Noname manuscript No.
(will be inserted by the editor)

Color image encryption using orthogonal Latin squares and a new 2D chaotic system

Zhongyun Hua · Zhihua Zhu · Yongyong Chen · Yuanman Li

Received: date / Accepted: date

Abstract Recently, many image encryption schemes have been developed using Latin squares. When encrypting a color image, these algorithms treat the color image as three greyscale images and encrypt these greyscale images one by one using the Latin squares. Obviously, these algorithms don't sufficiently consider the inner connections between the color images and Latin squares, and thus result in many redundant operations and low efficiency. To address this issue, in this paper, we propose a new color image encryption algorithm (CIEA) that sufficiently considers the properties of the color image and Latin square. First, we propose a two-dimensional chaotic system called 2D-LSM to address the weaknesses of existing chaotic systems. Then, we design a new CIEA using orthogonal Latin squares and 2D-LSM. The proposed CIEA can make full use of the inherent connections of the orthogonal Latin squares and color image, and executes the encryption process in the pixel level. Simulation and security analysis results show that the proposed CIEA has a high level of security and can outperform some representative image encryption algorithms.

Keywords Image security · chaotic system · chaos · color image encryption · orthogonal Latin squares

Zhongyun Hua(✉) · Zhihua Zhu · Yongyong Chen
School of Computer Science and Technology, Harbin Institute of Technology Shenzhen, Shenzhen 518055, China
E-mail: huazyum@gmail.com; huazhongyun@hit.edu.cn

Yuanman Li
College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China

1 Introduction

The fast development of digital technology makes daily life more and more convenient. The information security becomes important, since it is very easy to spread digital information through different networks. Because digital image has data redundancy and can carry much visual information, it is a most widely used data format. The illegal accesses of the secret image may cause serious information security accidents. Thus, it is quite important to prevent the contents of digital images from being unauthorizedly accessed. To keep the security of digital images, researchers proposed many technologies including the data hiding [40], water marking [36] and image encryption [15,44]. Among all these technologies, the image encryption is a straightforward and effective one by transforming a meaningful image as an unrecognizable image. Only using the correct key, the original image's information can be recovered [38,57].

One method of encrypting is encrypting a digital image as a bit stream using some data encryption algorithms [34]. However, different from the bit steam, an image has many inner properties such as high data redundancy and adjacent pixel correlations. Encrypting an image as a bit stream cannot make full use of these properties and thus cause many shortcomings such as low encryption efficiency [58,24]. Therefore, designing image encryption algorithms considering the properties of images is necessary. Recently, many encryption algorithms for digital images have been proposed using different techniques [54,55,51] including DNA coding [41], frequency transformation [27], compressive sensing [28, 7] and chaos theory [20,23,17]. For example, the authors in [48] proposed an image encryption algorithm using the quaternion technique. It has many advantages and thus can achieve a high security level. Besides, the

authors in [7] simultaneously performed the encryption and compression to the digital images, which provides a novel strategy for image encryption technique.

A Latin square/cube is a special 2D/3D matrix that each element exists once in each column/row. With this significant property, the Latin squares/cubes are widely used to design the algorithm structures of image encryptions [43]. However, when encrypting a color image, these algorithms either treat a color image as three greyscale images and encrypt these greyscale images one by one using Latin squares, or decompose a greyscale image to be a bit cube and then encrypt the bit cube using the Latin cubes [46]. Treating a color image as three greyscale images or treating a greyscale image as bit cube cannot sufficiently consider the inner connections between the images and Latin squares/cubes, and thus results in many redundant operations and low efficiency [49]. When designing an image encryption algorithm, the chaos theory is widely used to distribute the secret key and generate random numbers for encryption processes [16]. This is because a chaotic system owns many inner characteristics such as the random-like behavior, unpredictability and initial sensitivity [47,1]. These characteristics are similar with the basic concepts of image encryption [15,21]. However, existing chaotic systems used in image encryption have many disadvantages. First, their chaotic ranges are narrow and discontinuous [14,33]. When simulating a chaotic system in digital platforms, the discontinuous chaotic ranges may result in the chaos degradation because of the precision truncation [5]. Besides, the structures of existing chaotic systems are very simple that make their behaviors can be easily predicted [8]. When the behavior of a chaotic system is predicted, the practical applications using the system become ineffective [4].

From the discussions above, we get that many existing image encryption algorithms using chaos and Latin squares/cubes have obvious weaknesses in the algorithm structures and used chaotic systems. To address these issues, this paper presents a new color image encryption algorithm (CIEA) using three-dimensional (3D) orthogonal Latin squares and a new two-dimensional (2D) chaotic system. The proposed CIEA mainly contains the point-to-point permutation, cross-plane diffusion and finite field multiplication. The point-to-point permutation can simultaneously shuffle the pixel row positions and column positions within all the three color planes using 3D orthogonal Latin squares, the cross-plane diffusion processes pixels of all the three color planes in a random order, and the finite field multiplication transforms image pixels in finite field to further increase the security. The novelty and contributions of this paper are summarized as follows.

- We design a novel 2D chaotic system called 2D-LSM. Performance analysis shows that it has continuous and wide chaotic range and better performance than recently developed 2D chaotic maps.
- Using the designed 2D-LSM, we devise a new CIEA called LSM-CIEA that can totally consider the properties of Latin squares and color image.
- Simulation and security evaluation results demonstrate that the LSM-CIEA can make full use of the inner connections between the orthogonal Latin squares and color image, and thus has a high security level and efficiency. The comparison results indicate that it shows better performance than several state-of-the-art encryption algorithms.

The remainder of this paper is organized as follows. Section 2 reviews some representative works about image encryption algorithms using the Latin squares/cubes and chaotic systems, and introduces the generation of 3D orthogonal Latin squares. Section 3 introduces the proposed 2D-LSM and analyzes its performance. Section 4 presents the proposed LSM-CIEA and Section 5 simulates the LSM-CIEA and evaluates its security level. Section 6 gives a conclusion of this paper.

2 Related Works

This section reviews some image encryption algorithms using the Latin squares/cubes and chaotic systems, and analyzes their properties. In addition, we present the generation of 3D orthogonal Latin squares.

2.1 Encryption Algorithms Using Latin Squares/Cubes

First, we give a detailed description about the Lain square and Latin cube. Then, we discuss the existing encryption algorithms using the Lain square/cube.

2.1.1 Latin Square and Latin Cube

A Latin square is an $n \times n$ square matrix, where every element occurs only once in every row and every column [10]. Fig. 1 shows four different Latin squares with different symbol sets, and they can visually demonstrate the properties of the Latin square. A Latin cube is a 3D form of Latin square. Similar to the Latin square, a Latin cube of size $n \times n \times n$ has n different elements and every one occurs once in every axis-aligned plane. Fig. 2 shows a Latin cube of size $4 \times 4 \times 4$, which is composed of four elements $\{a, b, c, d\}$. As can be seen, the Latin cube consists of four Latin squares with size

4×4 and each element of $\{a, b, c, d\}$ appears only once in every row, column, and vertical, respectively.

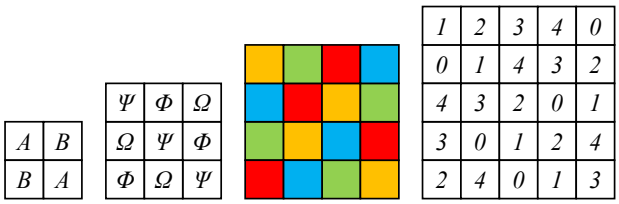


Fig. 1 Examples of Latin squares.

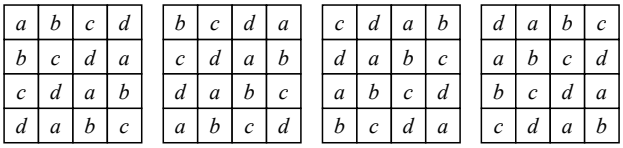


Fig. 2 Example of a Latin cube with size $4 \times 4 \times 4$.

2.1.2 Encryption Algorithms Using Latin Squares/Cubes

Because of the unique characteristics, the Latin squares/cubes have been used in many image encryption algorithms. These algorithms contain two categories. The first category applies the Latin squares to encrypt a greyscale image in 2D space [19]. For example, in [29], the authors proposed an encryption algorithm that uses a Latin square to perform the substitution process. In [26], the authors presented an image encryption algorithm by combing the Latin square and cellular neural network. This category of encryption algorithms can be only directly applied to greyscale images with square size. When encrypting a color image, one should first divide it to be three color planes, and then separately encrypt each of the color planes as a greyscale image, and finally combine the three results to obtain the cipher-image. This strategy for color images is shown as method 1 in Fig. 3. However, the three color planes of a color image have many inner properties. Treating them as three greyscale images cannot consider the properties among color planes, and thus may result in low efficiency.

The second category of encryption algorithms first decomposes an image into a bit cube and then encrypts the bit cube using Latin cubes, as shown in method 2 of Fig. 3. However, these encryption algorithms cause many shortcomings. First, The encryption efficiency is low. An digital image is composed of pixels. Encrypting

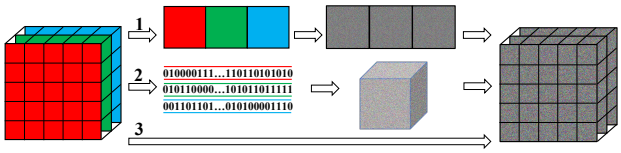


Fig. 3 The traditional encryption processes (methods 1 and 2) and the desired encryption process (method 3) for encrypting a color image.

image pixels in bit level can cause excessive and complicated operations and thus increases the computational cost [52,59,25]. Secondly, these encryption algorithms are only suitable for images with a specific size, and this causes inconvenient in practical applications [49]. For example, the authors in [46] first decompose a greyscale image with size 512×512 into a 3D bit matrix with size $128 \times 128 \times 128$, and then encrypt the 3D bit matrix using 3D Latin cubes. Obviously, this encryption strategy is only suitable for the images whose image size can be decomposed to be 3D bit matrix of size $n \times n \times n$. In addition, when encrypting a color image, these encryption algorithms should also encrypt each of the three color planes individually and then combine the three results to obtain the final cipher-image. Table 1 shows some representative image encryption algorithms using the Latin squares/cubes and their limitations. Thus, it is important to design new encryption structures that fully consider the inner properties of the color image and Latin cube/square. A desired encryption structure is shown as method 3 in Fig. 3. It shows that a color image with any size can be directly encrypted. It can be seen that the three color planes of an color image can be directly encrypted without any preprocessing.

2.2 Chaotic Systems

Chaos theory is a popular used technique for designing image encryption algorithms, due to its properties of initial sensitivity, unpredictability and random-like behavior [6,37]. When being applied in image encryption, the chaotic systems are to generate random numbers or their structures are used to distribute image pixels [50]. Based on the dimension number of the phase space, chaotic systems contain the one-dimensional (1D) and multi-dimensional (MD) ones. The famous 1D chaotic systems have the Logistic, Sine and Tent maps [9]. A 1D chaotic system usually owns a simple structure, which makes its chaotic signal easily to predict [22]. When used in image encryption, this property can lead to the successful prediction of encryption processes, and further causes security issues [11]. Examples of MD chaotic systems include the 3D-PLM [32] and NL4DLM [35]. A

Table 1 Some representative image encryption algorithms using the Latin square/cubes and their properties.

Encryption algorithms	Encryption strategy	Properties
Ref. [46]	Latin cube; bit level	Encrypt images only with specific size
Ref. [43]	Latin square; bit level	Encrypt images only with specific size
Ref. [29]	Latin square; chaos theory	Low quality; only greyscale images
Ref. [18]	Latin square; pixel level	Can't resist chosen plaintext attack
Ref. [49]	Latin cube; bit level	Can't resist chosen plaintext attack
Ref. [25]	Latin cube; DNA encoding	High complexity; low efficiency
Ref. [53]	3D orthogonal Latin squares	Encrypt images only with specific size
Ref. [3]	Latin square; chaos theory	High complexity; only greyscale images

MD chaotic system usually has a complex structure, which makes its chaotic signal hard to be predicted. This can enhance the security level when being used in image encryption. However, high dimension also lead to high implementation cost and low efficiency.

Since 2D chaotic systems can own complex chaotic behaviors and relatively low implementation cost, they can balance the efficiency and performance. Thus, they are widely used to design image encryption algorithms. Recently, some image encryption algorithms were developed using different 2D chaotic systems, including the 2D-LASM [15], 2D-SLMM [16], 2D-LSCM [12] and 2D-LSMCL [58]. Fig. 4 plots the trajectories and the Lyapunov exponents (LEs) of these 2D chaotic systems. The trajectory of a chaotic system shows its motion behavior and the Lyapunov exponent (LE) is an effective measurement for chaos. A nonlinear system with a positive LE is chaotic and two or more positive LEs indicate hyperchaotic behavior. When plotting the trajectories, the control parameters of these 2D chaotic maps are chosen as the typical settings reported in the original literatures. Since the 2D-SLMM and 2D-LSMCL own two parameters, their LEs are calculated with the change of parameter a by setting the other parameter b as a fixed value, namely set $b = 3$ in 2D-SLMM and 2D-LSMCL. As can be seen, these 2D chaotic systems have some notable properties. First, their trajectories cannot be uniformly distributed on the whole phase space, indicating their behaviors are not random-like. In addition, their chaotic intervals aren't continuous and have periodic windows. These properties greatly effect the security level when being used in image encryption.

2.3 Generation of 3D Orthogonal Latin Square

Here, we present a method of generating 3D orthogonal Latin square [46], which will be used in our new encryption algorithms. For two Latin squares $\mathbf{A}_1 = (a_{i,j}^{(1)})^{N\times N}$ and $\mathbf{A}_2 = (a_{i,j}^{(2)})^{N\times N}$, they are orthogonal if all the element pairs $(a_{i,j}^{(1)}, a_{i,j}^{(2)})$ are different from each other. The orthogonal Latin squares have the same properties as the Latin cube. A 3D orthogonal Latin square consists of three orthogonal Latin squares. It can be any three squares of a Latin cube. Thus, a 3D orthogonal Latin square with size $n \times n \times 3$ possesses the property that the same element only appears once in different rows, columns and verticals, respectively. Therefore, we can use the Latin cube generation method presented in [46] to generated three 3D orthogonal Latin squares $\mathbf{L}_1, \mathbf{L}_2$ and $\mathbf{L}_3$, and the generation processes are described as follows.

- *Step 1*: Produce a chaotic sequence $\mathbf{X}$ with length N using a chaotic system.
- *Step 2*: Sort $\mathbf{X}$ in an ascending order and obtain the index vector $\mathbf{I}$.
- *Step 3*: Generate three 3D orthogonal Latin squares $\mathbf{L}_1, \mathbf{L}_2$ and $\mathbf{L}_3$ by performing the arithmetics to the index vector $\mathbf{I}$ in the finite field.

Algorithm 1 gives the pseudo code of generating three 3D orthogonal Latin squares $\mathbf{L}_1, \mathbf{L}_2$ and $\mathbf{L}_3$. Every 3D orthogonal Latin square satisfies the properties of the Latin cube and its three Latin squares are orthogonal to each other. Therefore, the element pairs in the three Latin squares will not be repeated and this provides a theoretical basis for designing the permutation operation in color image encryption.

3 2D-LSM

This section designs a 2D chaotic system called 2D-LSM, evaluates its chaotic behaviors using bifurcation diagram, phase plane trajectory, LE and sample entropy (SE) [31], and compares it with some newly developed 2D chaotic systems.

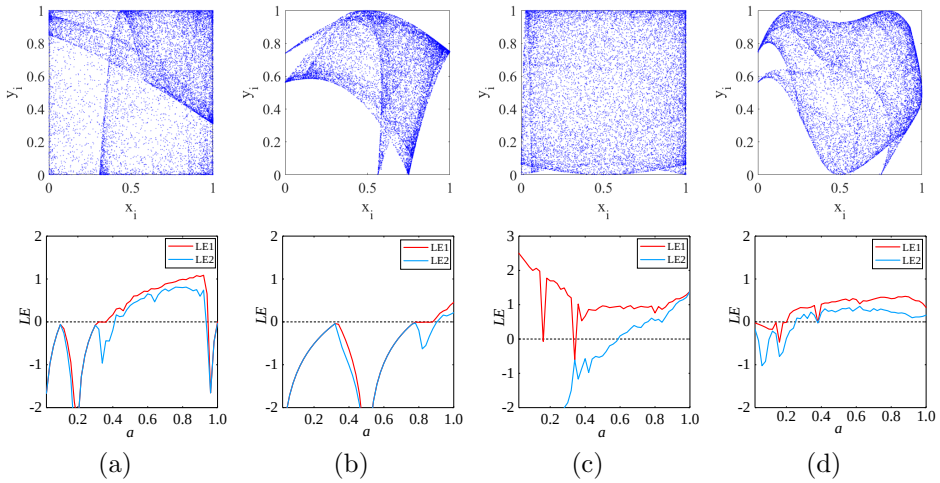


Fig. 4 The top row plots the trajectories of (a) the 2D-LASM under $a = 0.9$, (b) 2D-SLMM under $(a, b) = (1, 3)$, (c) 2D-LSCM under $a = 0.98$ and (d) 2D-LSMCL under $(a, b) = (0.75, 3)$, and the bottom row plots their two LEs by setting $b = 3$ in 2D-SLMM and 2D-LSMCL.

Algorithm 1 Generation of three 3D orthogonal Latin squares.

Input: Initial state of a chaotic system.

Generate a chaotic sequence $\mathbf{X} = \{x_0, x_1, ..., x_{N-1}\}$ by a chaotic system using the given initial.
Sort $\mathbf{X}$ and get an index matrix $\mathbf{I}$;
for $i = 0$ to $N - 1$ **do**
  for $j = 0$ to $N - 1$ **do**
    for $k = 0$ to 2 **do**
      $\mathbf{L}_1(i, j, k) = \mathbf{I}_k + \alpha \times \mathbf{I}_j + \alpha^2 \times \mathbf{I}_i$;
      $\mathbf{L}_2(i, j, k) = \mathbf{I}_k + \beta \times \mathbf{I}_j + \beta^2 \times \mathbf{I}_i$;
      $\mathbf{L}_3(i, j, k) = \mathbf{I}_k + \gamma \times \mathbf{I}_j + \gamma^2 \times \mathbf{I}_i$;
    end for
  end for
end for
where " $+$ " and " $\times$ " indicate the addition and multiplication in finite field F_N, respectively, α, β and γ denote the distinct nonzero elements in F_N.

Output: Three 3D orthogonal Latin squares $\mathbf{L}_1, \mathbf{L}_2$ and $\mathbf{L}_3$.

3.1 Definition of 2D-LSM

The classical 1D chaotic systems usually own simple system structures and low implementation cost. However, they cannot exhibit very complicated behaviors. Here, we derive a new 2D-LSM by first combining the nonlinearity of two 1D chaotic systems, and then expanding the phase space to 2D. The used two 1D chaotic systems are the Logistic and Sine maps, whose equations are defined as

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

and

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

respectively, where a and b are their parameters $a, b \in [0, 1]$. Then the 2D-LSM can be derived as

$$\begin{cases} x_{i+1} = \cos(4ax_i(1 - x_i) + b\sin(\pi y_i) + 1); \\ y_{i+1} = \cos(4ay_i(1 - y_i) + b\sin(\pi x_i) + 1). \end{cases} \tag{3}$$

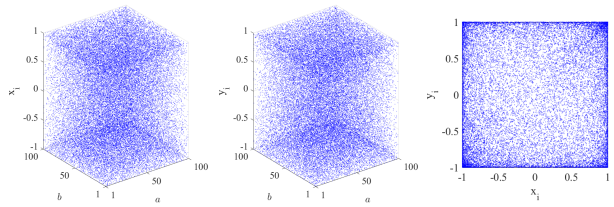
Obviously, the 2D-LSM has two control parameters a, b and they inherent from the Logistic and Sine maps, respectively. Because the cosine transform is a bounded transform for arbitrary input value, the parameters a, b can be any large values. As a result, the 2D-LSM can enlarge the ranges of the two control parameters. In this paper, we investigate the chaotic behaviors of the 2D-LSM for the two parameters $a, b \in [1, 100]$.

3.2 Performance of 2D-LSM

In this subsection, we evaluate the performance of the 2D-LSM and compare it with some newly developed 2D chaotic systems.

3.2.1 Bifurcation Diagram and Phase Plane Trajectory

The bifurcation diagram and trajectory can intuitively reflect the behaviors of a nonlinear system. For a 2D nonlinear system, its bifurcation diagram plots the visited states along the change of its control parameters, while its phase plane trajectory plots the visited points of two variables. Figs. 5(a) and (b) plot the bifurcation diagrams of variables x and y of the 2D-LSM,



(a) (b) (c)

Fig. 5 The 2D-LSM's bifurcation diagrams of (a) output x and (b) output y, and its (c) phase plane trajectory for parameters $(a, b) = (50, 50)$.

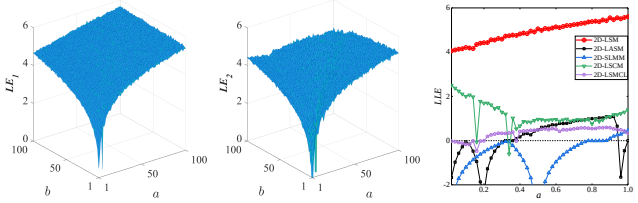
where the initials are set to $(x, y) = (0.2, 0.3)$ and the control parameters $a, b \in [1, 100]$. It can be seen that the two iterative outputs can be randomly distributed on the entire phase space within all the parameter settings, indicating random-like behaviors of the 2D-LSM. Fig. 5(c) indicates the trajectory of the first 2000 outputs of the 2D-LSM by setting the control parameters $a = b = 50$. The initial states are set as the same values as them in plotting the bifurcation diagrams. Obviously, the trajectory of the 2D-LSM can distribute throughout all the phase plane. These indicate that it owns continuous chaotic range and shows complex behaviors from the aspects of the bifurcation diagram and trajectory. With these properties, the 2D-LSM is suitable for many applications such as the encryption.

3.2.2 LE

Among all the criteria to test the chaos, the LE is a widely used one. It describe the exponential divergence of two close trajectories beginning from extremely close initials [2]. A high-dimensional dynamical systems has several LEs and its number of LEs equal to its dimension number. For a nonlinear system with global bound, its largest LE (LLE) indicates the existence of chaos. A positive LLE indicates the chaotic behavior and a larger LLE means faster divergence of close trajectories. A nonlinear system with two or more positive LEs has hyperchaotic behavior, which is a kind of more complicated behavior than the chaotic behavior.

Our experiments used the LE calculation toolbox LET[1] to obtain the LEs of different chaotic maps. First, we calculate the two LEs of the proposed 2D-LSM in the whole parameter space and the calculated values are plotted in Figs. 6(a) and (b). It can be seen that the 2D-LSM always has two positive LEs under all the setting of parameter, indicating that it has hyperchaotic behavior. Besides, we compare the LLEs of different 2D chaotic systems in Fig. 6(c). To obtain a visual effect, we linearly scale down the parameter a in the 2D-LSM from interval $(1, 100)$ to $(0, 1)$, and set its another parameter $b = 50$. For the 2D-SLMM and 2D-LSMCL, their parameter b is set as $b = 3$. As shown that, the 2D-LSM has larger LLEs than the other four 2D chaotic systems, and its chaotic range is continuous. On the contract, the other four 2D chaotic systems have discontinuous chaotic ranges.

[1] `https://ww2.mathworks.cn/matlabcentral/fileexchange/233-let?requestedDomain=zh`



(a) (b) (c)

Fig. 6 The LEs for different 2D chaotic systems. (a)-(b) The two LEs of the 2D-LSM; (c) the LLE comparisons among the 2D-LSCM ($a/100$, $b = 50$), 2D-LASM, 2D-SLMM ($b = 3$), 2D-LSCM, and 2D-LSMCL ($b = 3$).

3.2.3 SE

The SE is a kind of entropy to measure the complexity level in a time series [31]. It can quantitatively measure the complexity of the iterative outputs of a chaotic system. A positive SE means that the generated sequences does not have typical regularity, and thus show chaotic behaviors. A larger SE shows lower regularity of the sequence, and further indicates more complicated behavior of the chaotic system. Our experiments calculate the SEs of different chaotic systems using the method introduced in [31]. All the parameters in the 2D chaotic systems are set as the same values as them in the experiment of LE. Fig. 7 plots the SEs of the 2D-LSM and the SE comparisons of different 2D chaotic systems. As can be seen, the 2D-LSM can achieve positive SEs under all the control parameters and it has much larger SEs than other 2D chaotic systems. The experiment results are consistent with the results in LE experiment. As a result, the LE and SE experiments prove that our proposed 2D-LSM has superior performance than those representative 2D chaotic systems.

4 LSM-CIEA

In this section, we develop a new CIEA called LSM-CIEA using the 2D-LSM and 3D orthogonal Latin squares. Fig. 8 depicts the algorithm structure of the LSM-CIEA.

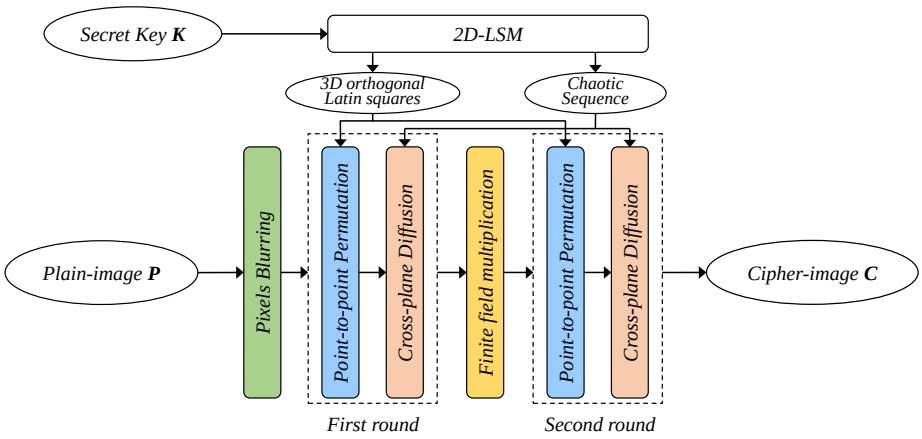


Fig. 8 The structure of the LSM-CIEA.

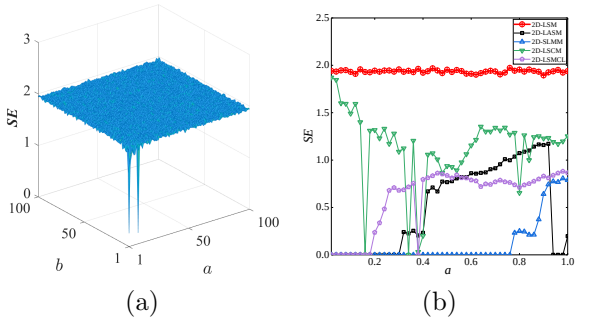


Fig. 7 The SEs for different 2D chaotic systems. (a) The SEs of the 2D-LSM; (b) the SE comparison among the 2D-LSM ($a/100$), 2D-LASM, 2D-SLMM ($b = 3$), 2D-LSCM, and 2D-LSMCL ($b = 3$).

The secret key generates the initials and control parameters of the 2D-LSM, and the chaotic sequences produced by the 2D-LSM produce 3D orthogonal Latin squares for encryption processes. First, random noises are added to the last two bits of pixels in the first column of the red color plane. Then, the point-to-point permutation is performed to randomly shuffle the pixel positions of three color planes, and the cross-plane diffusion randomly changes the pixel values. The finite filed multiplication is to enhance the security level. Since the diffusion process in the encryption algorithm can only affect the pixels behind the current pixel when the current pixel has value change, at least two rounds of encryption should be performed to totally diffuse the image pixels. Besides, two rounds of encryption can also achieve a much higher security level than only one round. Because the point-to-point permutation, cross-plane diffusion and finite filed multiplication are reversible, one can recover the original information of the plain-image with the correct secret key.

4.1 Secret Key Distribution

To defense the brute-force cracking using a computer with powerful capability, the key space should be sufficiently large. The secret key of the LSM-CIEA has the length of 256 bits and it includes 8 parts, namely $x_1, y_1, a_1, b_1, x_2, y_2, a_2$ and b_2, and each part contains 32 bits. The x_i and y_i $(i = 1, 2)$ are the initials of the 2D-LSM in the two rounds of encryption. The first bit is the sign bit, in which the 0 indicates positive value and the 1 indicates negative value. The remaining 31 bits are transformed to be a floating-point number. This strategy can ensure that the initial values are always within the output range of the 2D-LSM. The a_i and b_i are the corresponding control parameters. Their first 7 bits are converted to be the integer part, and the remaining 25 bits are converted to be the floating-point part. To avoid the ineffectiveness of the secret key with all zeros, the final parameters a and b are obtained by adding 1. Because the used 2D-LSM has continuous chaotic range when its control parameters $a, b \in [1, +\infty)$, the parameters generated from the secret key are always within the continuous chaotic range, which can avoid the non-effective keys. Suppose that $c_1c_2 \cdots c_n$ is an n-bit binary stream, its corresponding decimal floating-point number can be calculated as

$$v = \sum_{i=1}^{n} c_i 2^{-i}. \tag{4}$$

4.2 Pixels Blurring

To enhance the security of the encrypted results, we add some randomly generated noises to some pixels of the plain-image. Specifically, the pixels blurring strategy in the proposed LSM-CIEA adds some noises to

the last two bits of the first column in the red color plane. Because the last two bits only contain a little information of the pixel and the peripheral pixels only contain a little information of the image, this operation only causes an extremely small change to the plain image and don't affect its contents. After the encryption processes, these added noises can be spread to all the pixels of the three color planes. Because the added noises are random and different in every encryption, each encrypted result is totally different, even with a same secret key. With this property, the encrypted result has high ability to defense many security attacks including the chosen-plaintext attack and thus achieves a high security level.

4.3 Point-to-point Permutation

High correlations exist in the adjacent pixels of a natural image and an encryption algorithm should decorrelate these correlations efficiently. The permutation operation can effectively achieve this goal by randomly permutate the positions of pixels. In many previous works, the permutation is performed within every row/-column or obeying some predefined rules. This results in that every operation only shuffles pixels' row positions or column positions or cause low security level. In the proposed LSM-CIEA, we develop a point-to-point permutation that can totally permutate the image pixels within the three color planes of a color image using the 3D orthogonal Latin squares. This method can build a one-to-one mapping among the pixels of the plain-image and the shuffled result. In one-time operation, all the pixels' row and column positions can be totally changed, and this can achieve a high permutation efficiency. The detailed operations of the point-to-point permutation are described as follows.

- *Step 1*: Generate three orthogonal 3D Latin squares $\mathbf{L}_1, \mathbf{L}_2$ and $\mathbf{L}_3$ using the generation method introduced in Algorithm 1.
- *Step 2*: Combine the elements at the same position of $\mathbf{L}_1, \mathbf{L}_2$ and $\mathbf{L}_3$ to generate a 3D coordinate matrix $\mathbf{L}'$, namely $\mathbf{L}'(i,j,k) = (\mathbf{L}_1(i,j,k), \mathbf{L}_2(i,j,k), \mathbf{L}_3(i,j,k))$.
- *Step 3*: Reset the values in the third dimension of $\mathbf{L}'$ to get the shuffling matrix $\mathbf{L}$. Specifically, find the three coordinates in $\mathbf{L}'$ whose first two dimensions are the same, and set their third dimensions to 0, 1, and 2, respectively, in ascending order.
- *Step 4*: Shuffle the pixels of the plain-image using $\mathbf{L}$ to obtain the scrambled image $\mathbf{T}$, namely $\mathbf{T}(\mathbf{L}(i,j,k)) = \mathbf{P}(i,j,k)$.

Fig. 9 shows a number example of point-to-point permutation for a color image owning size $5 \times 5 \times 3$. Fig. 9(a) demonstrates the generation of the shuffling matrix $\mathbf{L}$ from three orthogonal 3D Latin squares $\mathbf{L}_1, \mathbf{L}_2$ and $\mathbf{L}_3$. First, a coordinate matrix $\mathbf{L}'$ is generated by combing the elements at the same position of $\mathbf{L}_1, \mathbf{L}_2$ and $\mathbf{L}_3$. Then, the shuffling matrix $\mathbf{L}$ can be generated by finding the three coordinates in $\mathbf{L}'$ whose first two dimensions are the same, and setting their third dimensions to 0, 1, and 2. For example, for the three coordinates $(0,0,2)$, $(0,0,1)$ and $(0,0,4)$ with the first two same dimensions, sort their third dimensions and set the values as 0, 1 and 2. Then the three coordinates become $(0,0,1)$, $(0,0,0)$ and $(0,0,2)$. After all the coordinates are processed, the shuffling matrix $\mathbf{L}$ can be generated. Fig. 9(b) demonstrates the shuffling process using $\mathbf{L}$. For example, since $\mathbf{L}(0,0,0) = (3,2,2)$, $\mathbf{L}(0,0,1) = (0,0,0)$ and $\mathbf{L}(0,0,2) = (4,1,1)$, then permutate the pixels with positions $(0,0,0)$, $(0,0,1)$ and $(0,0,2)$ in $\mathbf{P}$ to the positions $(3,2,2)$, $(0,0,0)$ and $(4,1,1)$ in $\mathbf{T}$, respectively. This means that $\mathbf{T}(3,2,2) = \mathbf{L}(0,0,0)$, $\mathbf{T}(0,0,0) = \mathbf{L}(0,0,1)$ and $\mathbf{T}(4,1,1) = \mathbf{L}(0,0,2)$.

Because the used 3D Latin sequences are orthogonal, each element in the coordinate matrix $\mathbf{L}$ is unique. Besides, all the elements are uniformly and randomly distributed within the sequences. These guarantee that the permutation process is point to point, and the pixels in the plain image can be distributed very randomly. Using the same shuffling matrix $\mathbf{L}$, one can totally recover the plain-image.

4.4 Cross-plane Diffusion

The diffusion property shows that slight difference in the plaintext can be spread to the whole ciphertext and an encryption algorithm should own this property. To provide a more efficiency diffusion operation, we design a cross-plane diffusion that can simultaneously process all the image pixels in the three color planes. The cross-plane diffusion can cause the change of a pixel to the next pixel. Since the adjacent pixels in the shuffled image are from different color planes and the positions are randomly determined by chaotic outputs, the process order is secret and random. The detailed cross-plane diffusion can be presented as

$$\mathbf{C}'_{i,j,k} = \begin{cases} (\mathbf{T}_{i,j,k} + \mathbf{R}^{(1)}_{i,j,k} + r_1) \bmod F \\ \qquad\qquad \text{if } i = 1, k = 1, \\ (\mathbf{T}_{i,j,k} + \mathbf{R}^{(1)}_{i,j,k} + \mathbf{C}'_{M,j,k-1} + \mathbf{T}_{M,j,k-1}) \bmod F \\ \qquad\qquad \text{if } i = 1, k \neq 1, \\ (\mathbf{T}_{i,j,k} + \mathbf{R}^{(1)}_{i,j,k} + \mathbf{C}'_{i-1,j,k} + \mathbf{T}_{i-1,j,k}) \bmod F \\ \qquad\qquad \text{if } i \neq 1, \end{cases} \tag{5}$$

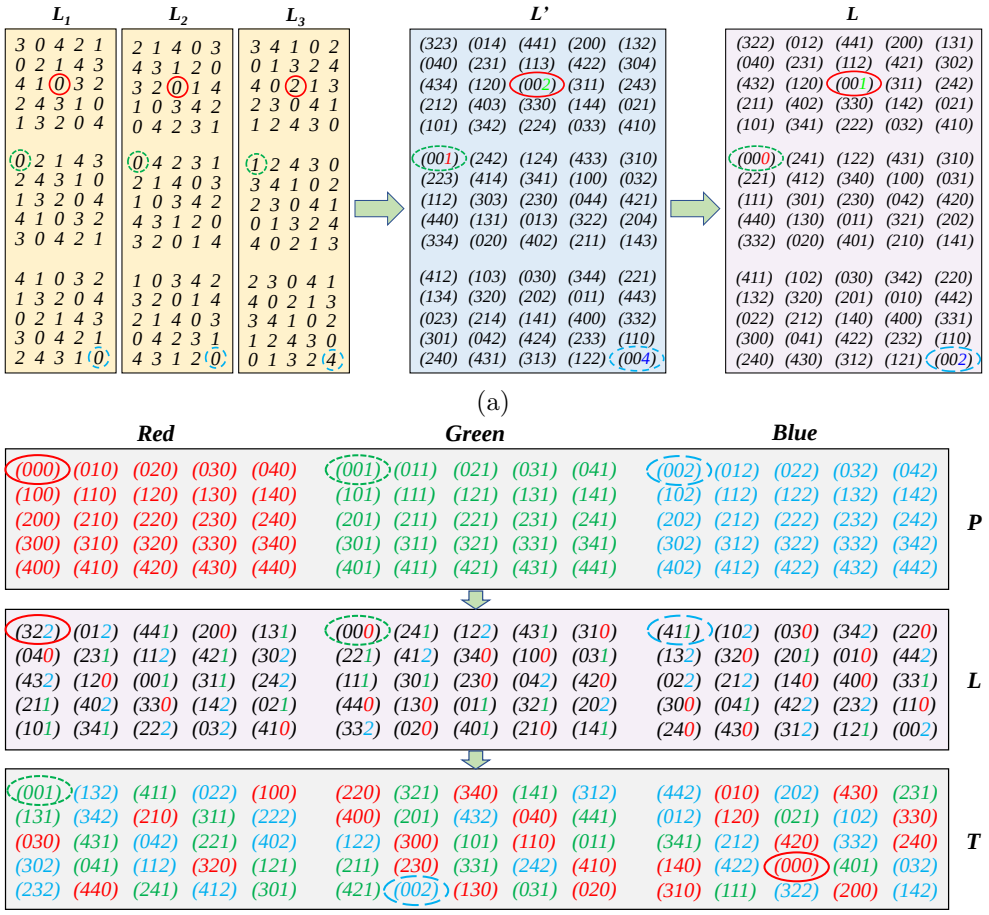


Fig. 9 A number example of the point-to-point permutation for a color image with size $5 \times 5 \times 3$. (a) Generation of the shuffling matrix $\mathbf{L}$; (b) pixel shuffling using $\mathbf{L}$.

$$\mathbf{C}_{i,j,k} = \begin{cases} (\mathbf{C}'_{i,j,k} + \mathbf{R}^{(2)}_{i,j,k} + r_2) \bmod F \\ \qquad\qquad \text{if } j = 1, k = 1, \\ (\mathbf{C}'_{i,j,k} + \mathbf{R}^{(2)}_{i,j,k} + \mathbf{C}_{i,N,k-1} + \mathbf{C}'_{i,N,k-1}) \bmod F \\ \qquad\qquad \text{if } j = 1, k \neq 1, \\ (\mathbf{C}'_{i,j,k} + \mathbf{R}^{(2)}_{i,j,k} + \mathbf{C}_{i,j-1,k} + \mathbf{C}'_{i,j-1,k}) \bmod F \\ \qquad\qquad \text{if } j \neq 1, \end{cases} \tag{6}$$

where $\mathbf{T}$ is the shuffled image, $\mathbf{R}^{(1)}$ and $\mathbf{R}^{(2)}$ are 3D chaotic matrices, r_1 and r_2 are random numbers generated by the 2D-LSM, $\mathbf{C}'$ is the temporary result after changing the pixel value in column order, $\mathbf{C}$ is the final diffusion result, and F is the greyscale levels and $F = 256$ for 8-bit image. To obtain a higher security level, the operations in Eqs. (5) and (6) can be performed in reverse order. After the cross-plane diffusion, the added random noises before encrypting can be spread to all the pixels. In the decryption operation, the shuffled image $\mathbf{T}$ can be obtained by the inverse operations using the same parameters.

4.5 Finite Field Multiplication

To obtain better diffusion characteristics and encryption effect, we divide the confused image into 4×4 image blocks, and perform multiplication in the finite field $GF(2^8)$ to each image block. First, divide the image into unduplicated 4×4 image blocks, and then the perform the finite filed multiplication as follows,

$$\mathbf{Q}_b = (\mathbf{L}_d \cdot \mathbf{C}_b \cdot \mathbf{L}_d)_{2^8}, \tag{7}$$

where $\mathbf{L}_d$ is a maximum distance separation matrix and it is defined as

$$\mathbf{L}_d = \begin{bmatrix} 4 & 2 & 1 & 3 \\ 1 & 3 & 4 & 2 \\ 2 & 4 & 3 & 1 \\ 3 & 1 & 2 & 4 \end{bmatrix}. \tag{8}$$

In the decryption operation, the inverse process of finite field multiplication can be expressed as

$$\mathbf{C}_b = (\mathbf{L}_d^{-1} \cdot \mathbf{Q}_b \cdot \mathbf{L}_d^{-1})_{2^8}, \tag{9}$$

where $\mathbf{L}_d^{-1}$ is the inverse matrix of $\mathbf{L}_d$ in the $GF(2^8)$ and it can be calculated as

$$\mathbf{L}_d = \begin{bmatrix} 71 & 216 & 173 & 117 \\ 173 & 117 & 71 & 216 \\ 216 & 71 & 117 & 173 \\ 117 & 173 & 216 & 71 \end{bmatrix}. \tag{10}$$

5 Simulation Results and Security Analysis

This section simulates the LSM-CIEA and evaluates its security level. The tested images are chosen from the USC-SIPI[1] and CVG-UGR[2] image sets. The experimental environment is Intel(R) Core(TM) i7-8700 CPU running at 3.20 GHz, with a 8 GB RAM under Windows 10 operation system.

5.1 Simulation Results

An encryption algorithm should transform all kinds of natural images to be unrecognizable cipher-images with uniform-distribution pixels. Only owning the correct key, the decryption process can totally recover all the original information of the original image. Without correct key, one cannot recover any useful information.

Fig. 10 shows the simulation results of the proposed LSM-CIEA to five color images. As shown from Fig. 10(a) that, the five test images have quit different pixel distributions and they can be encrypted to be cipher-images with uniform distributions. One cannot see any useful information from these cipher-images. Using the correct key, the LSM-CIEA can totally reconstruct the plain-images, as shown in Fig. 10(e). Thus, the LSM-CIEA is able to process all kind of color images with high security level.

5.2 Key Analysis

The secret key is very important for an encryption algorithm. The secret key's length in our proposed LSM-CIEA is 256 bits, whose key space is large to defense the brute-force attack in common scenarios. In addition, random noises are added to the last two bits of the first column of the red color plane. These noises are true random numbers and totally different in each encryption. This can also enlarge the key space.

The secret key is expected to be sensitive. To test the key sensitivity, we randomly produce a secret key K_1, and randomly change one bit in K_1 to get another two secret keys K_2 and K_3. Fig. 11 demonstrates the key sensitivity experiment in the encryption process. As can be seen, when the two secret keys only have one bit difference, the two cipher-images encrypted from a same plain-image are uniformly distributed (see Figs. 11 (b) and (c)), and totally different (see Fig. 11(d)). Fig. 12 demonstrates the key sensitivity experiment in the decryption process. As can be seen, only using the correct key, a cipher-image can be completely recovered (Fig. 12(b)). Using another keys with one bit difference, the decrypted images are noise-like (Figs. 12(c) and (d)), and completely different (Fig. 12(e)). These experiments denote that the secret keys of the LSM-CIEA have extremely sensitive encryption and decryption secret keys.

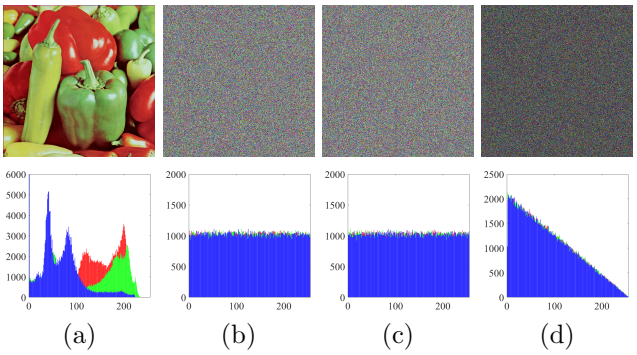


Fig. 11 Encryption key sensitivity test. (a) Plain image $\mathbf{P}$; (b) cipher image $\mathbf{C}_1 = Enc(\mathbf{P}, K_1)$; (c) cipher image $\mathbf{C}_2 = Enc(\mathbf{P}, K_2)$; (d) the difference between $\mathbf{C}_1$ and $\mathbf{C}_2$, $|\mathbf{C}_1 - \mathbf{C}_2|$.

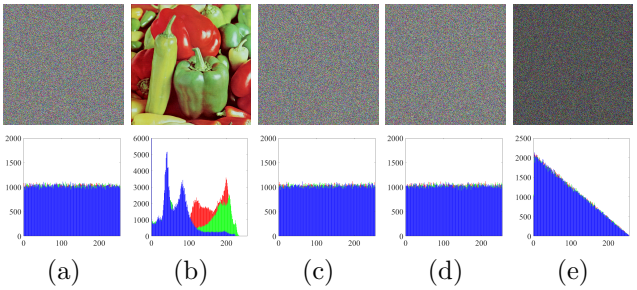


Fig. 12 Decryption key sensitivity test. (a) Cipher-image $\mathbf{C}_1$; (b) decrypted image $\mathbf{D}_1 = Dec(\mathbf{C}_1, K_1)$; (c) decrypted image $\mathbf{D}_2 = Dec(\mathbf{C}_1, K_2)$; (d) decrypted image $\mathbf{D}_3 = Dec(\mathbf{C}_1, K_3)$; (e) the difference between $\mathbf{D}_2$ and $\mathbf{D}_3$, $|\mathbf{D}_2 - \mathbf{D}_3|$.

[1] http://sipi.usc.edu/database/
[2] http://decsai.ugr.es/cvg/dbimagenes/

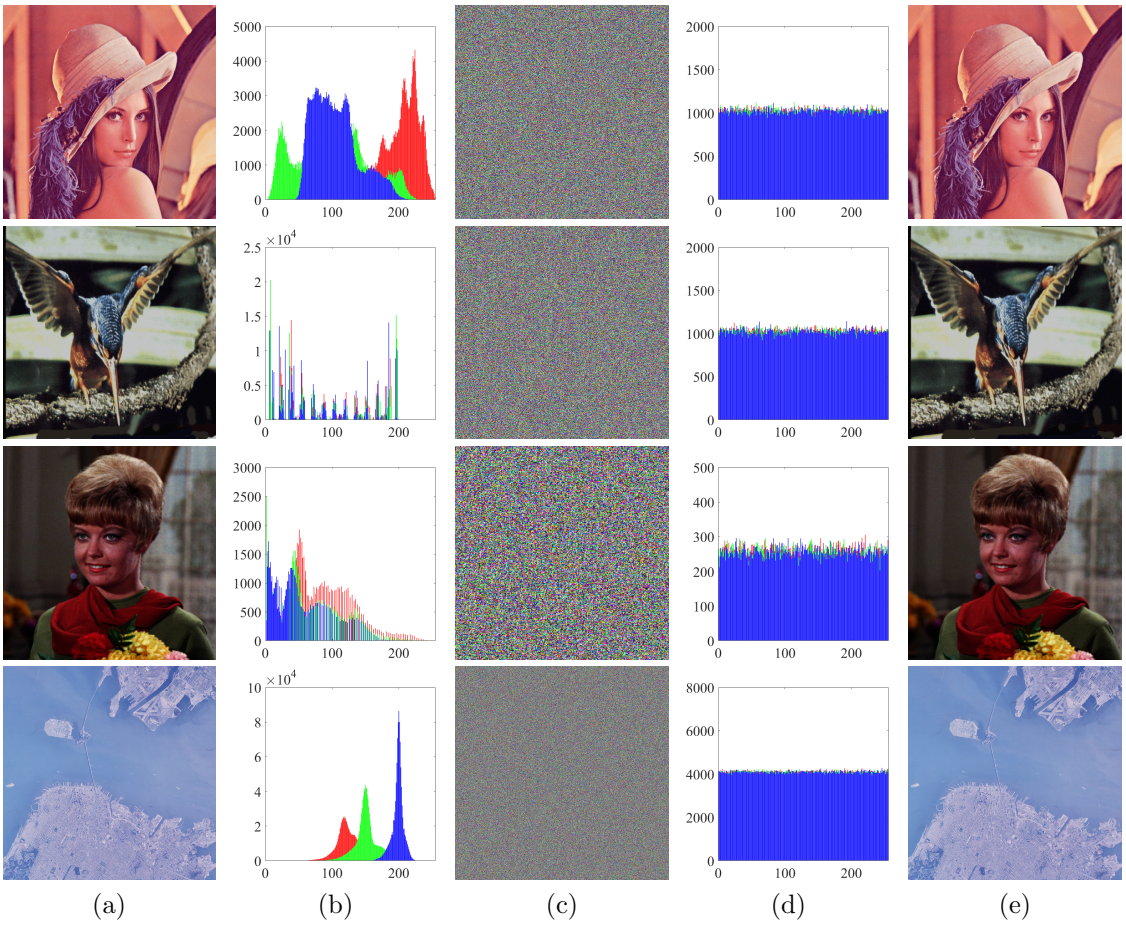


Fig. 10 Simulation results of the proposed LSM-CIEA for five color images. (a) Plain-images; (b) histograms of the plain-images; (c) cipher-images; (d) histograms of the cipher-images; (e) decrypted images.

5.3 Ability to Defense Chosen-Plaintext Attack

The chosen-plaintext attack is a commonly efficient attack mode. In this attack, attackers have the access to the encryption process and can obtain the related ciphertext for any plaintext. By choosing a certain number of plaintexts to encrypt and analyzing their related ciphertexts, one can build the inner connections between the plaintext and ciphertext. Utilizing these connections, the attackers aim to recover the information of the plaintext from the ciphertext without secret key.

The LSM-CIEA has the strong ability to defense this attack due to the following reasons: (1) The developed 2D-LSM has a continuous chaotic range and complex chaotic behaviors and thus can generate chaotic sequences with high randomness. (2) Random noises are added to the images in each encryption and these noises will be diffused over all pixels in the cipher-image. Since the added noises are randomly generated, the cipher-images produced in each encryption are different even being encrypted using a same secret key. (3) The cross-plane diffusion can diffuse the small change of pixels to all the pixels in a secret order, which depends on a chaotic sequence.

To visually show the strong ability of our proposed LSM-CIEA to defense this attack, we encrypt a plain-image twice using a same secret key and Fig. 13 depicts the experimental results. Figs. 13(b) and (c) are the cipher-images $\mathbf{C}_1$ and $\mathbf{C}_2$, which are encrypted from the twice encryptions, respectively, and Fig. 13(d) shows the difference between the two cipher-images. One can observe that the two cipher-images are completely different and this experimentally verifies that the proposed LSM-CIEA can obtain a different cipher-image in each encryption. With this property, the LSM-CIEA has strong ability to defense many commonly used security attacks including the chosen-plaintext attack.

5.4 Correlation Analysis

Because a plain-image usually exists high data redundancy, its adjacent pixels have high correlations. This

Table 2 The correlation coefficients of adjacent pixel pairs in the plain-images and their cipher-images encrypted by the proposed LSM-CIEA.

Image size	Name	Plain-image			Cipher-image		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
$128 \times 128 \times 3$	carafe	0.9571	0.9373	0.9440	0.0016	0.0008	-0.0047
	paper	0.9069	0.8988	0.8883	-0.0032	-0.0101	0.0026
	reno	0.9217	0.9025	0.8668	-0.0033	0.0019	-0.0046
$256 \times 256 \times 3$	4.1.01	0.9507	0.9701	0.9368	0.0020	-0.0025	-0.0101
	4.1.02	0.9539	0.9414	0.9093	-0.0045	-0.0023	0.0007
	4.1.03	0.9346	0.9781	0.9132	-0.0017	-0.0025	-0.0008
$512 \times 512 \times 3$	4.2.05	0.9529	0.9734	0.9299	0.0013	-0.0017	0.0005
	4.2.06	0.9521	0.9532	0.9398	0.0021	0.0050	-0.0034
	4.2.07	0.9646	0.9615	0.9547	0.0031	0.0046	-0.0021

Table 3 The correlation coefficients of adjacent pixel pairs in cipher-images by different image encryption algorithms.

Image encryption algorithms	Correlation coefficients		
	Horizontal	Vertical	Diagonal
LSM-CIEA	**0.0020**	**-0.0009**	**-0.0031**
Ref. [12]	-0.0132	-0.0085	0.0058
Ref. [39]	0.0143	0.0113	0.0237
Ref. [16]	0.0086	-0.0058	0.0055
Ref. [56]	0.0079	-0.0027	-0.0041
Ref. [13]	0.0036	0.0033	-0.0062
Ref. [30]	0.0085	-0.0037	-0.0097
Ref. [45]	0.0068	-0.0037	-0.0039

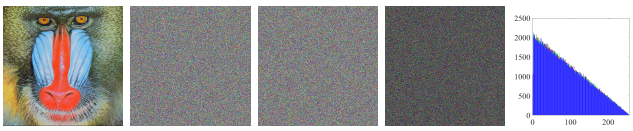


Fig. 13 Demonstration about the ability of defending chosen-plaintext attack. (a) A plain-image $\mathbf{P}$; (b) the first cipher-image $\mathbf{C}_1$; (c) the second cipher-image $\mathbf{C}_2$; (d) the difference between $\mathbf{C}_1$ and $\mathbf{C}_2$, $|\mathbf{C}_1 - \mathbf{C}_2|$; (e) the histogram of (d).

indicates that a pixel usually has similar value to its adjacent pixels. An encryption algorithm should remove these high correlations along the horizontal direction, vertical direction and diagonal direction.

To directly show the effect of our proposed LSM-CIEA to decorrealte the high correlations of plain-image, Fig. 14 plots the adjacent pixel pairs along the horizontal direction, vertical direction and diagonal direction for both the plain-image and its related cipher-image. Obviously, the pixel pairs of the plain-image are all distributed on or close to the diagonal lines of the phase space, which indicates high correlation. On the contrast, all the pixel pairs of the cipher-image are randomly distributed on the whole phase space, which means weak correlation. This shows the high security level of our proposed LSM-CIEA.

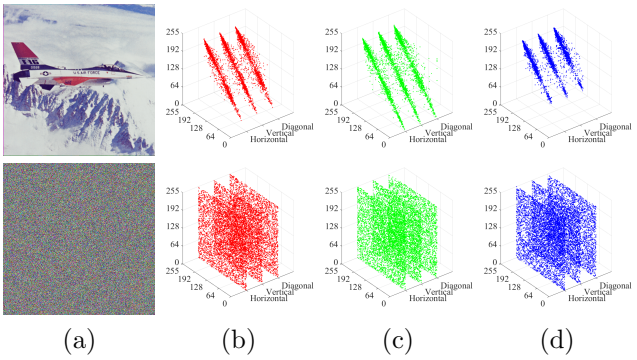


Fig. 14 Correlation analysis of the LSM-CIEA. The first row shows the plain-image, and the second row demonstrates the corresponding cipher-image. (a) Plain-image and its related cipher-image; (b) red color plane; (b) green color plane; (b) blue color plane.

To test the correlations of the adjacent pixels in the cipher-images encrypted by the proposed LSM-CIEA, we randomly chose 5000 adjacent pixel pairs along the horizontal direction, vertical direction and diagonal direction in both the plain-image and its cipher-image. Suppose $\mathbf{X}$ and $\mathbf{Y}$ are these two adjacent pixel sequences, their correlation can be calculated using the correlation coefficient, whose definition is shown as follows,

$$C(x,y) = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))}{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - E(y))^2}}, \tag{11}$$

where $E(\mathbf{X})$ and $E(\mathbf{Y})$ indicate the mathematical expectation of the sequences $\mathbf{X}$ and $\mathbf{Y}$, respectively. A large correlation coefficient indicates the high correlation of the sequences $\mathbf{X}$ and $\mathbf{Y}$ and a correlation coefficient closing to 0 indicate weak correlation.

Table 2 lists the correlation coefficients of different plain-images with their related cipher-images encrypted by our proposed LSM-CIEA. All the results are calculated from the red color plane. As can be seen, the correlation coefficients of the adjacent pixels in the plain-images are relatively high, because the plain-images have high data redundancy. However, the correlation coefficients in the cipher images are all close to 0. These indicate that the LSM-CIEA can effectively decorrelate the high correlations of adjacent pixels in the plain-images.

To show the superiority of the LSM-CIEA, we compare the correlation coefficients in the cipher-images by different encryption algorithms. The used test image is the *Lena* image with size $512 \times 512 \times 3$, and the correlation coefficients are calculated from the adjacent pixels in the red color plane. Table 3 lists the correlation coefficients of these cipher-images. The LSM-CIEA can achieve the values that are closest to 0. This further proves that the LSM-CIEA can remove the high correlations of the images efficiently.

5.5 NPCR and UACI Tests

The differential attack is another common security attack model. By selecting two plaintexts with small difference to encrypt and comparing their ciphertexts, the attackers can also build useful connections between the plaintexts and ciphertexts. An encryption algorithm can well defense this attack if it owns diffusion property. The diffusion property indicates that small changes in the plaintexts can cause the total difference in the ciphertexts.

The number of pixel change rate (NPCR) and unified averaged changed intensity (UACI) [42] are two indicators to quantitatively measure the ability of image encryption algorithms to defense the differential attack. Assuming that the two cipher-images $\mathbf{C}_1$ and $\mathbf{C}_2$ are generated by encrypting two plain-images owning only one bit difference, their NPCR and UACI can be calculated as

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

and

$$UACI(\mathbf{C}_1, \mathbf{C}_2) = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|\mathbf{C}_1(i,j) - \mathbf{C}_2(i,j)|}{H \times Q} \times 100\%, \tag{13}$$

respectively, where $M \times N$ is the size of one color plane, H indicates the total number of pixels in one color plane, Q represents the maximum allowed pixel value, and $\mathbf{W}$ is the difference between $\mathbf{C}_1$ and $\mathbf{C}_2$. $\mathbf{W}(i,j) = 0$ if $\mathbf{C}_1(i,j) = \mathbf{C}_2(i,j)$; otherwise, $\mathbf{W}(i,j) = 1$.

According to the introduction in [42], an image encryption algorithm is considered to own strong ability to defense the differential attack if the obtained NPCR is larger than a threshold 99.6094% and a larger NPCR indicates better performance. For the UACI test, an encryption algorithm is expected to have better performance if the UACI is closer to a theoretical value 33.4635%. In our experiment, we selected different sizes of images as test images and Tables 4 and 5 show the test results. From Table 4, the proposed LSM-CIEA can achieve the largest NPCR scores in most test images than the other image encryption algorithms. Besides, Table 4 shows that the LSM-CIEA can obtain UACI scores that are closer to the theoretical value 33.4635% in most images. These mean that the LSM-CIEA owns a strong ability to defense the differential attack.

5.6 Information Entropy

The information entropy is an indicator to describe the uncertainty of an signal and it can measure the distribution of image pixel. For an image $\mathbf{I}$ with F kinds of greyscale values $x_i (i = 0, 1, ..., F-1)$, its information entropy is calculated as

$$H(\mathbf{I}) = -\sum_{i=1}^{F} Pr(x_i)\log_2 Pr(x_i), \tag{14}$$

where $Pr(x_i)$ is the probability of the x_i-th possible value. When the probabilities of each possible value

Table 4 The NPCR scores for different image encryption algorithms.

Image size	Name	NPCR(%)							
		LSM-CIEA	Ref. [12]	Ref. [39]	Ref. [16]	Ref. [56]	Ref. [13]	Ref. [30]	Ref. [45]
$128 \times 128 \times 3$	carafe	**99.6644**	99.6216	99.6643	99.6277	99.4202	99.6399	99.6226	99.6053
	paper	99.6299	99.6031	**99.6501**	99.6216	99.3286	99.5239	99.6185	99.6195
	reno	**99.6436**	99.5667	99.6033	99.6326	99.4568	99.5789	99.6145	99.6134
$256 \times 256 \times 3$	4.1.01	**99.6561**	99.6460	99.6338	99.6033	99.6124	99.5773	99.5962	99.6257
	4.1.02	**99.6601**	99.5972	99.6338	99.4308	99.5422	99.2722	99.5937	99.6312
	4.1.03	99.6357	99.5743	99.5941	**99.6674**	99.4080	99.6048	99.5962	99.5865
$512 \times 512 \times 3$	4.2.05	**99.6393**	99.5819	99.6178	99.6143	99.5068	99.6220	99.6170	99.6023
	4.2.06	**99.6204**	99.6067	99.6124	99.6166	99.4507	99.6033	99.6102	99.6126
	4.2.07	**99.6413**	99.6113	99.5918	99.6342	99.5659	99.5136	99.6116	99.6218

Table 5 The UACI scores for different image encryption algorithms.

Image size	Name	UACI(%)							
		LSM-CIEA	Ref. [12]	Ref. [39]	Ref. [16]	Ref. [56]	Ref. [13]	Ref. [30]	Ref. [45]
$128 \times 128 \times 3$	carafe	**33.4636**	33.4060	33.5531	33.4682	33.6163	33.6268	33.4553	33.6138
	paper	**33.4591**	33.2230	33.6508	33.5856	34.6291	33.4857	33.4858	33.5083
	reno	33.4204	33.3425	33.4104	33.6054	33.8498	**33.4332**	33.5239	33.4997
$256 \times 256 \times 3$	4.1.01	33.4973	33.5283	**33.4731**	33.3283	33.8578	33.4265	33.2700	33.4270
	4.1.02	**33.4557**	33.3415	33.4031	33.1908	33.6860	33.4252	33.5132	33.4552
	4.1.03	**33.4623**	33.4113	33.4796	33.6099	33.2224	33.4561	33.4522	33.4688
$512 \times 512 \times 3$	4.2.05	**33.4641**	33.4643	33.4417	33.4891	32.7451	33.5089	33.4497	33.4732
	4.2.06	**33.4597**	33.4318	33.3606	33.4568	33.9676	33.4686	33.4694	33.4355
	4.2.07	**33.4670**	33.4519	33.4601	33.4125	32.9809	33.5075	33.4018	33.4938

Table 6 Information entropies of plain-images and their cipher-images by different image encryption algorithms.

Image size	Name	Plain image	Encrypted image							
			LSM-CIEA	Ref. [12]	Ref. [39]	Ref. [16]	Ref. [56]	Ref. [13]	Ref. [30]	Ref. [45]
$128 \times 128 \times 3$	carafe	3.8892	**7.9911**	7.9867	7.9891	7.9894	7.9889	7.9895	7.9884	7.9894
	paper	2.8532	**7.9910**	7.9898	7.9883	7.9908	7.9884	7.9880	7.9883	7.9887
	reno	4.3386	**7.9898**	7.9883	7.9890	7.9888	7.9881	7.9888	7.9893	7.9893
$256 \times 256 \times 3$	4.1.01	6.4200	7.9974	7.9972	7.9972	**7.9975**	7.9971	7.9968	7.9974	7.9974
	4.1.02	6.2499	**7.9977**	7.9972	7.9976	**7.9977**	7.9968	7.9975	7.9970	7.9971
	4.1.03	5.7150	**7.9976**	7.9974	7.9973	7.9973	7.9973	**7.9976**	7.9971	7.9970
$512 \times 512 \times 3$	4.2.05	6.7178	**7.9993**	7.9992	**7.9993**	**7.9993**	7.9992	**7.9993**	**7.9993**	7.9992
	4.2.06	7.3124	**7.9994**	7.9993	7.9993	**7.9994**	7.9993	7.9993	7.9993	7.9992
	4.2.07	7.3388	**7.9994**	7.9992	7.9991	7.9993	7.9993	**7.9994**	7.9993	7.9993

are equal, the information entropy can achieve a maximum value. A large information entropy indicates more uniform distribution. For an 8-bit image, it has 256 greyscale levels and its maximum information entropy can be achieved when each probability is 1/256 and the maximum information entropy is $H(\mathbf{I})_{\max} = 8$. A larger information entropy indicates more uniform distribution of the image pixels.

In our experiments, we selected 9 different images with obvious patterns as the test images. Table 6 lists the information entropies of these plain-images and their corresponding cipher-images by different image encryption algorithms. As can be seen, all the plain-images have relatively small entropies. However, the cipher-images have large entropies that are close to the theoretical maximum value 8. In addition, the proposed

LSM-CIEA can generate cipher-images with larger information entropies than other encryption algorithms. This indicates that it can outperform these other encryption algorithms and has a high security level.

6 Conclusion

With unique properties, the Lain square is an effective tool for designing image encryption algorithms. However, existing image encryption algorithms using Latin square have performance limitations in redundant operations and low efficiency, because they either treat a color image as three greyscale images or decompose a greyscale image of size 512×512 to a bit cube of size $128 \times 128 \times 128$ when performing the encryption. To solve these issues, in this paper, we first devised a new chaotic system called 2D-LSM that can overcome the weaknesses of existing chaotic systems. Using the 2D-LSM and orthogonal Latin squares, we then proposed a new CIEA called LSM-CIEA that can fully make use of the orthogonal Latin squares and color images. The LSM-CIEA mainly contains the point-to-point permutation and plane-cross-plane diffusion that can directly process the image pixels of three color planes of an color image. Simulation results show that the developed LSM-CIEA can encrypt different color images to be unrecognizable cipher-images. Security analysis show its high level of security and better performance than some state-of-the-art encryption algorithms. Since this algorithm can achieve a high performance in color image, we will explore its future application in video encryption or medical image encryption.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China under Grants 2018YFB1003800 and 2018YFB1003805, and the National Natural Science Foundation of China under Grants 62071142, 62001304 and 61701137, and the Guangdong Basic and Applied Basic Research Foundation under Grant 2019A1515110410.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Bao, H., Hua, Z., Wang, N., Zhu, L., Chen, M., Bao, B.: Initials-boosted coexisting chaos in a 2-D sine map and its hardware implementation. IEEE Transactions on Industrial Informatics **17**(2), 1132–1140 (2020)
2. Briggs, K.: An improved method for estimating Liapunov exponents of chaotic time series. Physics Letters A **151**(1-2), 27–32 (1990)
3. Chai, X., Zhang, J., Gan, Z., Zhang, Y.: Medical image encryption algorithm based on latin square and memristive chaotic system. Multimedia Tools and Applications **78**(24), 35,419–35,453 (2019)
4. Ergün, S.: On the security of chaos based ture random number generators. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences **99**(1), 363–369 (2016)
5. Fan, C., Ding, Q., Tse, C.K.: Counteracting the dynamical degradation of digital chaos by applying stochastic jump of chaotic orbits. International Journal of Bifurcation and Chaos **29**(08), 1930,023 (2019)
6. Ghadirli, H.M., Nodehi, A., Enayatifar, R.: An overview of encryption algorithms in color images. Signal Processing **164**, 163–185 (2019)
7. Gong, L., Qiu, K., Deng, C., Zhou, N.: An optical image compression and encryption scheme based on compressive sensing and RSA algorithm. Optics and Lasers in Engineering **121**, 169–180 (2019)
8. Han, M., Ren, W., Xu, M., Qiu, T.: Nonuniform state space reconstruction for multivariate chaotic time series. IEEE Transactions on Cybernetics **49**(5), 1885–1895 (2019)
9. Hirsch, M.W., Smale, S., Devaney, R.L.: Differential Equations, Dynamical Systems, and an Introduction to Chaos. Academic Press (2012)
10. Hsiao, M.Y., Bossen, D.C., Chien, R.T.: Orthogonal latin square codes. IBM Journal of Research and Development **14**(4), 390–394 (2010)
11. Hu, G., Xiao, D., Wang, Y., Li, X.: Cryptanalysis of a chaotic image cipher using latin square-based confusion and diffusion. Nonlinear Dynamics **88**(2), 1305–1316 (2017)
12. Hua, Z., Jin, F., Xu, B., Huang, H.: 2D Logistic-Sine-coupling map for image encryption. Signal Processing **149**, 148–161 (2018)
13. Hua, Z., Xu, B., Jin, F., Huang, H.: Image encryption using Josephus problem and filtering diffusion. IEEE Access **7**, 8660–8674 (2019)
14. Hua, Z., Zhang, Y., Zhou, Y.: Two-dimensional modular chaotification system for improving chaos complexity. IEEE Transactions on Signal Processing **68**, 1937–1949 (2020)
15. Hua, Z., Zhou, Y.: Image encryption using 2D Logistic-adjusted-Sine map. Information Sciences **339**, 237–253 (2016)
16. Hua, Z., Zhou, Y., Pun, C.M., Chen, C.L.P.: 2D Sine Logistic modulation map for image encryption. Information Sciences **297**, 80–94 (2015)
17. Hua, Z., Zhou, Y., Pun, C.M., Chen, C.P.: Image encryption using 2D Logistic-Sine chaotic map. In: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3229–3234 (2014)
18. Jun-xin, Chen, Zhi-liang, Zhu, Chong, Fu, Li-bo, Zhang, Yushu, Zhang: An efficient image encryption scheme using lookup table-based confusion and diffusion. Nonlinear Dynamics **81**(3), 1151–1166 (2015)
19. Kumar, S.N., Kumar, H.S., Panduranga, H.: Hardware software co-simulation of dual image encryption using Latin square image. In: 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1–5 (2013)

20. Lan, R., He, J., Wang, S., Gu, T., Luo, X.: Integrated chaotic systems for image encryption. Signal Processing **147**, 133–145 (2018)
21. Lan, R., He, J., Wang, S., Liu, Y., Luo, X.: A parameter-selection-based chaotic system. IEEE Transactions on Circuits and Systems II: Express Briefs **66**(3), 492–496 (2018)
22. Lazzús, J.A., Rivera, M., López-Caraballo, C.H.: Parameter estimation of lorenz chaotic system using a hybrid swarm intelligence algorithm. Physics Letters A **380**(11), 1164–1171 (2016)
23. Li, C.L., Li, Z.Y., Feng, W., Tong, Y.N., Du, J.R., Wei, D.Q.: Dynamical behavior and image encryption application of a memristor-based circuit system. AEU-International Journal of Electronics and Communications **110**, 152,861 (2019)
24. Li, C.L., Zhou, Y., Li, H.M., Du, W.F.J.R.: Image encryption scheme with bit-level scrambling and multiplication diffusion. Multimedia Tools and Applications (to be published, 2021)
25. Li, T., Shi, J., Li, X., Wu, J., Pan, F.: Image encryption based on pixel-level diffusion with dynamic filtering and dna-level permutation with 3d latin cubes. Entropy **21**(3), 319 (2019)
26. Lin, M., Long, F., Guo, L.: Grayscale image encryption based on Latin square and cellular neural network. In: 2016 Chinese Control and Decision Conference (CCDC), pp. 2787–2793 (2016)
27. Luo, Y., Du, M., Liu, J.: A symmetrical image encryption scheme in wavelet and time domain. Commun Nonlinear Sci Numer Simul **20**(2), 447 – 460 (2015)
28. Luo, Y., Lin, J., Liu, J., Wei, D., Cao, L., Zhou, R., Cao, Y., Ding, X.: A robust image encryption algorithm based on Chua's circuit and compressive sensing. Signal Process. **161**, 227–247 (2019)
29. Panduranga, H.T., Naveen Kumar, S.K., Kiran: Image encryption based on permutation-substitution using chaotic map and latin square image cipher. European Physical Journal Special Topics **223**(8), 1663–1677 (2014)
30. Ping, P., Wu, J., Mao, Y., Xu, F., Fan, J.: Design of image cipher using life-like cellular automata and chaotic map. Signal Processing **150**, 233–247 (2018)
31. Richman, J.S., Moorman, J.R.: Physiological time-series analysis using approximate entropy and sample entropy. American Journal of Physiology-Heart and Circulatory Physiology **278**(6), H2039–H2049 (2000)
32. Sahari, M.L., Boukemara, I.: A pseudo-random numbers generator based on a novel 3D chaotic map with an application to color image encryption. Nonlinear Dynamics **94**(1), 723–744 (2018)
33. Schuster, H.G., Just, W.: Deterministic Chaos: An Introduction. John Wiley & Sons (2006)
34. Smith, R.E.: Elementary information security. Jones & Bartlett Learning (2019)
35. Stalin, S., Maheshwary, P., Shukla, P.K., Maheshwari, M., Gour, B., Khare, A.: Fast and Secure Medical Image Encryption Based on Non Linear 4D Logistic Map and DNA Sequences. Journal of medical systems **43**(8), 267 (2019)
36. Wang, C., Wang, X., Xia, Z., Zhang, C.: Ternary radial harmonic fourier moments based robust stereo image zero-watermarking algorithm. Information Sciences **470**, 109–120 (2019)
37. Wang, C., Xia, H., Zhou, L.: A memristive hyperchaotic multiscroll Jerk system with controllable scroll numbers. International Journal of Bifurcation and Chaos **27**(06), 1750,091 (2017)
38. Wang, S., Wang, C., Xu, C.: An image encryption algorithm based on a hidden attractor chaos system and the knuth–durstenfeld algorithm. Optics and Lasers in Engineering **128**, 105,995 (2020)
39. Wang, X., Wang, Q., Zhang, Y.: A fast image algorithm based on rows and columns switch. Nonlinear Dynamics **79**(2), 1141–1149 (2015)
40. Weng, S., Shi, Y., Hong, W., Yao, Y.: Dynamic improved pixel value ordering reversible data hiding. Information Sciences **489**, 136–154 (2019)
41. Wu, X., Kurths, J., Kan, H.: A robust and lossless dna encryption scheme for color images. Multimedia Tools and Applications **77**(10), 12,349–12,376 (2018)
42. Wu, Y., Noonan, J.P., Agaian, S., et al.: NPCR and UACI randomness tests for image encryption. Cyber journals: multidisciplinary journals in science and technology, Journal of Selected Areas in Telecommunications (JSAT) **1**(2), 31–38 (2011)
43. Wu, Y., Zhou, Y., Noonan, J.P., Agaian, S.: Design of image cipher using latin squares. Information Sciences **264**, 317–339 (2014)
44. Xu, C., Sun, J., Wang, C.: An image encryption algorithm based on random walk and hyperchaotic systems. International Journal of Bifurcation and Chaos **30**(4), 2050,060 (2020)
45. Xu, L., Li, Z., Li, J., Hua, W.: A novel bit-level image encryption algorithm based on chaotic maps. Optics and Lasers in Engineering **78**(MAR.), 17–25 (2016)
46. Xu, M., Tian, Z.: A novel image cipher based on 3D bit matrix and latin cubes. Information Sciences **478**, 1–14 (2019)
47. Xu, Y.M., Yao, Z., Hobiny, A., Ma, J.: Differential coupling contributes to synchronization via a capacitor connection between chaotic circuits. Frontiers of Information Technology & Electronic Engineering **20**(4), 571–583 (2019)
48. Ye, H.S., Zhou, N.R., Gong, L.H.: Multi-image compression-encryption scheme based on quaternion discrete fractional hartley transform and improved pixel adaptive diffusion. Signal Processing **175**, 107,652 (2020)
49. Zhang, W., Yu, H., Zhao, Y.L., Zhu, Z.L.: Image encryption based on three-dimensional bit matrix permutation. Signal Processing **118**, 36–50 (2016)
50. Zhang, Y.: The fast image encryption algorithm based on lifting scheme and chaos. Information Sciences **520**, 177–194 (2020)
51. Zhang, Y., Ren, G., Hobiny, A., Ahmad, B., Ma, J.: Mode transition in a memristive dynamical system and its application in image encryption. International Journal of Modern Physics B **34**(27), 2050,244 (2020)
52. Zhang, Z., Yu, S.: On the security of a latin-bit cube-based image chaotic encryption algorithm. Entropy **21**(9), 888 (2019)
53. Zhou, J., Zhou, N.R., Gong, L.H.: Fast color image encryption scheme based on 3D orthogonal Latin squares and matching matrix. Optics & Laser Technology **131**, 106,437 (2020)
54. Zhou, N., Hu, Y., Gong, L., Li, G.: Quantum image encryption scheme with iterative generalized Arnold transforms and quantum image cycle shift operations. Quantum Information Processing **16**(6), 164 (2017)
55. Zhou, N., Pan, S., Cheng, S., Zhou, Z.: Image compression–encryption scheme based on hyper-chaotic system and 2D compressive sensing. Optics & Laser Technology **82**, 121–133 (2016)

56. Zhou, Y., Bao, L., Chen, C.P.: A new 1D chaotic system for image encryption. Signal processing **97**, 172–182 (2014)
57. Zhou, Y., Li, C., Li, W., Li, H., Feng, W., Qian, K.: Image encryption algorithm with circle index table scrambling and partition diffusion. Nonlinear Dynamics (to be published, 2021)
58. Zhu, H., Zhao, Y., Song, Y.: 2D Logistic-Modulated-Sine-Coupling-Logistic Chaotic Map for Image Encryption. IEEE Access **7**, 14,081–14,098 (2019)
59. Zhu, Z.l., Zhang, W., Wong, K.w., Yu, H.: A chaos-based symmetric image encryption scheme using a bit-level permutation. Information Sciences **181**(6), 1171–1186 (2011)