotic maps have security problems if they have weak chaos performance [34–36]. This will also cause security problems to the corresponding chaos-based encryption algorithms [37,38]. Thus, designing encryption structures with higher security and developing new chaotic systems with better chaos performance can significantly promote the chaos-based image encryption.

To design new chaotic map with better chaos performance for image encryption, this paper presents a two-dimensional (2D) Logistic-Sine-coupling map (2D-LSCM). It is generated by first coupling the Logistic and Sine maps, and then extending the dimension from one-dimensional (1D) to 2D. Chaos performance estimations demonstrate that 2D-LSCM has better ergodicity, more complex chaotic behavior and wider chaotic interval than several newly designed 2D chaotic maps. Using 2D-LSCM, we further present a 2D-LSCM-based image encryption algorithm (LSCM-IEA). The secret key is to obtain the initial states of the 2D-LSCM, and then produce chaotic sequences. The chaotic sequences are used to do permutation and diffusion operations to the plain-image. Simulation results prove the ability of LSCM-IEA. Efficiency evaluation shows that it can achieve faster encryption speed than several other algorithms. The security analysis demonstrates that LSCM-IEA can outperform several advanced image encryption algorithms in security performance.

We organize the rest of this paper as follows. Section 2 introduces the proposed 2D-LSCM and evaluates its chaos performance. Section 3 presents the developed image encryption algorithm, LSCM-IEA. Section 4 simulates LSCM-IEA and analyzes its efficiency. Section 5 analyzes the security level of LSCM-IEA and Section 6 concludes this paper.

2. 2D Logistic-Sine-coupling map

This section presents a novel 2D chaotic map, called 2D Logistic-Sine-coupling map (2D-LSCM), and then discusses its chaotic complexity.

2.1. Definition of 2D-LSCM

The 2D-LSCM is derived from two existing 1D chaotic maps, namely the Logistic map [39] and the Sine map [29]. The Logistic map is defined as

$$x_{i+1} = 4\eta x_i(1 - x_i), \tag{1}$$

where its control parameter $\eta \in [0, 1]$. The Sine map is given as

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

where β is a control parameter and it also has an interval of [0,1].

The Logistic and Sine maps have many disadvantages such as simple behaviors and frail chaotic intervals, and these disadvantages may bring negative effects for some chaos-based applications [40]. However, when coupling the Logistic and Sine maps, we can obtain a new chaotic map with quite complex chaos, namely 2D-LSCM, which can be defined as

$$\begin{cases} x_{i+1} = \sin(\pi(4\theta x_i(1 - x_i) + (1 - \theta)\sin(\pi y_i))); \\ y_{i+1} = \sin(\pi(4\theta y_i(1 - y_i) + (1 - \theta)\sin(\pi x_{i+1}))), \end{cases} \tag{3}$$

where θ is the control parameter and $\theta \in [0, 1]$. As can be observed from its definition, the 2D-LSCM is obtained by first coupling the Logistic and Sine maps together, and then performing a sine transform to the coupling result, and last extending the dimension from 1D to 2D. By this way, the complexity of the Logistic map and Sine map can be sufficiently mixed, which can obtain complex chaotic behavior.

2.2. Performance evaluation

The proposed 2D-LSCM can inherently enhance the chaos performance of the Logistic and Sine maps. To show its superiority, we evaluate its chaos performance and compare it with several newly generated 2D chaotic maps. The evaluations are performed in terms of chaos trajectory, Lyapunov exponent [41], and Kolmogorov entropy [42].

2.2.1. Chaos trajectory

Trajectory demonstrates the motion starting from a given initial state with the time increases. The trajectory of a periodic motion is a closed curve and the trajectory of a chaotic behavior will never close or repeat in theory. Thus, the chaos trajectory usually occupies a part of phase space and it can reflect the randomness of the outputs of a chaotic system. A chaotic system has better random outputs if its chaos trajectory can occupy a larger phase space.

Fig. 1shows the trajectories of four 2D chaotic maps. When plotting these trajectories, all the initial states are set as (0.8,0.5) and the control parameters are selected as the settings that can make the corresponding chaotic maps obtain their best chaos performance. Specifically, the control parameters of the 2D Logistic map, 2D Sine Logistic modulation map (2D-SLMM) [40], 2D Logistic-adjusted-Sine map (2D-LASM) [43], and 2D-LSCM are set as 1.19, 1, 0.9 and 0.99, respectively. To show the actual behaviors of chaotic systems in stable state, we plot the iteration points from 5000 to 35,000 in each trajectory. One can see from Fig. 1 that the trajectories of the 2D Logistic map and 2D-SLMM only occupy a small space in the phase plane, while that of the 2D-LASM and 2D-LSCM can occupy all phase plane. Besides, It is obvious that the points of the 2D-LSCM distribute more uniform than that of the 2D-LASM. Thus, the proposed 2D-LSCM can generate more random output sequences than other three chaotic maps.

2.2.2. Lyapunov exponent

The initial state sensitivity is the most obvious feature of chaotic behavior. The Lyapunov exponent (LE) [41] can provide a quantitative description to the initial state sensitivity. For two trajectories of a chaotic system beginning with two close initial states, the LE describes their average separation rate. For a differentiable one-dimensional dynamical system $x_{i+1} = f(x_i)$, its LE can be defined as

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln\left|f'(x_i)\right|. \tag{4}$$

A high-dimensional dynamical system has more than one LE and the maximum LE (MLE) determines whether a high-dimensional system has chaotic behavior or not. A positive MLE means that the close trajectories of a dynamical system diverge in each unit time and will evolve to completely different trajectories with the increasement of time. Thus, a dynamical system is chaotic if its MLE is positive and larger MLE means better performance. If a dynamical system can obtain more than one positive LE, its trajectories will diverge in multi-directions, which makes it has hyperchaotic behavior. The hyperchaotic behavior is a much more complicated motion than the chaotic behavior.

A 2D chaotic system has two LEs and Fig. 2 plots the two LEs of different 2D chaotic maps with the change of their control parameters. One can observe that the 2D-SLMM has chaotic behavior when $\alpha \in (0.84, 1)$, and has hyperchaotic behavior when $\alpha \in (0.91, 1)$, the 2D-LASM has chaotic behavior when $\mu \in (0.32, 1)$, and has hyperchaotic behavior when $\mu \in (0.45, 1)$, the 2D-LSCM has chaotic behavior when $\theta \in (0, 1)$, and has hyperchaotic behavior when $\theta \in (0, 0.34) \cup (0.67, 1)$. This shows that the proposed 2D-LSCM has much wider chaotic range and hyperchaotic range than

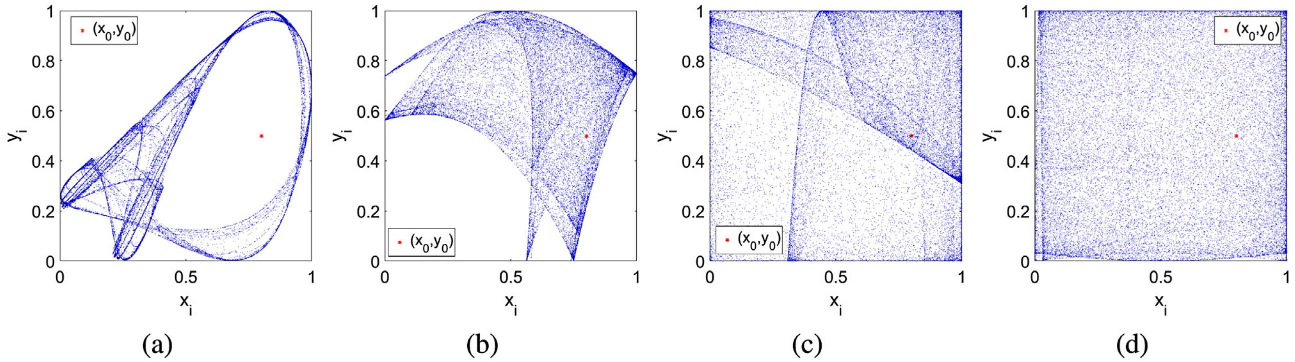


Fig. 1. Trajectories of four 2D chaotic maps: (a) the 2D Logistic map with parameter $r = 1.19$; (b) the 2D-SLMM with parameter $\alpha = 1$; (c) the 2D-LASM with parameter $\mu = 0.9$; (d) the 2D-LSCM with parameter $\theta = 0.99$.

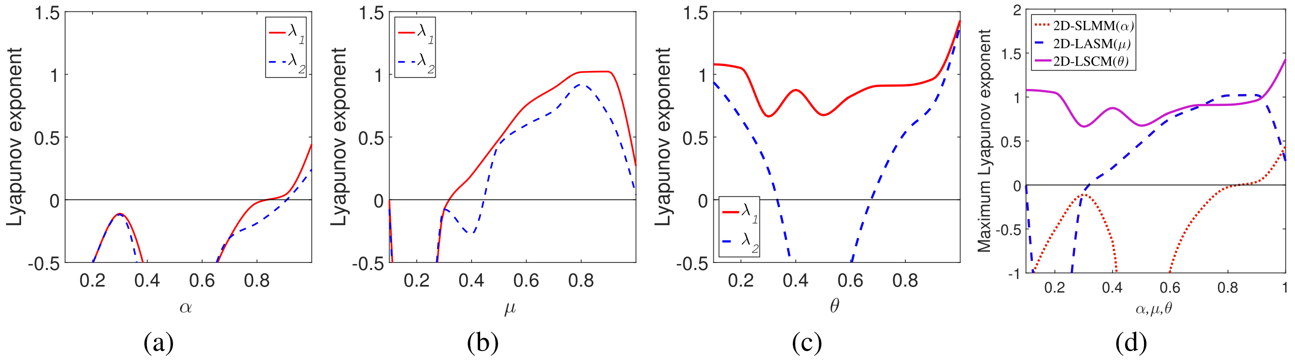


Fig. 2. The two LEs of different 2D chaotic maps: (a) the 2D-SLMM; (b) the 2D-LASM; (c) the 2D-LSCM; (d) the MLE comparison of 2D-SLMM, 2D-LASM and 2D-LSCM.

the other chaotic maps. Besides, Fig. 2(d) compares the MLEs of different chaotic maps. It shows that 2D-LSCM has the largest MLE in most parameter settings. This further demonstrate that the proposed 2D-LSCM has more complex chaotic behavior.

2.2.3. Kolmogorov entropy

The Kolmogorov entropy (KE) is a type of entropy that describes the state evolution of dynamical system [42]. It can be used to measure the degree of chaos by testing the needed extra information of predicting the future trajectory using the previous states. Dividing the n-dimensional phase space into a number of boxes $(i_0, i_1, \ldots, i_n)$ with ε size, the KE can be described as

$$K = -\lim_{\tau\to 0}\lim_{\varepsilon\to 0}\lim_{n\to\infty}\frac{1}{n\tau}\sum_{i_0,i_1,\ldots,i_n} p(i_1,\ldots,i_n)\ln p(i_1,\ldots,i_n), \tag{5}$$

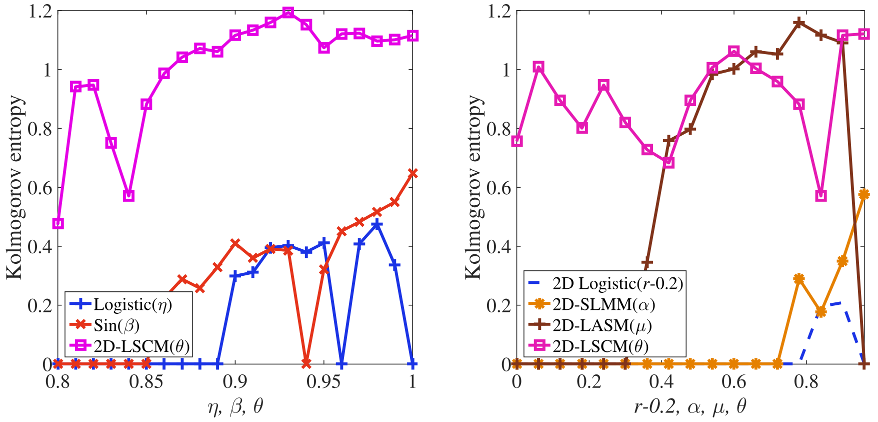
where n is the embedding dimension, τ is the time delay, and $p(i_1, \ldots, i_n)$ represents the joint probability when the trajectory of system is in i_0 at the starting time, in i_1 at the time τ, …, and in i_n at the time $n\tau$. A positive KE indicates that extra information is required to predict the trajectory of the dynamical system and larger KE demonstrates more required information. Thus, a dynamical system is unpredictable if it has a positive KE and larger KE indicates better unpredictability.

Our experiment uses the Grassberger method provided in [42] to calculate the KEs of different chaotic systems and Fig. 3 plots the obtaining results. One can see that Fig. 3(a) plots the KEs of the Logistic map, Sine map and 2D-LSCM, while Fig. 3(b) compares the KEs of the 2D Logistic map, 2D-SLMM, 2D-LASM and 2D-LSCM. To provide a better comparison environment, we shift the parameter of the 2D Logistic map when plotting its KEs. As can be seen from Fig. 3(a), although 2D-LSCM is derived from the Logistic and Sine maps, it has much better unpredictability than the Logistic and Sine maps. From Fig. 3(b), we can see that the proposed 2D-LSCM has positive KEs in the whole parameter range. It can achieve larger KEs than 2D-LASM in most parameter settings and can outperform 2D Logistic map and 2D-SLMM in all the parameter ranges. This sufficiently proves that the proposed 2D-LSCM has good unpredictability.

2.2.4. Dynamical degradation analysis

For a dynamical system with chaotic behavior, its trajectory will never close or repeat in theory. However, as the finite precision domain cannot own infinite states, the close states in the phase plane will overlap when a chaotic map is digitalized in the finite precision platforms. This phenomenon is known as the dynamical degradation [44]. The dynamical degradation is unavoidable for digitalized chaotic maps and it causes many negative effects for chaos-based applications. However, many chaos-based applications only use finite states of a chaotic trajectory. Chaotic maps with dynamical degradation are still available to these applications if the cycle lengths of the digitalized chaotic maps are larger than the required cycle lengths.

To investigate the dynamical degradation of different chaotic maps, we calculate the cycle lengths of these chaotic maps using different precisions. For each chaotic map, we first randomly generate a number groups of initial states, where the control parameters are all within the chaotic ranges, and then generate trajectories using these initial states under different precisions, and finally calculate the average cycle lengths of these trajectories. Table 1 lists the average cycle lengths of different chaotic maps under various precisions. One can observe that our proposed 2D-LSCM can obtain the largest average cycle lengths under most precisions. Its cycle length fast increases with the increasement of precision and it can achieve 4,455,734 under the precision 10^{-8}. As the precisions of the commonly used platforms are usually much higher than 10^{-8}, the cycle lengths of 2D-LSCM in these platforms are far larger than 4455734. On the other hand, when chaotic map is used in im-



(a) (b)

Fig. 3. KE comparison of (a) the Logistic map, Sine map and 2D-LSCM; (b) the 2D Logistic map, 2D-SLMM, 2D-LASM and 2D-LSCM.

Table 1
The average cycle lengths of different chaotic maps with different precisions.

Precisions	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
2D Logistic	324	1432	8215	45,266	259,535
2D-SLMM	796	4858	36,959	339,576	3,127,743
2D-LASM	74	555	4474	39,522	195,389
2D-LSCM	666	5431	40,544	413,618	4,455,734

Table 2
NIST SP800-22 test results of binary sequences generated using 2D-LSCM.

		P-value	Result
Sub-tests		≥ 0.01	
Approximate Entropy($m = 10$)		0.704009	Pass
Block Frequency($M = 128$)		0.993633	Pass
Cumulative Sums	Forward	0.069202	Pass
	Reverse	0.077280	Pass
FFT		0.840006	Pass
Frequency		0.076727	Pass
Linear Complexity($M = 500$)		0.504113	Pass
Longest Run		0.447729	Pass
Non-Overlapping Template($m = 9$)[a]		0.472143	Pass
Overlapping Template($m = 9$)		0.936519	Pass
Random Excursions[a]		0.217525	Pass
Random Excursions Variant[a]		0.416696	Pass
Rank		0.151412	Pass
Runs		0.740543	Pass
Serial($m = 16$)	*P*-value1	0.163838	Pass
	P-value2	0.400104	Pass
Universal		0.958368	Pass

[a] The average values of multiple tests.

age encryption, the number of required chaotic outputs approximates to the size of the image, e.g. almost 1,000,000 chaotic outputs for an image of size 1000 × 1000. Thus, in the commonly used platforms, the cycle lengths of 2D-LSCM are larger than the cycle lengths required in image encryption.

To further show that the proposed 2D-LSCM is suitable for designing image encryption algorithm, we use the National Institute of Standards and Technology (NIST) SP800-22 [45] to test the randomness of the output sequences of 2D-LSCM. The NIST SP800-22 has 15 sub-tests and each sub-test can generate a *P*-value. According to the recommendation of Bassham et al. [45], 100 binary streams with 1,000,000 bits are suggested as input and the generated *P*-value is expected to fall into the range [0.01,1] to pass the corresponding sub-test. Our experiment uses the double float data format to present the iterative outputs of 2D-LSCM. For each output of 2D-LSCM, we transform its fractional part to be a binary stream with 49 bits. The input binary streams are obtained by combining these binary streams from the outputs. Table 2 shows the test results and one can see that binary streams obtained from the outputs of 2D-LSCM can pass all the sub-tests. This indicates that 2D-LSCM can generate a long sequence of aperiodic outputs, which are suitable for image encryption.

3. 2D-LSCM-based image encryption algorithm

Using the developed 2D-LSCM, this section presents a 2D-LSCM-based image encryption algorithm (LSCM-IEA) and its structure is shown in Fig. 4. The secret key is to generate initial state of the 2D-LSCM, and the chaotic matrices generated by 2D-LSCM are used to do 2D-LSCM permutation and 2D-LSCM diffusion. The 2D-LSCM permutation can efficiently shuffle pixel positions and the 2D-LSCM diffusion can completely change pixel values and spread few changes in plain-image to the whole cipher-image. As the 2D-LSCM permutation can achieve excellent confusion property and the 2D-LSCM diffusion can obtain good diffusion, two rounds of permutation and diffusion can obtain a high security encryption result in theory. However, more encryption rounds can achieve higher security results. Our proposed LSCM-IEA uses four encryption rounds, as four encryption rounds can obtain high security encryption results and can balance the trade-off between the efficiency and security. Next, we will describe each of the encryption processes in detail.

3.1. Initial state generation

According to the discussion in [46], the key length of a chaos-based encryption algorithm should be larger than 100 bits to resist brute-force attack. We set the length of secret key as 256 bits in LSCM-IEA, considering the rapid enhancement of computer computing ability. Specially, $K = \{x_0, y_0, r, a_1, a_2, a_3, a_4\}$, where (x_0, y_0) are the initial values, r is the control parameter and $a_1 \sim a_4$ are the perturbation coefficients to change r in the four encryption rounds. The x_0, y_0 and r have size of 52 bits, and they can be converted to float numbers using the IEEE 754 Floating-Point standard. Suppose $b_1 b_2 \ldots b_{52}$ is a 52-bit binary string, the conversion equation is as follows,

$$v = \sum_{i=1}^{52} b_i 2^{-i}. \tag{6}$$

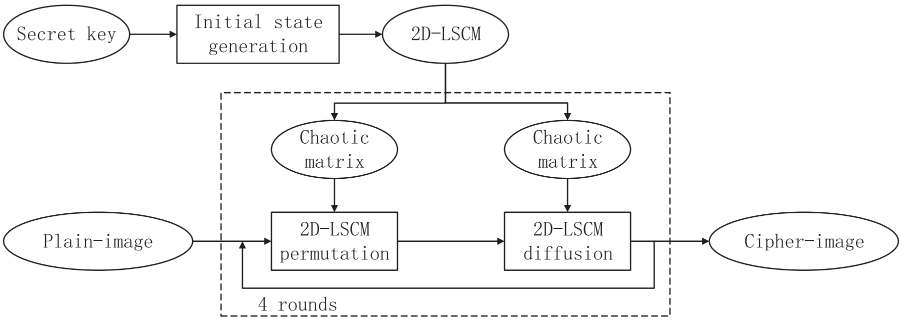


Fig. 4. The structure of LSCM-IEA.

The a_i $(i = 1, 2, 3, 4)$ is an integer that can be obtained by directly transforming a 25-bit binary string to a decimal integer.

The secret key is to generate four initial states for 2D-LSCM. The initial value of the first encryption round $(x_0^{(1)}, y_0^{(1)})$ is directly set as (x_0, y_0), and the initial values of the second, third and fourth encryption rounds are set as the last iteration state of 2D-LSCM in previous encryption round. The control parameters in four encryption rounds can be generated as

$$\begin{cases} r^{(1)} = (r \times a_1) \bmod 1; \\ r^{(2)} = (r \times a_2) \bmod 1; \\ r^{(3)} = (r \times a_3) \bmod 1; \\ r^{(4)} = (r \times a_4) \bmod 1. \end{cases} \tag{7}$$

Using the four initial states $(x_0^{(i)}, y_0^{(i)}, r^{(i)})$ $(i = 1, 2, 3, 4)$, the 2D-LSCM can generate chaotic matrices for the following 2D-LSCM permutation and 2D-LSCM diffusion.

3.2. 2D-LSCM permutation

High correlations and data redundancy may exist between adjacent pixels of a natural image, as the image pixel is usually represented using 8 or even more bits. An efficient image encryption algorithm should de-correlate these high correlations. Pixel permutation can randomly shuffle adjacent pixels to different positions and it can de-correlate their high correlations.

Most of the existing permutation operations shuffle image pixels row-by-row or column-by-column. Then each operation can only change a pixel's row position or column position. Multiple permutation operations are required to obtain a totally shuffled result. To obtain better shuffling efficiency, we designed a new 2D-LSCM permutation that can simultaneously shuffle the image's row and column positions in one operation. The detail procedure can be described as follows,

- *Step 1*: Suppose the plain-image **P** is of size $M \times N$, a chaotic matrix **S** of size $M \times N$ is generated using the 2D-LSCM with the initial state;
- *Step 2*: Sort each column of **S** and obtain the index matrix **O**;
- *Step 3*: Set row index $m = 1$;
- *Step 4*: Select the pixels in **P** with positions $\{(\mathbf{O}_{m,1}, 1), (\mathbf{O}_{m,2}, 2), \ldots, (\mathbf{O}_{m,N}, N)\}$;
- *Step 5*: Sort the values in **S** with positions $\{(\mathbf{O}_{m,1}, 1), (\mathbf{O}_{m,2}, 2), \ldots, (\mathbf{O}_{m,N}, N)\}$ and obtain an index vector **v**;
- *Step 6*: Shuffle these selected pixels in **P** using **v**;
- *Step 7*: Iterate *Step 3* to *Step 6* for $m = 2 \sim M$.

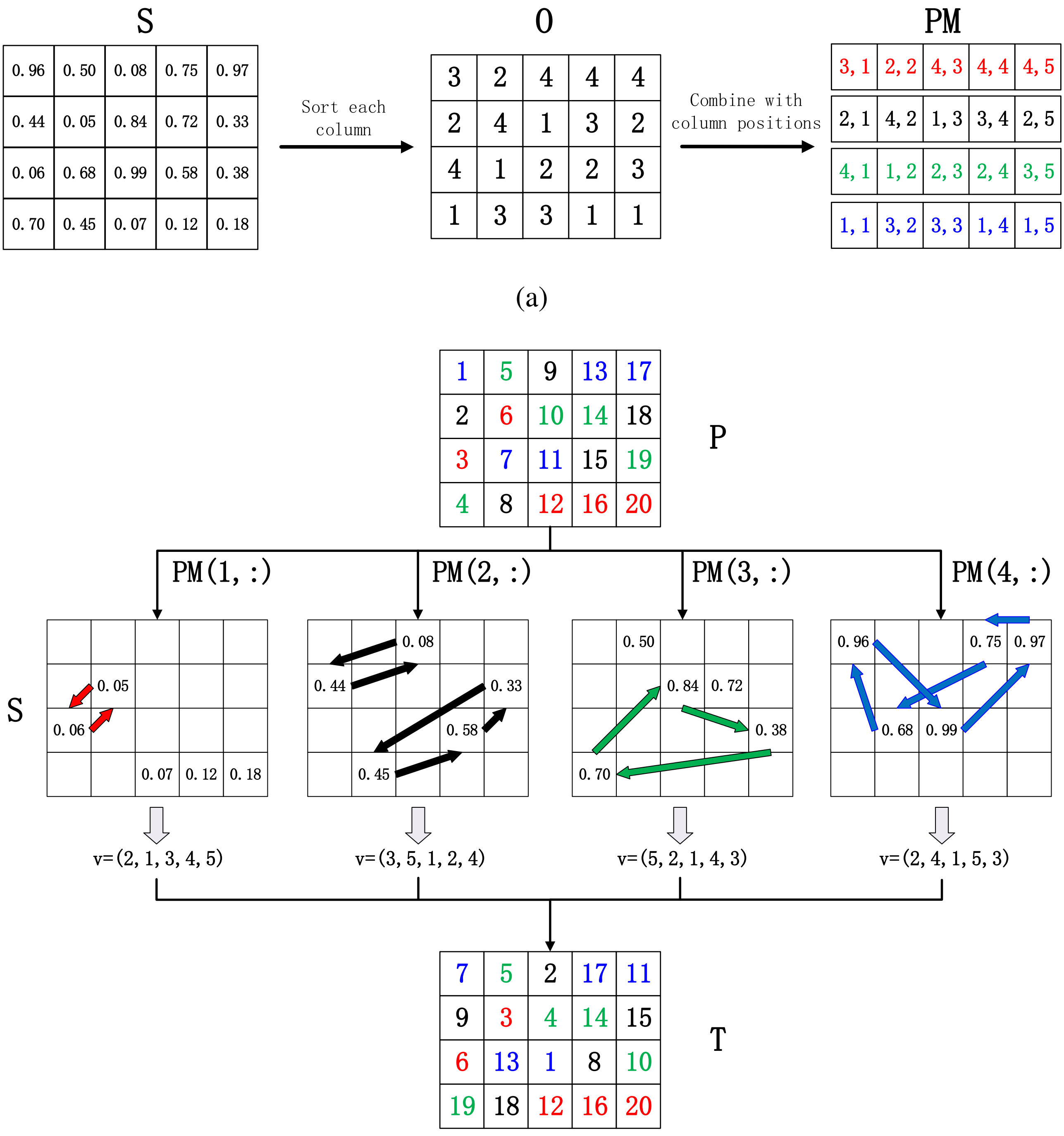
To better explain the procedure of 2D-LSCM permutation, we provide a numeral example with the image size of 4×5 and it is shown in Fig. 5. Fig. 5(a) shows the generation of permutation matrix **PM** from the chaotic sequence **S**. First, sort each column of **S** with ascending order to obtain the sorted result **S**′ and an index matrix **O**, where $\mathbf{S}'_{i,j} = \mathbf{S}_{\mathbf{O}_{i,j},j}$. Using the index matrix **O** as the row position, we can obtain a position matrix **PM**. Fig. 5(b) shows the detail pixel shuffling using **PM** and **S**. The detail pixel shuffling procedure can be described as follows.

- The 1-st row of **PM** is {(3, 1), (2, 2), (4, 3), (4, 4), (4, 5)}. Select the values in **S** with these positions and sort them with ascending order to obtain the index vector $\mathbf{v} = \{2, 1, 3, 4, 5\}$. Then use the obtained **v** to shuffle the pixels in **P** with these positions, namely $\mathbf{T}_{3,1} = \mathbf{P}_{2,2}$, $\mathbf{T}_{2,2} = \mathbf{P}_{3,1}$, $\mathbf{T}_{4,3} = \mathbf{P}_{4,3}$, $\mathbf{T}_{4,4} = \mathbf{P}_{4,4}$, $\mathbf{T}_{4,5} = \mathbf{P}_{4,5}$.
- The 2-nd row of **PM** is {(2, 1), (4, 2), (1, 3), (3, 4), (2, 5)}. Select the values in **S** with these positions and sort them with ascending order to obtain the index vector $\mathbf{v} = \{3, 5, 1, 2, 4\}$. Then use the obtained **v** to shuffle the pixels in **P** with these positions, namely $\mathbf{T}_{2,1} = \mathbf{P}_{1,3}$, $\mathbf{T}_{4,2} = \mathbf{P}_{2,5}$, $\mathbf{T}_{1,3} = \mathbf{P}_{2,1}$, $\mathbf{T}_{3,4} = \mathbf{P}_{4,2}$, $\mathbf{T}_{2,5} = \mathbf{P}_{3,4}$.
- The 3-rd row of **PM** is {(4, 1), (1, 2), (2, 3), (2, 4), (3, 5)}. Select the values in **S** with these positions and sort them with ascending order to obtain the index vector $\mathbf{v} = \{5, 2, 1, 4, 3\}$. Then use the obtained **v** to shuffle the pixels in **P** with these positions, namely $\mathbf{T}_{4,1} = \mathbf{P}_{3,5}$, $\mathbf{T}_{1,2} = \mathbf{P}_{1,2}$, $\mathbf{T}_{2,3} = \mathbf{P}_{4,1}$, $\mathbf{T}_{2,4} = \mathbf{P}_{2,4}$, $\mathbf{T}_{3,5} = \mathbf{P}_{2,3}$.
- The 4-th row of **PM** is {(1, 1), (3, 2), (3, 3), (1, 4), (1, 5)}. Select the values in **S** with these positions and sort them with ascending order to obtain the index vector $\mathbf{v} = \{2, 4, 1, 5, 3\}$. Then use the obtained **v** to shuffle the pixels in **P** with these positions, namely $\mathbf{T}_{1,1} = \mathbf{P}_{3,2}$, $\mathbf{T}_{3,2} = \mathbf{P}_{1,4}$, $\mathbf{T}_{3,3} = \mathbf{P}_{1,1}$, $\mathbf{T}_{1,4} = \mathbf{P}_{1,5}$, $\mathbf{T}_{1,5} = \mathbf{P}_{3,3}$.

Algorithm 1shows the pseudo-code of the 2D-LSCM permuta-

Algorithm 1 The 2D-LSCM permutation.

```
Input: The plain-image P and the chaotic matrix S. Both have the size $M \times N$.
Sort each column of S with ascending order and obtain O and S′, where $\mathbf{S}'_{i,j} = \mathbf{S}_{\mathbf{O}_{i,j},j}$;
Set $\mathbf{T} \in \mathbb{N}^{M\times N}$, $\mathbf{b} \in \mathbb{N}^{1\times N}$, $\mathbf{t} \in \mathbb{N}^{1\times N}$;
for $i = 1$ to $M$ do
    for $j = 1$ to $N$ do
        $\mathbf{t}_j = \mathbf{P}_{\mathbf{O}_{i,j},j}$, $\mathbf{b}_j = \mathbf{S}_{\mathbf{O}_{i,j},j}$;
    end for
    Sort b with ascending order and obtain v and b′, where $\mathbf{b}' = \mathbf{b}_{\mathbf{v}}$;
    for $j = 1$ to $N$ do
        $\mathbf{T}_{\mathbf{O}_{i,j},j} = \mathbf{t}_{\mathbf{v}_j}$;
    end for
end for
Output: The permuted result T.
```



(a)

(b)

Fig. 5. An example of 2D-LSCM permutation using the image **P** of size 4 × 5: (a) the generation procedure of permutation matrix **PM** from chaotic sequence **S**; (b) permutation to **P** using **PM** and **S**.

tion.

3.3. 2D-LSCM diffusion

An image encryption algorithm should have diffusion property, which means that slight change in plain-image can cause total difference in cipher-image. In the proposed LSCM-IEA, we designed a 2D-LSCM diffusion to achieve the diffusion property. Using the chaotic sequence generated by 2D-LSCM, the image pixels can be randomly changed. Using the two previous pixel values to change the current one, the 2D-LSCM diffusion can efficiently spread few changes of plain-image to the whole cipher-image. Suppose both the permutation result **T** and chaotic matrix **R** are with the size of $M \times N$, the 2D-LSCM diffusion is described as

$$\mathbf{C}_i = \begin{cases} (\mathbf{T}_1 + \mathbf{T}_G + \mathbf{T}_{G-1} + \lfloor \mathbf{R}_i \times 2^{32} \rfloor) \bmod F & \text{if } i = 1; \\ (\mathbf{T}_2 + \mathbf{C}_1 + \mathbf{T}_G + \lfloor \mathbf{R}_i \times 2^{32} \rfloor) \bmod F & \text{if } i = 2; \\ (\mathbf{T}_i + \mathbf{C}_{i-1} + \mathbf{C}_{i-2} + \lfloor \mathbf{R}_i \times 2^{32} \rfloor) \bmod F & \text{if } i \in [3, G], \end{cases} \tag{8}$$

where F is the number of allowed pixel values in plain-image **P**, e.g. $F = 256$ if **P** is 8-bit grayscale image, and the operation $\lfloor x \rfloor$ is to obtain the largest integer that is smaller than or equals to x.

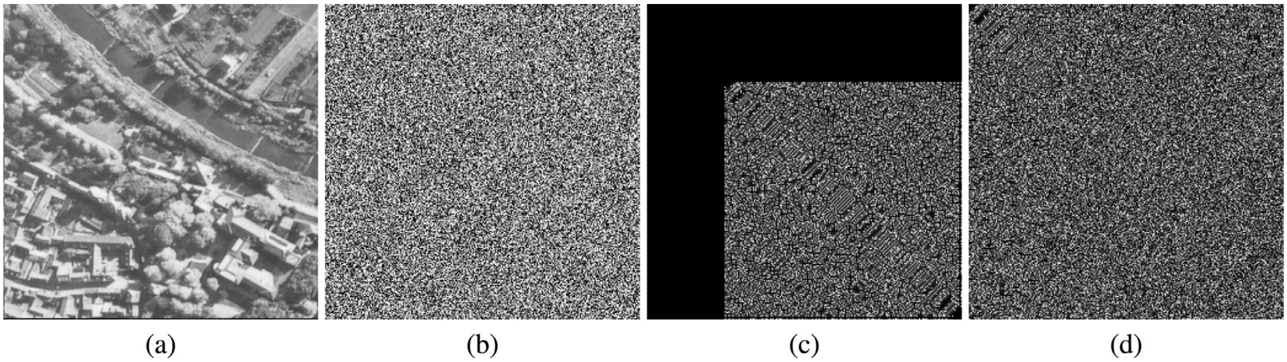


Fig. 6. Demonstration of 2D-LSCM diffusion: (a) plain-image $\mathbf{I}_1$; (b) 2D-LSCM diffusion result of $\mathbf{I}_1$; (c) the difference of 2D-LSCM diffusion results to $\mathbf{I}_1$ and $\mathbf{I}_2$, where $\mathbf{I}_2$ is another plain-image that has one pixel difference with $\mathbf{I}_1$ in position (64,64); (d) the difference of two rounds of 2D-LSCM diffusion to $\mathbf{I}_1$ and $\mathbf{I}_2$.

The 2D-LSCM diffusion can be divided into two steps: row diffusion and column diffusion. When doing the row diffusion, $G = N$ and Eq. (8) is applied to each row. When performing the column diffusion, $G = M$ and Eq. (8) is applied to each column.

The 2D-LSCM diffusion in the decryption process is the inverse of the forward operation. The inverse 2D-LSCM diffusion is defined as

$$\mathbf{T}_i = \begin{cases} (\mathbf{C}_i - \mathbf{C}_{i-1} - \mathbf{C}_{i-2} - \lfloor \mathbf{R}_i \times 2^{32} \rfloor) \bmod F & \text{if } i \in [3, G]; \\ (\mathbf{C}_2 - \mathbf{C}_1 - \mathbf{T}_G - \lfloor \mathbf{R}_i \times 2^{32} \rfloor) \bmod F & \text{if } i = 2; \\ (\mathbf{C}_1 - \mathbf{T}_G - \mathbf{T}_{G-1} - \lfloor \mathbf{R}_i \times 2^{32} \rfloor) \bmod F & \text{if } i = 1. \end{cases} \quad (9)$$

To show the performance of the 2D-LSCM diffusion, we provide an image example, which is shown in Fig. 6. One can see that the 2D-LSCM diffusion can randomly change pixel values, which is shown in Fig. 6(b). When using the same secret key to do the 2D-LSCM diffusion to two plain-images with only one bit difference, the difference can be spread to all the pixels behind the different pixel, which is shown in Fig. 6(c). After two rounds of diffusion, the change of one pixel can be spread all over the image, which can be seen from Fig. 6(d). Thus, the 2D-LSCM diffusion can achieve good diffusion property.

4. Simulation results and efficiency analysis

This section simulates the proposed LSCM-IEA and analyzes its efficiency. Most of test images in our experiments are selected from the USC-SIPI image dataset[1] (grayscale images and color images) and Brown Univ Large Binary image database[2](binary images).

4.1. Simulation results

An efficient image encryption algorithm must have the ability to encrypt different types of digital images into unrecognized cipher-images. Only with the correct key, one can completely decrypt the cipher-image. Without key or with a wrong key, one can't obtain any useful information about the original image. Fig. 7 shows the encryption procedures of the binary, grayscale and color images. One can observe that all the plain-images have many patterns that make them hard to be processed. However, their cipher-images are all random-like and their pixel values distribute very randomly, which can be seen from Fig. 7(c) and (d). Attackers can't obtain any useful information about the original images from their pixel distributions. Using the same secret key, the decryption process can totally recover the original images, which can be seen in Fig. 7(e).

4.2. Efficiency analysis

The high efficiency of image encryption is required, as a large number of digital images with high resolutions are generated every moment. The proposed LSCM-IEA has low time complexity, as it can achieve the following properties: (1) the used chaotic map is a 2D discrete-time map and has low implementation cost; (2) the 2D-LSCM permutation can shuffle the pixel column and row positions simultaneously, and thus has high shuffling efficiency; (3) four rounds of encryption processes can guarantee a high security level. Table 3 compares the time complexity and encryption time of several advanced image encryption algorithms using images of different sizes. For Xu et al. and Diaconu et al.s' [47] algorithms, the time complexity is reported in their papers and thus we directly refer their results. To provide a fair comparison, we adopt the same principle with Diaconu et al.'s algorithm to calculate the time complexity of our proposed LSCM-IEA and Zhou et al.'s [29] algorithm. The second column of Table 3 lists the time complexity of different algorithms and the results show that our proposed LSCM-IEA has the lowest time complexity. To compare the actual encryption time of these encryption algorithms, we implement these algorithms using Matlab R2015b and use images of different sizes to test their actual encryption time. The experimental environments are as follows: Intel(R) Core(TM) i5-3320M CPU @ 2.6 GHz with 8GB memory, Windows 7 Operation system. One can see that the proposed LSCM-IEA requires the least time when encrypting images with different sizes. This further indicates that it has the higher encryption efficiency than other three algorithms.

5. Security analysis

The security performance is the most important indictor of an image encryption algorithm. This section analyzes the security of the proposed LSCM-IEA in terms of key security, ability of defending differential attack, local Shannon entropy and contrast analysis.

5.1. Key security

The secret key plays an important role in an encryption algorithm. On one hand, the secret key should have proper size to resist the brute-force attack. As mentioned in Section 3.1 that the length of secure key should be bigger than 100 bits. Considering the rapid enhancement of computer computing ability, we set the length of the secret key of LSCM-IEA as 256 bits. On the other hand, the secret key must be very sensitive. If a secret key isn't sensitive, an equivalent secure key can be obtained and this will greatly reduce the actual key space.

[1] http://sipi.usc.edu/database/ .

[2] http://vision.lems.brown.edu/content/available-software-and-databases/ .

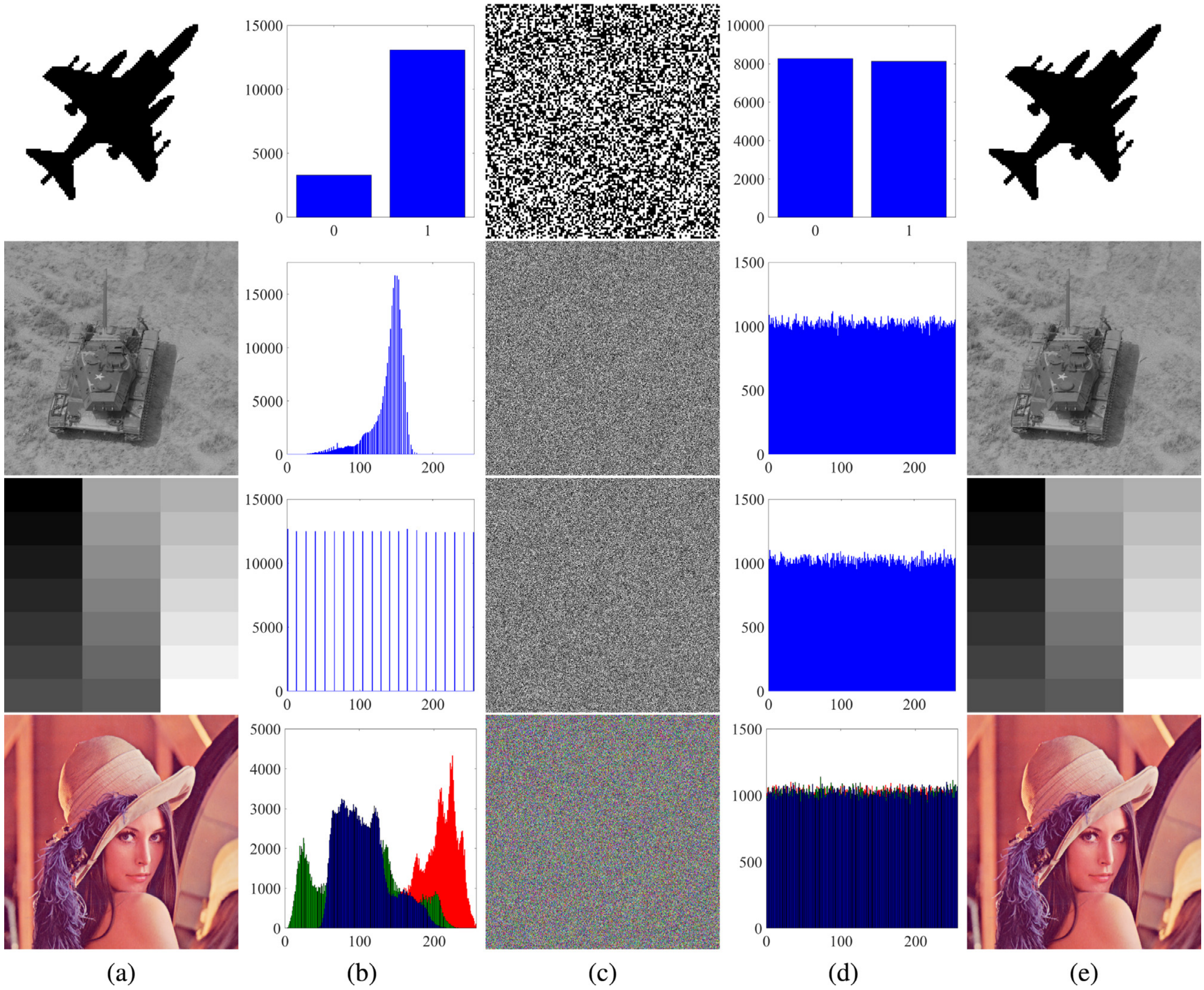


Fig. 7. Simulation results of LSCM-LEA: (a) the binary, 8-bit grayscale, and 24-bit color images; (b) histograms of (a); (c) encrypted results of (a); (d) histograms of (c); (e) decrypted results of (c).

Table 3
Time complexity and encryption time (second) of different image encryption algorithms for images with different sizes.

Image size	Time complexity	128×128	256×256	512×512	1024×1024
Xu's [25]	$O(M\log(8N) + 8N\log M + M + 8N)$	0.0321	0.1484	0.6921	2.8115
Diaconu's [47]	$O(9MN)$	0.0687	0.2637	1.1003	4.3618
Zhou's [29]	$O(8MN)$	0.0814	0.3042	1.2030	4.8264
LSCM-IEA	$O(4(M\log N + M + N))$	0.0196	0.0800	0.4842	2.2848

To visually display the key sensitivity of LSCM-IEA, we first randomly generate a secret key K_1,

$$\begin{aligned} K_1 = {} & EFC796D47FDFFFE9AB7DF3DFFF3CE7AFDEFEFC6977757 \\ & FC9DA69D93F4D76FC7F, \end{aligned}$$

and then change one bit of K_1 to obtain two other keys, K_2 and K_3. Fig. 8 shows the key sensitivity in the encryption process and Fig. 9 demonstrates the key sensitivity in decryption process. One can see that when encrypting a plain-image using two secret keys with only one bit difference, the two obtained cipher-images are completely different (see Fig. 8(d)). Only the correct key can totally recover the original image (see Fig. 9(b)). When decrypting a cipher-image with two slightly different keys, the obtained two decrypted results are random-like (see Fig. 9(c) and (d)), and also totally different (see Fig. 9(e)).

To quantitatively test the key sensitivity, we use the number of bit change rate (NBCR) to calculate the difference of images. For two sequences S_1 and S_2 with the same length, their NBCR can be described as

$$NBCR = \frac{Hm[S_1, S_2]}{L_b} \times 100\%, \tag{10}$$

where L_b is the length of S_1 or S_2 and $Hm[S_1, S_2]$ is to calculate their Hamming distance [48]. If S_1 and S_2 are two statistic-independent data sequences, their NBCR will approach to 50%.

For each of the 256 bits in K_1, we set the experiments as follows. (1) Change the bit to obtain a slightly different key; (2) use the two secret keys to encrypt a same plain-image and calculate the NBCR of the two encrypted results; (3) use the two secret keys to decrypt a same cipher-image and calculate the NBCR of two de-

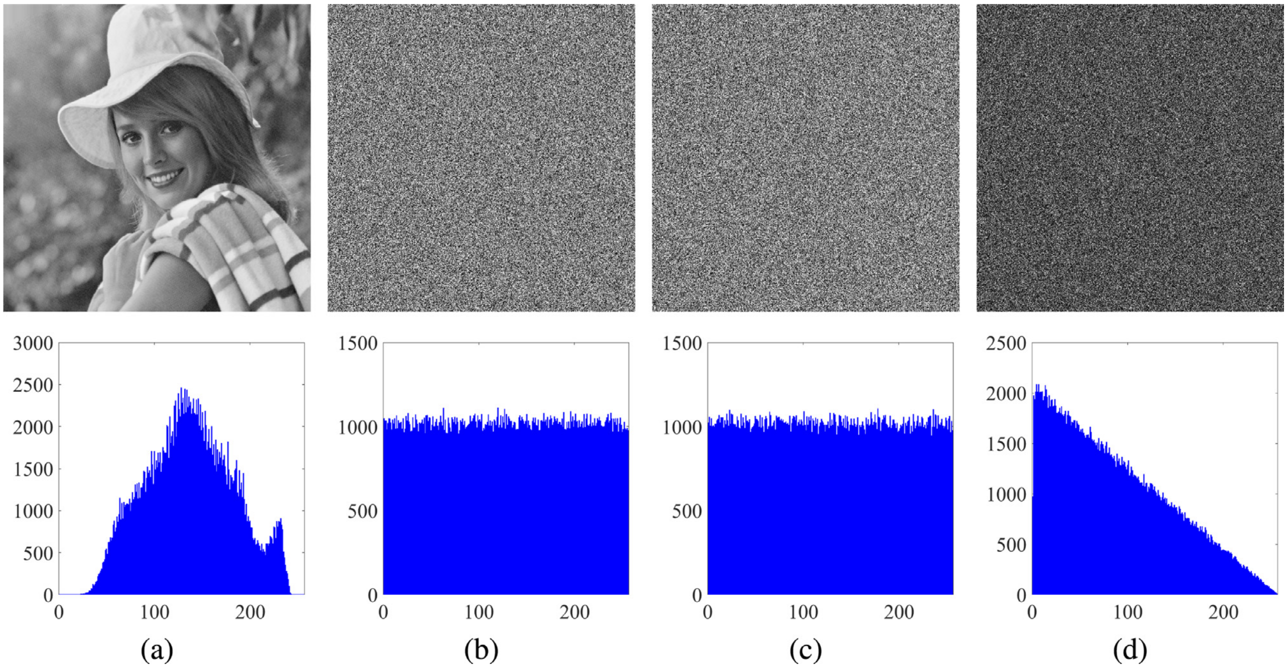


Fig. 8. Key sensitivity analysis in encryption process: (a) plain-image **P**; (b) cipher-image $\mathbf{C}_1 = Enc(\mathbf{P}, K_1)$; (c) cipher-image $\mathbf{C}_2 = Enc(\mathbf{P}, K_2)$; (d) the difference between $\mathbf{C}_1$ and $\mathbf{C}_2$, $|\mathbf{C}_1 - \mathbf{C}_2|$.

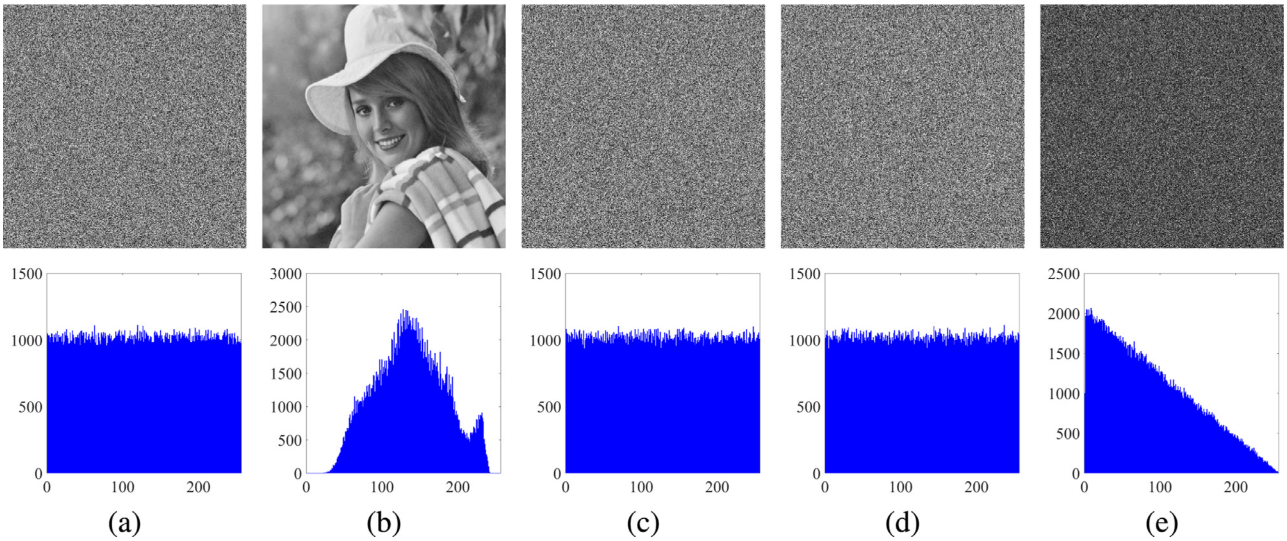


Fig. 9. Key sensitivity analysis in decryption process: (a) cipher-image $\mathbf{C}_1$; (b) decrypted result $\mathbf{D}_1 = Dec(\mathbf{C}_1, K_1)$; (c) decrypted result $\mathbf{D}_2 = Dec(\mathbf{C}_1, K_2)$; (d) decrypted result $\mathbf{D}_3 = Dec(\mathbf{C}_1, K_3)$; (e) the difference between $\mathbf{D}_2$ and $\mathbf{D}_3$, $|\mathbf{D}_2 - \mathbf{D}_3|$.

crypted results. Fig. 10 shows the test results. One can see that when changing any one bit of a randomly generated secret key, the two obtained encrypted results are totally different (see Fig. 10(a)) and the two obtained decrypted results are also independent (see Fig. 10(b)). This means that LSCM-IEA has quite sensitive encryption and decryption keys.

5.2. Ability of defending differential attack

The differential attack is a kind of chosen-plaintext attacks. By tracing how the slight change in plaintexts can affect the ciphertexts, the differential attack tries to find the connections between the plaintexts and ciphertexts, and uses the built connections to recover the ciphertext without secret key. For an image encryption algorithm, its ability of defending differential attack can be tested using the number of pixel changing rate (NPCR) and the unified averaged changed intensity (UACI). For two cipher-images, $\mathbf{C}_1$ and $\mathbf{C}_2$, encrypted from two plain-images with one bit difference, their NPCR and UACI are defined as

$$NPCR(\mathbf{C}_1, \mathbf{C}_2) = \sum_{i=1}^{M}\sum_{j=1}^{N} \frac{\mathbf{W}(i, j)}{H} \times 100\%, \tag{11}$$

and

$$UACI(\mathbf{C}_1, \mathbf{C}_2) = \sum_{i=1}^{M}\sum_{j=1}^{N} \frac{|\mathbf{C}_1(i, j) - \mathbf{C}_2(i, j)|}{H \times Q} \times 100\%, \tag{12}$$

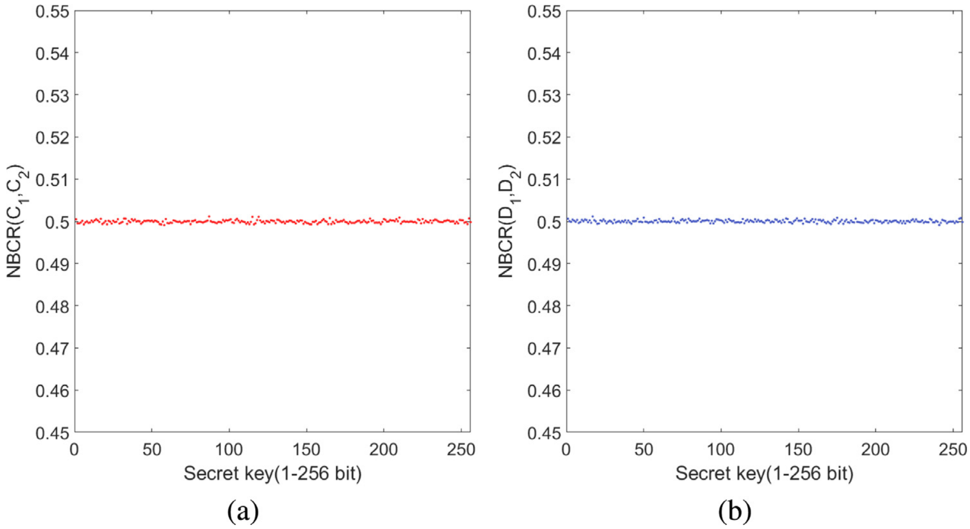


Fig. 10. NBCR in encryption and decryption processes: (a) NBCR between $\mathbf{C}_1$ and $\mathbf{C}_2$, which $\mathbf{C}_1$ and $\mathbf{C}_2$ are two cipher-images encrypted from a same plain-image and two secret keys with only one bit difference; (b) NBCR between $\mathbf{D}_1$ and $\mathbf{D}_2$, which $\mathbf{D}_1$ and $\mathbf{D}_2$ are two decrypted images from cipher-image $\mathbf{C}_1$ and two secret keys with only one bit difference.

respectively, where H is the total number of pixels in an image, Q represents the largest allowed pixel value in the image, and

$$\mathbf{W}(i,j)=\begin{cases}0 & \text{if } \mathbf{C}_1(i,j)=\mathbf{C}_2(i,j);\\ 1 & \text{if } \mathbf{C}_1(i,j)\neq\mathbf{C}_2(i,j).\end{cases} \tag{13}$$

Recently, more strict criterions about the NPCR and UACI were developed in [49]. For a significance level α, a critical NPCR score $\mathcal{N}^*_\alpha$ is obtained by

$$\mathcal{N}^*_\alpha=\frac{Q-\Phi^{-1}(\alpha)\sqrt{Q/H}}{Q+1}. \tag{14}$$

An image encryption scheme can be considered to pass the NPCR if the obtained NPCR is larger than $\mathcal{N}^*_\alpha$. The critical UACI interval $(\mathcal{U}^{*-}_\alpha,\mathcal{U}^{*+}_\alpha)$ can be calculated by

$$\begin{cases}\mathcal{U}^{*-}_\alpha=\mu_{\mathcal{U}}-\Phi^{-1}(\alpha/2)\sigma_{\mathcal{U}};\\ \mathcal{U}^{*+}_\alpha=\mu_{\mathcal{U}}+\Phi^{-1}(\alpha/2)\sigma_{\mathcal{U}},\end{cases} \tag{15}$$

where

$$\mu_{\mathcal{U}}=\frac{Q+2}{3Q+3}, \tag{16}$$

and

$$\sigma^2_{\mathcal{U}}=\frac{(Q+2)(Q^2+2Q+3)}{18(Q+1)^2QH}. \tag{17}$$

If the obtained UACI falls into range $(\mathcal{U}^{*-}_\alpha,\mathcal{U}^{*+}_\alpha)$, the corresponding encryption algorithm is considered to have high security level.

A number of 28 grayscale images in USC-SIPI image database are selected in our experiment. Among these 28 images, six images have size of 256 × 256; eighteen images have size of 512 × 512 and four images have size of 1024 × 1024. According to the discussions in [49], we set the significance level $\alpha=0.05$, then for images of size 256 × 256, $\mathcal{N}^*_\alpha=99.5693\%$ and $(\mathcal{U}^{*-}_\alpha,\mathcal{U}^{*+}_\alpha)=(33.2824\%,33.6447\%)$; for images of size 512 × 512, $\mathcal{N}^*_\alpha=99.5893\%$ and $(\mathcal{U}^{*-}_\alpha,\mathcal{U}^{*+}_\alpha)=(33.3730\%,33.5541\%)$; and for images of size 1024 × 1024, $\mathcal{N}^*_\alpha=99.5994\%$ and $(\mathcal{U}^{*-}_\alpha,\mathcal{U}^{*+}_\alpha)=(33.4183\%,33.5088\%)$. In each test, we randomly change one bit of an image to obtain another image, and then encrypt the two images using a same secret key to get two encrypted results, and calculate the NPCR and UACI scores of the two encrypted results. Fig. 11 plots the NPCR scores of different image encryption algorithms and Fig. 12 shows their UACI scores. The LSCM-IEA can obtain NPCR and UACI scores that are all within the accepted intervals. On the other hand, other image encryption schemes fail to pass some tests. This indicates that the proposed LSCM-IEA can achieve higher ability of defending differential attack than these other encryption algorithms.

5.3. Local Shannon entropy

The pixels of a cipher-image are expected to randomly distribute to resist various security attacks. The local Shannon entropy (LSE) can provide a strict description to the randomness of image pixel [50]. For an image $\mathbf{I}$, randomly select k non-overlapping image blocks $\mathbf{S}_1,\mathbf{S}_2,\ldots,\mathbf{S}_k$ with T_B pixels, the LSE can be defined as

$$\overline{H_{k,T_B}}(\mathbf{I})=\sum_{i=1}^{k}\frac{H(\mathbf{S}_i)}{k}, \tag{18}$$

where $H(\mathbf{S}_i)$ is the Shannon entropy of image block $\mathbf{S}_i$ and can be defined as

$$H(\mathbf{S}_i)-\sum_{l=1}^{L}P(l)\log(P(l)), \tag{19}$$

where L is the total number of pixel values and $P(l)$ is the probability of lth values.

Our experiment also uses the images from USC-SIPI image dataset to do the simulation. According to the recommendation in [50], we set the parameters $(k,T_B)=(30,1936)$ and significance $\alpha=0.05$, then the ideal LSE is 7.902469317 and an image is considered to pass the test if the obtained LSE falls into the interval (7.901901305, 7.903037329). Table 4 lists the LSE scores of cipher-images encrypted by several image encryption schemes. One can see that LSCM-IEA has 20 cipher-images that are within the accepted interval and its pass rate is the highest. This means that the proposed LSCM-IEA can encrypt images into cipher-images with high randomness.

5.4. Contrast analysis

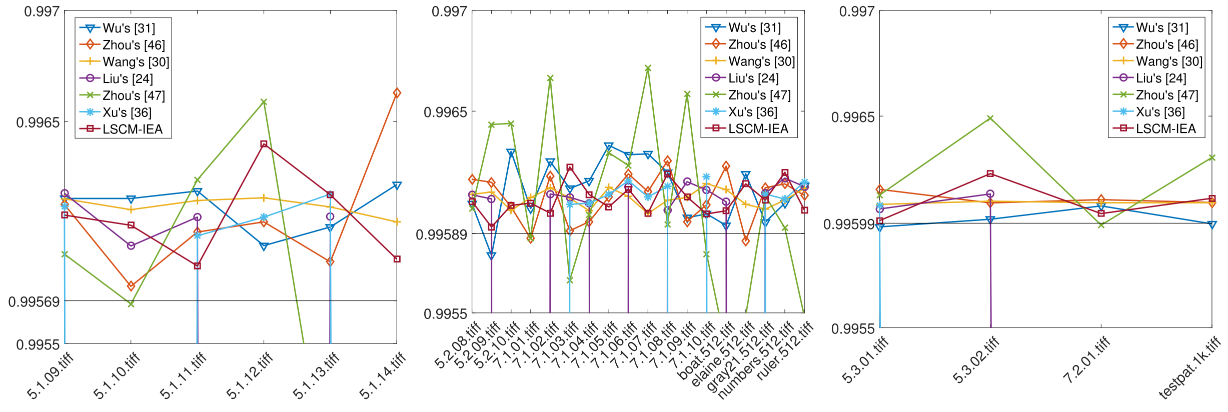
Contrast feature is a kind of statistical texture characteristic and it can reflect the clarity degree of image and the texture of the

Table 4
The LSE scores of cipher-images encrypted by different image encryption schemes.

Images	LSE scores						
	Wu's [51]	Zhou's [26]	Wang's [4]	Liu's [5]	Zhou's [29]	Xu's [25]	LSCM-IEA
5.1.09	7.903223	7.903595	**7.902682**	**7.901914**	**7.903032**	7.903543	**7.902281**
5.1.10	7.903087	**7.902314**	7.903397	7.900288	**7.901979**	7.903137	**7.902198**
5.1.11	7.906766	7.903901	7.904131	7.900441	7.901630	7.905119	7.899982
5.1.12	7.903390	7.900834	**7.902789**	7.900276	7.904191	7.904896	**7.902827**
5.1.13	7.899016	**7.902525**	7.903841	7.904302	7.901417	**7.901933**	**7.902281**
5.1.14	7.901087	7.903649	7.901668	7.899342	7.905588	7.901559	7.903117
5.2.08	7.900827	7.901765	7.903854	**7.902088**	7.904109	**7.902714**	**7.902304**
5.2.09	7.903732	7.900433	7.905012	7.905057	7.899799	7.898275	**7.902022**
5.2.10	7.901648	**7.901966**	**7.902882**	7.900481	7.900969	7.904609	7.906701
7.1.01	7.898618	7.901058	**7.902966**	7.901860	7.900260	7.903507	**7.902191**
7.1.02	7.904654	7.903413	7.906349	7.901194	7.900644	**7.902496**	**7.902047**
7.1.03	7.901633	7.904178	7.900470	7.901148	7.901404	7.899554	**7.902584**
7.1.04	7.905116	**7.902179**	7.900964	7.903347	**7.902802**	7.900657	**7.901913**
7.1.05	**7.902414**	**7.902124**	**7.901991**	7.901434	7.900019	**7.902717**	**7.902392**
7.1.06	7.901472	7.904908	**7.902182**	7.903007	7.904535	7.903967	**7.902565**
7.1.07	**7.902247**	7.903210	7.900828	7.901576	7.903704	**7.902364**	7.904015
7.1.08	7.903583	7.904767	7.901676	**7.901945**	7.901698	**7.902537**	7.901096
7.1.09	7.905126	**7.902820**	7.901032	7.903082	7.901889	**7.902012**	**7.902933**
7.1.10	7.904126	7.904401	7.903549	**7.902345**	7.901574	7.900796	**7.902534**
boat.512	**7.902755**	7.900889	7.901836	**7.902716**	7.903704	7.899492	7.901782
elaine.512	**7.902115**	**7.902934**	**7.902525**	7.904935	7.901016	**7.902890**	**7.902569**
gray21.512	7.904832	**7.902972**	7.901614	7.900796	7.901009	7.901349	**7.902593**
numbers.512	7.901345	7.900308	7.901442	**7.901988**	7.901481	7.904014	**7.902295**
ruler.512	**7.902244**	7.900604	7.915057	7.901759	7.900907	**7.902885**	7.904102
5.3.01	7.899751	7.904291	**7.902397**	7.899133	7.906108	7.900850	**7.902119**
5.3.02	7.903496	7.903330	**7.902727**	7.903761	7.903088	7.904658	**7.902658**
7.2.01	7.903118	**7.902309**	7.897698	7.903574	**7.901969**	**7.902648**	**7.902529**
testpat.1k	7.901350	7.903963	7.905429	7.900457	7.900552	**7.902370**	7.904472
Mean	**7.902599**	**7.902701**	**7.902964**	**7.901937**	**7.902181**	**7.902412**	**7.902611**
Std	0.0019	0.0014	0.0029	0.0016	0.0017	0.0017	0.0012
Pass/All	5/28	9/28	9/28	6/28	4/28	11/28	20/28

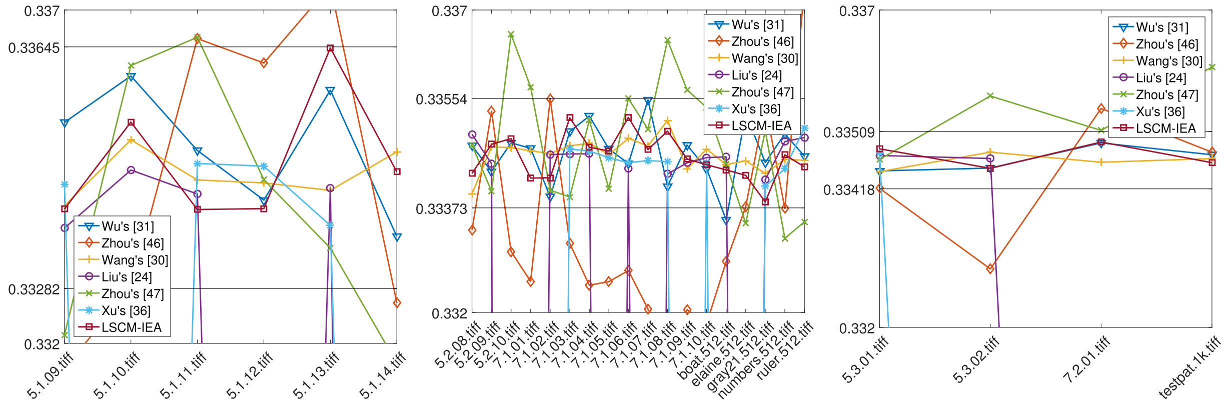
Table 5
The average contrast feature scores of plain-images and cipher-images encrypted by different encryption schemes.

Image names	Plain-images	Cipher-images						
		Wu's [51]	Zhou's [26]	Wang's [4]	Liu's [5]	Zhou's [29]	Xu's [25]	LSCM-IEA
5.1.09	146.58	10,990.18	10,825.09	10,893.38	10,931.11	10,895.04	10,864.31	10,918.65
5.1.10	643.59	10,928.06	10,759.94	10,899.89	10,903.62	10,932.14	10,921.68	10,971.56
5.1.11	162.52	10,970.74	10,951.97	11,208.19	11,013.58	10,910.65	10,879.77	10,892.87
5.1.12	319.59	10,947.53	10,984.96	10,878.80	10,907.35	10,916.19	10,887.21	10,906.51
5.1.13	2156.50	10,970.09	11,063.32	10,673.72	10,968.20	10,961.14	10,915.41	10,918.48
5.1.14	373.42	10,915.49	10,822.34	10,882.79	10,957.51	10,948.08	10,888.47	10,935.12
5.2.08	365.48	10,924.60	10,825.03	10,950.11	10,941.72	10,920.95	10,958.19	10,930.46
5.2.09	495.37	10,874.32	10,913.67	10,620.65	10,934.30	10,940.19	10,923.55	10,912.54
5.2.10	500.50	10,883.53	10,862.59	10,927.80	10,914.24	10,921.69	10,912.08	10,894.26
7.1.01	112.71	10,939.47	10,779.52	10,828.41	10,901.89	10,873.45	10,904.39	10,869.71
7.1.02	66.00	10,890.05	10,967.52	11,053.26	10,920.88	10,891.17	10,924.32	10,925.80
7.1.03	111.90	10,930.22	10,831.80	10,799.62	10,929.77	10,955.71	10,936.30	10,934.25
7.1.04	87.06	10,953.86	10,794.12	10,783.33	10,906.88	10,938.24	10,925.36	10,927.72
7.1.05	223.11	10,920.42	10,779.67	10,727.20	10,956.37	10,923.36	10,893.59	10,944.33
7.1.06	214.00	10,909.55	10,814.31	10,983.37	10,927.87	10,918.13	10,902.22	10,908.84
7.1.07	169.91	10,938.58	10,778.70	10,816.53	10,934.56	10,889.13	10,881.50	10,939.20
7.1.08	80.87	10,925.85	10,748.99	10,689.40	10,960.78	10,916.22	10,873.91	10,879.07
7.1.09	173.58	10,908.33	10,811.99	10,800.38	10,959.07	10,952.96	10,870.78	10,901.27
7.1.10	82.16	10,931.56	10,733.89	10,756.54	10,944.49	10,911.70	10,914.01	10,958.93
boat.512	264.54	10,916.86	10,815.90	10,808.99	10,908.80	10,938.30	10,920.49	10,908.76
elaine.512	118.06	10,952.38	10,849.81	10,873.72	10,951.17	10,938.32	10,940.97	10,913.04
gray21.512	32.12	10,918.52	10,920.27	10,780.19	10,946.62	10,943.65	10,890.10	10,932.86
numbers.512	2412.02	10,938.82	10,878.45	10,842.31	10,928.22	10,912.27	10,914.78	10,922.89
ruler.512	9887.66	10,969.83	11,132.50	10,964.11	10,918.64	10,917.91	10,918.19	10,915.47
5.3.01	180.32	10,910.96	10,886.72	10,893.26	10,919.96	10,918.70	10,940.02	10,931.70
5.3.02	294.80	10,931.86	10,818.72	10,765.51	10,926.88	10,937.98	10,928.71	10,919.60
7.2.01	65.67	10,933.63	11,005.99	10,888.40	10,925.96	10,891.52	10,929.63	10,911.24
testpat.1k	3004.42	10,928.63	10,906.06	10,785.69	10,922.59	10,917.30	10,916.52	10,908.33
Mean		10,930.50	10,866.56	10,849.13	10,934.39	10,922.57	10,909.87	10,919.05
Std		26.4372	97.3630	120.0099	24.2624	21.7385	23.6144	21.7588



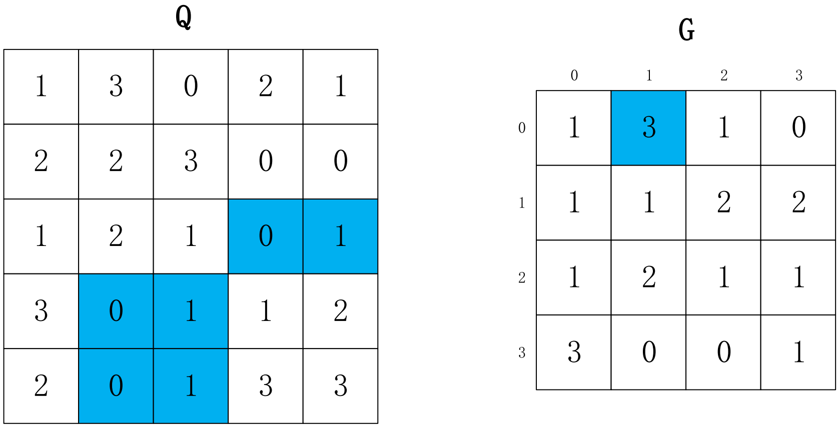
(a) (b) (c)

Fig. 11. NPCR scores of several image encryption schemes using different size of images: (a) images of size 256 × 256; (b) images of size 512 × 512; (c) images of size 1024 × 1024.



(a) (b) (c)

Fig. 12. UACI scores of several image encryption schemes using different size of images: (a) images of size 256 × 256; (b) images of size 512 × 512; (c) images of size 1024 × 1024.



Q

1	3	0	2	1
2	2	3	0	0
1	2	1	0	1
3	0	1	1	2
2	0	1	3	3

G

	0	1	2	3
0	1	3	1	0
1	1	1	2	2
2	1	2	1	1
3	3	0	0	1

Fig. 13. An example of generating the gray level co-occurrence matrix.

groove depth [52]. It can be used to measure pixel distribution and pixel local variation of an image matrix and its mathematical definition is shown as

$$C = \sum_{i,j} |i - j|^2 \mathbf{G}'(i, j), \tag{20}$$

where $\mathbf{G}'$ is gray level co-occurrence matrix, which indicates the probability of fixed patterns with a predefined distance and direction. Fig. 13 shows an example of generating a unnormalized gray level co-occurrence matrix **G** from an image **Q** with 1 distance and horizontal direction. One can see that the size of **G** is 4 × 4 as the grayscale level of **Q** is 4. $\mathbf{G}(0, 1) = 3$ as the number of pattern (0,1) (see the blue cells) in horizontal direction is 3. By this way, we can obtain each value of **G**, which can be seen from the figure. As the total number of patterns with distance 1 and horizontal direction in **Q** is 20, we can obtain the gray level co-occurrence matrix, namely $\mathbf{G}' = \mathbf{G}/20$.

In our experiments, the test images are all 8-bit grayscale images, and thus we can obtain the size of $\mathbf{G}'$ as 256×256. For an ideally random image, each of its patterns is expected to be the same, namely $\mathbf{G}'(i, j) = 1/(256 \times 256)$, where $i \in [0, 255]$ and $j \in [0, 255]$. Thus, the contrast feature of an ideally random image is $C_{expected} = \sum_{i=1}^{256}\sum_{j=1}^{256} |i-j|^2/(256 \times 256) = 10,922.50$. In each experiment, we first calculate four gray level co-occurrence matrices from four directions 0°, 45°, 90°, 135° with 1 distance, and then obtain four feature contrast values and get their average score. Table 5 lists the average feature contrast scores of plain-images and their cipher-images encrypted by different encryption algorithms. One can see that Zhou's [29] algorithm can achieve the mean value score that is closest to the expected value and our proposed LSCM-IMA can achieve the second-best performance. Besides, LSCM-IMA can obtain a pretty small standard deviation.

6. Conclusion

In this paper, we presented a new chaotic map called 2D-LSCM. It is derived from the existing Logistic and Sine maps. First, couple the outputs of the Logistic and Sine maps using the sine transform and then extend the phase plane from 1D to 2D to enhance the complexity. The chaos performance of 2D-LSCM was analyzed using trajectory, Lyapunov exponent, Kolmogorov entropy and dynamical degradation. The analysis results demonstrate that it has better chaos performance than several newly developed 2D chaotic maps, and is suitable for designing encryption algorithms. To show the applications of 2D-LSCM, we further designed a 2D-LSCM-based image encryption algorithm (LSCM-IEA). It has two main components, the 2D-LSCM permutation and 2D-LSCM diffusion. The former can fast shuffle pixel row and column positions simultaneously to achieve confusion property, and the latter is able to spread few changes of plain-image to the whole cipher-image to obtain diffusion property. Simulation results show that LSCM-IEA can encrypt different types of images into unrecognized cipher-images with high efficiency. We have also analyzed the security of LSCM-IEA in terms of key security, ability of defending differential attack, local Shannon entropy and contrast analysis. The analysis results show that LSCM-IEA has a high security level and can outperform some advanced image encryption algorithms. As the proposed LSCM-IEA has high efficiency and security level, our future work will investigate its application in video encryption.

Acknowledgements

This work is financially supported by National Natural Science Foundation of China under grant nos. 61701137, 11371004 and 61672195, National Science and Technology Major Project under grant nos. 2016YFB0800804 and 2017YFB0803002 and Shenzhen Science and Technology Plan under grant nos. JCYJ20170307150704051, JCYJ20160318094336513, JCYJ20160318094101317 and KQCX20150326141251370.

References

[1] M. Murillo-Escobar, C. Cruz-Hernndez, F. Abundiz-Prez, R. Lpez-Gutirrez, O.A.D. Campo, A RGB image encryption algorithm based on total plain image characteristics and chaos, Signal Process. 109 (2015) 119–131.

[2] X.W. Li, I.K. Lee, Robust copyright protection using multiple ownership watermarks, Opt. Express 23 (3) (2015) 3035–3046.

[3] L.Y. Zhang, Y. Liu, F. Pareschi, Y. Zhang, K.W. Wong, R. Rovatti, G. Setti, On the security of a class of diffusion mechanisms for image encryption, IEEE Trans. Cybern. (2017), doi:10.1109/TCYB.2017.2682561.

[4] X. Wang, Q. Wang, Y. Zhang, A fast image algorithm based on rows and columns switch, Nonlinear Dyn. 79 (2) (2015) 1141–1149.

[5] W. Liu, K. Sun, C. Zhu, A fast image encryption algorithm based on chaotic map, Opt. Lasers Eng. 84 (2016) 26–36.

[6] G. Ye, X. Huang, L.Y. Zhang, Z.X. Wang, A self-cited pixel summation based image encryption algorithm, Chin. Phys. B 26 (1) (2017) 010501.

[7] X. Li, D. Xiao, Q.H. Wang, Error-free holographic frames encryption with ca pixel-permutation encoding algorithm, Opt. Lasers Eng. 100 (2018) 200–207.

[8] Z. Hua, S. Yi, Y. Zhou, Medical image encryption using high-speed scrambling and pixel adaptive diffusion, Signal Process. 144 (2018) 134–144.

[9] FIPS PUB 46, Data encryption standard (DES), 1999.

[10] FIPS PUB 197, Advanced encryption standard (AES), 2001.

[11] Y. Zhang, D. Xiao, An image encryption scheme based on rotation matrix bit-level permutation and block diffusion, Commun. Nonlinear Sci. Numer. Simul. 19 (1) (2014) 74–82.

[12] Y. Zhang, D. Xiao, Y. Shu, J. Li, A novel image encryption scheme based on a linear hyperbolic chaotic system of partial differential equations, Signal Process. Image Commun. 28 (3) (2013) 292–300.

[13] C. Fu, J. Chen, H. Zou, W. Meng, Y. Zhan, Y. Yu, A chaos-based digital image encryption scheme with an improved diffusion strategy, Opt. Express 20 (3) (2012) 2363–2378.

[14] G. Ye, X. Huang, Spatial image encryption algorithm based on chaotic map and pixel frequency, Sci. China-Inf. Sci. 61 (5) (2018) 058104.

[15] Y.Q. Zhang, X.Y. Wang, A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice, Inf. Sci. 273 (2014) 329–351.

[16] Y.Q. Zhang, X.Y. Wang, A new image encryption algorithm based on non-adjacent coupled map lattices, Appl. Soft Comput. 26 (2015) 10–20.

[17] R. Enayatifar, H.J. Sadaei, A.H. Abdullah, M. Lee, I.F. Isnin, A novel chaotic based image encryption using a hybrid model of deoxyribonucleic acid and cellular automata, Opt. Lasers Eng. 71 (2015) 33–41.

[18] J. Chen, Z.L. Zhu, L.B. Zhang, Y. Zhang, B.Q. Yang, Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption, Signal Process. 142 (2018) 340–353.

[19] D. Jiang, Y. Chen, X. Gu, L. Xie, L. Chen, Efficient and universal quantum key distribution based on chaos and middleware, Int. J. Mod. Phys. B 31 (2) (2017) 1650264.

[20] N. Zhou, Y. Hu, L. Gong, G. Li, Quantum image encryption scheme with iterative generalized Arnold transforms and quantum image cycle shift operations, Quantum Inf. Process. 16 (6) (2017) 164.

[21] N. Zhou, S. Pan, S. Cheng, Z. Zhou, Image compression–encryption scheme based on hyper-chaotic system and 2D compressive sensing, Opt. Laser Technol. 82 (2016) 121–133.

[22] X. Chai, Z. Gan, Y. Chen, Y. Zhang, A visually secure image encryption scheme based on compressive sensing, Signal Process. 134 (2017) 35–51.

[23] Y. Zhou, K. Panetta, S. Agaian, C.L.P. Chen, Image encryption using P-Fibonacci transform and decomposition, Opt. Commun. 285 (5) (2012) 594–608.

[24] Z. Hua, Y. Zhou, Design of image cipher using block-based scrambling and image filtering, Inf. Sci. 396 (2017) 97–113.

[25] L. Xu, Z. Li, J. Li, W. Hua, A novel bit-level image encryption algorithm based on chaotic maps, Opt. Lasers Eng. 78 (2016) 17–25.

[26] Y. Zhou, L. Bao, C.L.P. Chen, Image encryption using a new parametric switching chaotic system, Signal Process. 93 (11) (2013) 3039–3052.

[27] L. Gong, X. Liu, F. Zheng, N. Zhou, Flexible multiple-image encryption algorithm based on log-polar transform and double random phase encoding technique, J. Mod. Opt. 60 (13) (2013) 1074–1082.

[28] D. Xiao, X. Liao, S. Deng, One-way Hash function construction based on the chaotic map with changeable-parameter, Chaos Solitons Fractals 24 (1) (2005) 65–71.

[29] Y. Zhou, L. Bao, C.L.P. Chen, A new 1D chaotic system for image encryption, Signal Process. 97 (2014) 172–182.

[30] C. Pak, L. Huang, A new color image encryption using combination of the 1D chaotic map, Signal Process. 138 (2017) 129–137.

[31] C. Li, Y. Liu, T. Xie, M.Z.Q. Chen, Breaking a novel image encryption scheme based on improved hyperchaotic sequences, Nonlinear Dyn. 73 (3) (2013) 2083–2089. https://doi.org/10.1007/s11071-013-0924-6.

[32] C. Li, Cracking a hierarchical chaotic image encryption algorithm based on permutation, Signal Process. 118 (2016) 203–210. https://doi.org/10.1016/j.sigpro.2015.07.008.

[33] E.Y. Xie, C. Li, S. Yu, J. Lü, On the cryptanalysis of Fridrich's chaotic image encryption scheme, Signal Process. 132 (2017) 150–154.

[34] M. Liu, S. Zhang, Z. Fan, M. Qiu, H_∞ state estimation for discrete-time chaotic systems based on a unified model, IEEE Trans. Syst. Man Cybern. Part B 42 (4) (2012) 1053–1063.

[35] L. Lin, M. Shen, H.C. So, C. Chang, Convergence analysis for initial condition estimation in coupled map lattice systems, IEEE Trans. Signal Process. 60 (8) (2012) 4426–4432.

[36] A.N. Srivastava, S. Das, Detection and prognostics on low-dimensional systems, IEEE Trans. Syst. Man Cybern. Part C 39 (1) (2009) 44–54.

[37] S. Li, X. Mou, Y. Cai, Z. Ji, J. Zhang, On the security of a chaotic encryption scheme: problems with computerized chaos in finite computing precision, Comput. Phys. Commun. 153 (1) (2003) 52–58.

[38] D. Arroyo, G. Alvarez, S. Li, C. Li, J. Nunez, Cryptanalysis of a discrete-time synchronous chaotic encryption system, Phys. Lett. A 372 (7) (2008) 1034–1039.

[39] R.M. May, Simple mathematical models with very complicated dynamics, Nature 261 (5560) (1976) 261–5560.

[40] Z. Hua, Y. Zhou, C.M. Pun, C.L.P. Chen, 2D sine logistic modulation map for image encryption, Inf. Sci. 297 (2015) 80–94.

[41] K. Briggs, An improved method for estimating Liapunov exponents of chaotic time series, Phys. Lett. A 151 (1–2) (1990) 27–32.

[42] P. Grassberger, I. Procaccia, Estimation of the Kolmogorov entropy from a chaotic signal, Phys.Rev. A 28 (4) (1983) 2591.

[43] Z. Hua, Y. Zhou, Image encryption using 2D logistic-adjusted-sine map, Inf. Sci. 339 (2016) 237–253.

[44] S. Li, G. Chen, X. Mou, On the dynamical degradation of digital piecewise linear chaotic maps, Int. J. Bifurcation Chaos 15 (10) (2005) 3119–3151.

[45] L.E. Bassham III, SP 800-22 Rev. 1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications, Tech. rep. sp 800-22, National Institute of Standards & Technology, 2010.

[46] G. Alvarez, S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, Int. J. Bifurcation Chaos 16 (08) (2006) 2129–2151.

[47] A.V. Diaconu, Circular inter-intra pixels bit-level permutation and chaos-based image encryption, Inf. Sci. 355 (2016) 314–327.

[48] J.C.H. Castro, J.M. Sierra, A. Seznec, A. Izquierdo, A. Ribagorda, The strict avalanche criterion randomness test, Math. Comput. Simul. 68 (1) (2005) 1–7.

[49] Y. Wu, J.P. Noonan, S. Agaian, NPCR and UACI randomness tests for image encryption, cyber journals: multidisciplinary journals in science and technology, J. Sel. Areas Telecommun. (JSAT) (2011) 31–38.

[50] Y. Wu, Y. Zhou, G. Saveriades, S. Agaian, J.P. Noonan, P. Natarajan, Local Shannon entropy measure with statistical tests for image randomness, Inf. Sci. 222 (2013) 323–342.

[51] 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), pp. 23–27.

[52] R.M. Haralick, K. Shanmugam, Textural features for image classification, IEEE Trans. Syst. Man Cybern. 6 (1973) 610–621.