Complex & Intelligent Systems (2023) 9:927–950
https://doi.org/10.1007/s40747-022-00835-1

ORIGINAL ARTICLE

CIE-LSCP: color image encryption scheme based on the lifting scheme and cross-component permutation

Xiuhui Chen[1] · Mengxin Gong[1] · Zhihua Gan[2] · Yang Lu[1] · Xiuli Chai[1] · Xin He[2]

Received: 8 July 2021 / Accepted: 16 June 2022 / Published online: 16 August 2022

Abstract
Encryption of meaningful images into unidentifiable noise-like images can improve the security of images during storage and transmission. In this paper, a novel color image encryption method based on the lifting scheme and cross-component permutation (CIE-LSCP) is introduced. First, original image is divided into red, green and blue components, and then the three components are processed by a preprocessing strategy based on the lifting scheme (PSLS) to change the statistical distribution of the pixels. Second, a block-based cross-component permutation based on index vectors (BCPIV) is presented to divide three preprocessed components into blocks and perform the cross-component permutation operation on them, and three random matrices are utilized to determine the target component, target block and target pixel position of the current pixel to be moved, respectively. Subsequently, a multi stochastic diffusion based on random sequences (MSDRS) is developed to alter the pixel values of the permutated components, and finally a ciphertext image is gotten by merging the three components. Moreover, the SHA256 hash values of the plaintext image are used to obtain the initial parameters of the chaotic system, and the obtained chaotic sequences are applied in the image encryption process. Wherein the generated random sequences are highly reliant on the plaintext image, making the encryption scheme resistant to both known-plaintext attacks and chosen-plaintext attacks. Experimental results demonstrate that the proposed scheme has good security and effectiveness and can be applied for secure transmission of digital images over the Internet.

Keywords Color image encryption · Chaotic system · The lifting scheme · Cross-component permutation · Multi stochastic diffusion

Introduction

With the advent of the era of big data, more and more images are generated, transmitted and stored over the internet, and the value of data has become increasingly prominent. Some images may contain individual secret information, military secret, and even national strategy. Once they are copied and tampered with by malicious attackers, the consequences are very serious. Therefore, it is urgent to find effective schemes to protect image information. The encryption-decryption process and security of traditional cryptography are based on mathematical puzzles, and with the exception of one-time-pad cryptography [1], all other encryption systems are only computationally secure to the extent that they can be deciphered if an attacker has sufficient computational power. For the scrambling-only encryption process, it just changes the position of pixels in the plaintext image without altering its histogram [2], thus they are less able to resist statistical attacks. To solve the security problems of images transmission, various methods have been proposed by many scholars [3–6], such as information hiding, image encryption, etc. Image encryption has demonstrated its superior performance to secure images, and some image encryption methods have been introduced.

Fridrich [7] gave a breakthrough confusion-diffusion framework for the image encryption system in 1998, and

✉ Zhihua Gan
gzh@henu.edu.cn

✉ Xiuli Chai
chaixiuli@henu.edu.cn

✉ Xin He
hx_henu@126.com

[1] School of Artificial Intelligence, Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng 475004, China

[2] School of Software, Intelligent Data Processing Engineering Research Center of Henan Province, Institute of Intelligent Network System, Henan University, Kaifeng 475004, China

then scholars have proposed some encryption schemes [8, 9] using this structure. For instance, Wang et al. [10] developed a fast image encryption algorithm based on parallel computing system. The plaintext image was first performed with a permutation algorithm that combined cyclic shift and sorting, and next a parallel diffusion method was utilized to obtain the ciphertext image. Saiyma et al. [11] presented a novel bit-permutation-based image encryption algorithm, and used a three-dimensional puzzle along with chaos for further diffusion and confusion. Besides, Wang et al. [12] introduced a spatiotemporal chaos and its application in a bit-level image encryption scheme, this algorithm contained three main process: the generation of keys, confusion and sequential diffusion operations. Wen et al. [13] proposed a hyper-chaos-based image cryptosystem with a classical bi-modular architecture, performed random permutation and sequential diffusion on original images. Wang et al. introduced a novel chaotic image encryption algorithm based on DNA sequence operations [14] and a novel chaotic block image encryption scheme using dynamic random growth technique [15]. In [8–15], the plaintext images are all grayscale image, but in real life, color images are more universal, and there is intense correlation between red, green and blue components of color images. However, if the above algorithms are directly applied for color images, it will lead to low security. Therefore, designing color image encryption schemes is necessary.

Recently, many encryption schemes for color image have been developed [16, 17]. Rehman et al. [18] gave a color image encryption algorithm based on DNA complementary rules, which first designed a permutation method based on DNA complementary rules, and then used random sequences and DC-boxes to diffuse the permuted matrix to generate high-quality ciphertext images. Dua et al. [19] designed a scheme to encrypt images based on the cosine transformation of a 3-D intertwining Logistic map, used the generated chaotic sequences to displace, rotate, and replace the plain image to form a secure ciphertext image, and the experiments showed that the proposed encryption scheme is more efficient. Wang et al. [20] introduced an image encryption algorithm based on matrix semi-tensor product and composite key, designed a new permutation method and composite key generation approach, making the generated encrypted images more secure and effective. They also designed single-parameter fractal ordering matrices [21] and two-parameter fractal ordering matrices [22] for the encryption process, which improved the encryption speed and the security of the encrypted images. Khan et al. [23] introduced an image cryptographic scheme based on a permutation-diffusion framework, which is more productive due to the use of predefined S-boxes, Chebyshev polynomials and fractal Tromino in the encryption process. Moreover, a fast color image cipher was proposed by zhou et al. [24]. The color image was transformed into a 3D matrix, and then it was permutated by the 3D orthogonal Latin square and matching matrix generated from the 3D sine map, subsequently, the shift operation and diffusion operation were performed on the permutated image to find the cipher image. Teng et al. [25] presented a color image encryption based on fisher-yates scrambling algorithm, the SHA-384 was first used to generate the key, and then three groups of chaotic sequences were obtained to encrypt red, green and blue components of the plain color image, respectively.

Because of its pseudo randomness and sensitivity to initial parameters, chaotic systems [26, 27] are used by many encryption projects. Wang et al. [28] proposed a novel one-dimensional chaotic map generator and applied it in a new index-representation-based image encryption scheme. The chaotic system obtained by their proposed one-dimensional chaotic map amplifier enhances the chaotic behavior and has high sensitivity. Talhaoui et al. [29] designed a new one-dimensional chaotic map and used it in the encryption method, experimental results show that it has sound cryptographic properties. However, due to the low-dimensional chaotic systems with less parameters, simplicity of orbits, and easy estimation of parameters and initial values, high-dimensional chaotic systems [30, 31] and hyper-chaotic systems are employed in image encryption process to overcome these disadvantages, and there is theoretical evidence that chaotic systems can resist dynamic degradation [32]. Mou et al. [33] constructed an improper fractional-order laser chaotic system, for the purpose of studying the application of fractional-order chaos systems in image encryption. Wang et al. [34] proposed a multi-dynamic segmented coupled mapping network with well chaotic behavior and designed an image encryption scheme based on it, which satisfies the cryptographic requirements. In [35], Dua et al. suggested a cryptographic operation for color images using alternate logistic maps. The parameters of the Arnold cat mapping were generated with logistic sine maps and logistic tent maps, the produced random sequences were applied to the bit-plane permutation, followed by combining the bit planes into pixel matrices, which were then scrambled and diffused by the chaotic sequences generated from alternate logistic maps. Mou et al. [36] found a novel image encryption algorithm based on the fractional-order hyper-chaotic complex system and Galois field, and later, they [37] also proposed another encryption method using hyper-chaotic system. In this article, a three-dimensional Lorenz–Haken laser chaotic system is utilized to generate quasi-random chaotic sequences for image encryption and enhance the resistance of the method to attacks.

Though the above image encryption methods have relative satisfactory performance, there exists some security and efficiency problems. First, in the above image encryption, most algorithms are based on permutation-diffusion

structure, wherein, when some special images (such as all black and all white images) are encrypted at pixel level, the problem of invalid scrambling often occurs, this is to say, the scrambled image is the plaintext images even after many round permutation operations, which seriously affects the security and efficiency of the image cryptosystems. Besides, as for those color image encryption, red, green and blue components are encrypted separately, the high correlation between three components is not removed effectively, which makes it possible for hackers to attack them by statistical attack. Additionally, some diffusion operations of image encryption have been presented, chaotic sequences are adopted to perform XOR operation with pixels of plaintext images in order, this kind of diffusion methods are simple and have low randomness, and are easily cracked by attackers who know the chaotic sequences beforehand, it reduces the security of image encryption algorithms.

To settle with these issues, an effective color image encryption algorithm is designed based on the lifting scheme and cross-component permutation in this article, named as CIE-LSCP. First, the color image is decomposed to red, green and blue components, three chaotic sequences are generated based on the plaintext image information and Lorenz–Haken laser chaotic system, then the chaotic sequences are applied to sequentially interfere the three components, the lifting scheme is employed for preprocessing operation. Second, a block-based cross-component permutation based on index vectors is designed in this paper to alter the positions of pixels in three components obtained by preprocessing method. Finally, three components acquired from permutation strategy are processed by a multi random diffusion operation, and a color ciphertext image is obtained by combining the three diffused components. The contributions of this paper are summarized as follows:

1. A preprocessing strategy based on the lifting scheme (PSLS) is adopted to vary the statistical distribution of the plaintext images pixels. PSLS is implemented before permutation-diffusion manipulation, the information of plaintext images and Lorenz–Haken laser chaotic system are employed to generate three random sequences, and then addition or XOR operations are performed between two selected components or between component and random sequence through the use of the lifting scheme. It can prevent attackers from using all black and all white images to invalidate the scrambling process, thus improving the security of our image encryption scheme.
2. A block-based cross-component permutation based on index vectors (BCPIV) is developed to shuffle the pixels among three components efficiently. The sequences associated with plaintext images are manipulated and sorted to obtain the corresponding index vectors, then three random matrices are obtained by cyclic shift operations on index vectors. Besides, three random matrices are utilized to determine the target component, target block and target pixel position of the current pixel to be moved respectively, which makes BCPIV extremely correlated with the plaintext images and enhances the stochasticity of permutation procedure.
3. A multi stochastic diffusion based on random sequences (MSDRS) is introduced to modify the pixel values of images. In MSDRS, the pixel of current component to be diffused is determined by a random sequence, the random value involved in the operation, the pixel of another component and the target position of the diffused pixel are established by stochastic sequences. As opposed to the diffusion according to the order of the image pixels, MSDRS is more random and is less likely to be attacked by hackers.
4. The proposed image encryption is highly sensitive to the plaintext image. A three-dimensional Lorenz–Haken laser chaotic system is utilized for generating key sequences to be used in preprocessing, permutation and diffusion stage. To improve the dependence of image encryption with the plaintext image, the SHA 256 hash values of the original image are computed to obtain the initial parameters of the chaotic system, thus the produced chaotic sequences highly depend on the plaintext image, which makes the known-plaintext and chosen-plaintext attacks invalid.

The remainder of this paper is organized as follows. The fundamental knowledge including the Lorenz–Haken laser chaotic system and the lifting scheme is provided in "Fundamentals". The image encryption scheme is introduced in "The proposed image encryption scheme". The experimental results are presented in "Simulation results and performance analyses", followed by conclusion in "Comparative analysis".

Fundamentals

In this section, we will describe the Lorenz–Haken laser chaotic system and the lifting scheme. The random sequences produced by the chaotic system are crucial for our encryption scheme, and the lifting scheme is implemented for the preprocessing step in the encryption procedure.

Lorenz–Haken laser chaotic system

Lorenz–Haken laser chaotic system [38] is adopted in this paper, which can be expressed by

$$\begin{cases} \dot{x}=\sigma y-\sigma x \\ \dot{y}=-xz+\gamma x-y, \\ \dot{z}=xy-bz \end{cases} \tag{1}$$

where x, y and z are the system state variables, σ, γ and b are parameters of the system. The choice of three parameters in the system plays an important role in whether the system will enter a chaotic state or not, σ, γ and b can be taken as any value greater than 0, they are typically taken as $\sigma = 10$, $\gamma = 28$, and $b = 8/3$. When $\sigma < (b+1)$, the system is always in a steady state. When $\sigma > (b+1)$, if $\gamma < \frac{\sigma(\sigma+b+3)}{\sigma-(b+1)}$, the system is in a stable state, and if $\gamma > \frac{\sigma(\sigma+b+3)}{\sigma-(b+1)}$, the system is in a chaotic state. Therefore, keeping $\sigma = 10$ and $b = 8/3$, the system starts to enter into the chaotic state when $\gamma > 24.74$ [39].

The used Lorenz–Haken laser chaotic system is characterized by unstable, irreducible and unpredictable behavioral performance, periodic and non-periodic motions are entangled with each other, while being very sensitive to initial conditions, making the generated sequences more random. Besides, the system structure is more complex than low-dimensional chaotic systems, and the values of system variables are harder to predict. We can also treat the random sequences produced by the chaotic system, for example, multiple sequences are combined to generate new ones, which makes the design of stochastic sequences for encryption algorithms more flexible. Additionally, the initial values of the three system state variables and the three system parameters of Lorenz–Haken laser chaotic system can be used as keys for generating random sequences, and thus the key space of the used image encryption algorithm will be much larger than those using the low-dimensional chaotic systems. Furthermore, this chaotic system also has some limitations. For example, it will take longer time to iterate Lorenz–Haken laser chaotic system to obtain random sequences compared to the low-dimensional chaotic system.

The lifting scheme

In 1994, Sweldens [40] introduced a new approach to calculate wavelet transform named as the lifting scheme, which is mainly divided into three stages: splitting, prediction and updating.

Splitting the process is to divide the data set into two subsets, and the data set is represented as $T = \{T_i^0\}, i = 0, 1, 2, \ldots, n-1$. It is usually divided according to the odd-indexed and even-indexed, the data of even-indexed constitute a subset, the other data form another subset, and this step is expressed as:

$$st_i^1 = T_{2i}^0,\ dt_i^1 = T_{2i+1}^0,\ i = 0, 1, 2, \ldots, n/2-1. \tag{2}$$

Prediction during this step, subset $\{dt_i^1\}$ can be calculated with another subset $\{st_i^1\}$ to get a new one, which is still represented by $\{dt_i^1\}$, such that:

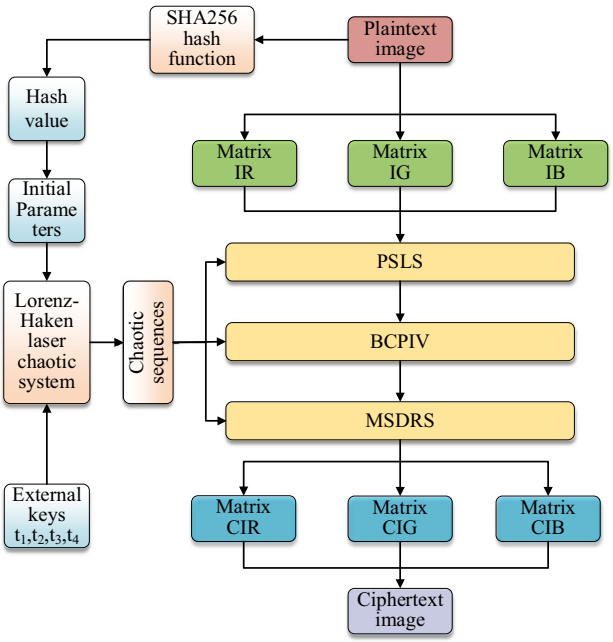

Fig. 1 The flowchart of the proposed image encryption scheme

$$\mathrm{d}t_i^1 = \mathrm{d}t_i^1 - \frac{1}{2}\left(st_i^1 + st_{i+1}^1\right),\ i = 0, 1, 2, \ldots, n/2-1. \tag{3}$$

Updating in this process, the remaining subset $\{st_i^1\}$ is updated by the subset $\{dt_i^1\}$, i.e., $\{st_i^1\}$ is calculated with the subset $\{dt_i^1\}$ to get a new subset $\{st_i^1\}$, as follows:

$$st_i^1 = st_i^1 + \frac{1}{4}\left(\mathrm{d}t_{i-1}^1 + \mathrm{d}t_i^1\right),\ i = 0, 1, 2, \ldots, n/2-1. \tag{4}$$

In this article, the lifting scheme is utilized to preprocess the color plaintext images to alter their pixel values in the first time.

The proposed image encryption scheme

Encryption algorithm

In this section, a new color image encryption scheme named as CIE-LSCP is proposed and illustrated in Fig. 1, and the main contents are described in detail as follows. Three random sequences are obtained based on the plaintext images information and Lorenz–Haken laser chaotic system. Next, preprocessing strategy based on the lifting scheme (PSLS) is developed to process the plaintext images. Subsequently, the permutation and diffusion operations are manipulated on the preprocessed image to get the ciphertext image.

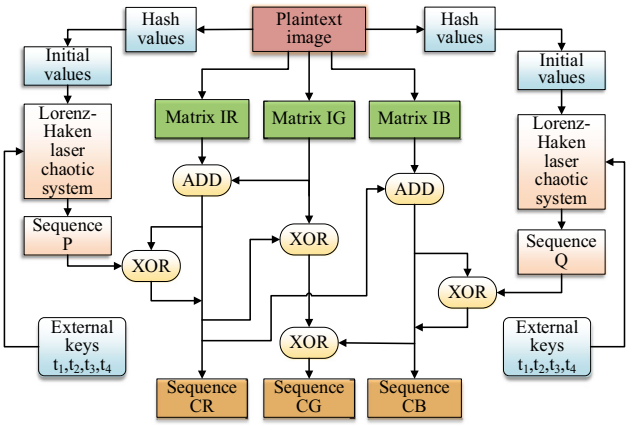

Fig. 2 The flowchart of PSLS

Generation of initial values of chaotic system

A 256-bit hash value H of the plaintext images I sized of $M \times N$ is generated by the SHA 256 hash function in the proposed encryption scheme. Then, H is divided into 16-bit blocks, which are converted into 16 decimal numbers as: $H = h_1, h_2, \ldots, h_{16}$. Next, calculate the initial values of the chaotic system by

$$\begin{cases} x(0) = \mathrm{mod}\left(\dfrac{1}{M \times N} \times t_1 \times \dfrac{h_1}{h_2} + h_5 + h_6 + h_7 + h_8,\ 10\right) - 5 \\ y(0) = \mathrm{mod}\left(\dfrac{1}{M \times N} \times t_2 \times \dfrac{h_1}{h_3} \times (h_9 \oplus h_{10} \oplus h_{11} \oplus h_{12}),\ 30\right) - 15, \\ z(0) = \mathrm{mod}(\dfrac{1}{M} \times t_3 \times (\dfrac{h_1}{h_4} + \dfrac{h_{13}}{h_{14}} + t_4) \times (h_{15} \oplus h_{16}),\ 30) + 30 \end{cases} \tag{5}$$

where t_1, t_2, t_3, t_4 are external keys, M is the row number of the plaintext images, N is the column number of the plaintext images, $x(0)$, $y(0)$ and $z(0)$ are initial values of the chaotic system, and $\oplus$ indicates the XOR operation.

Preprocessing strategy based on the lifting scheme (PSLS)

To solve the problem that the confusion is invalid when image encryption with confusion-diffusion architecture encrypts all-black and all-white images, preprocessing strategy based on the lifting scheme (PSLS) is proposed in this algorithm, and applied to process the plaintext image before permutation. As shown in Fig. 2, the red, green, blue components of the color plaintext images are preprocessed and the detailed operations are described as follows.

Step 1 According to the hash value of the plaintext image I with the size $M \times N$, three initial values of chaotic system can be obtained. Then, iterate this system for N_0 ($N_0 \geq 1000 + M \times N$) times with them according to Eq. (1), discard the first k ($k \geq 500$) values to avoid the transient effect, then generate three chaotic sequences O, P and Q with the length of MN. Convert sequences O, P, Q into integer series by

$$\begin{cases} O = \mathrm{mod}\left(floor\left(|O + P| \times 10^{10}\right), 256\right) \\ P = \mathrm{mod}\left(floor\left(|P + Q| \times 10^{10}\right), 256\right). \\ Q = \mathrm{mod}\left(floor\left(|Q + P| \times 10^{10}\right), 256\right) \end{cases} \tag{6}$$

Step 2 The color image I is decomposed into red, green and blue components, and they are represented as three matrices IR, IG, IB sized of $M \times N$. For each matrix, they are expanded into one-dimensional vectors R_1, G_1, B_1 with length MN row by row. The corresponding index O_1, P_1 and Q_1 are obtained by arranging O, P and Q in ascending order. Three arrays R_1, G_1 and B_1 were sorted by O_1, P_1 and Q_1 respectively, and then three sequences R_2, G_2 and B_2 with length MN were obtained by

$$\begin{cases} R\_2 = R\_1(O\_1) \\ G\_2 = G\_1(P\_1). \\ B\_2 = B\_1(Q\_1) \end{cases} \tag{7}$$

Step 3 Sequence G_2 was applied to predict alignment R_2, and a new array NR was obtained by

$$NR_i = \mathrm{mod}\left(floor\left(\frac{1}{2}(G\_2_{i-1} + G\_2_i)\right) + R\_2_i,\ 256\right), \tag{8}$$

where $\mathrm{mod}(a, b)$ denotes the modular operation of a to b, the $\mathrm{floor}(x)$ function signifies the maximum integer that is not larger than x. NR_i, G_2_i, R_2_i are the ith numerical element in the sequence NR, G_2, R_2 respectively, and i is an integer, $i \in [1,\ M \times N]$, $G\_2_0 = 0$.

Step 4 Sequences G_2 and P were used to disturb with series NR, and a new serial CR was obtained by

$$CR_i = \mathrm{mod}\left(floor\left(\frac{1}{2}(G\_2_i + \mathrm{mod}(P_i, 256))\right), 256\right) \oplus NR_i, \tag{9}$$

where CR_i, P_i, NR_i are the ith numerical element in the series CR, P, NR respectively, and i is an integer, $i \in [1,\ M \times N]$.

Step 5 Sequence G_2 was updated by array CR, and a new serial CR was obtained by

$$NG_i = \mathrm{mod}(CR_{i-1} \oplus CR_i + G\_2_i, 256), \tag{10}$$

where NG_i is the ith numerical element in the sequence NG, and i is an integer, $i \in [1,\ M \times N]$, $CR_0 = 0$.

Step 6 Array *CR* was employed to predict sequence *B_2*, and a new serial *NB* was gotten by

$$NB_i = \mathrm{mod}\left(floor\left(\frac{1}{2}(CR_{i-1} + CR_i)\right) + B\_2_i,\ 256\right), \tag{11}$$

where NB_i, B_2_i are the ith numerical element in the alignments *NB* and *B_2* separately, and i is an integer, $i \in [1,\ M \times N]$.

Step 7 Sequences *CR* and *Q* were used to disturb with array *NB*, and a new serial *CB* was obtained by

$$CB_i = \mathrm{mod}\left(floor\left(\frac{1}{2}(CR_i + \mathrm{mod}(Q_i,\ 256))\right),\ 256\right) \oplus NB_i, \tag{12}$$

where CB_i, Q_i are the ith numerical elements in the sequences *CB* and *Q* severally, and i is an integer, $i \in [1,\ M \times N]$.

Step 8 Alignment *NG* was updated by sequence *CB*, and a new serial *CG* was obtained by

$$CG_i = \mathrm{mod}(CB_{i-1} \oplus CB_i + NG_i,\ 256), \tag{13}$$

where CG_i is the ith numerical element in the sequence *CG*, and i is an integer, $i \in [1,\ M \times N]$, $CB_0 = 0$.

Finally, three sequences *CR*, *CG* and *CB* with length *MN* are obtained.

In the PSLS, the lifting scheme and random sequences are applied to perform addition and XOR operations on plaintext images pixels, which changes the pixel values and prevents the problem of permutation invalidation caused by attackers using all zeros and all ones images against encryption algorithm. The pseudocode of PSLS is described in Algorithm 1.

Block-based cross-component permutation based on index vectors (BCPIV)

Three sequences *CR*, *CG* and *CB* are recombined to get three matrices *PR*, *PG* and *PB* with size $M \times N$, and then the three matrices are respectively shuffled by the proposed BCPIV as follows:

Step 1 Select three matrices *PR_2*, *PG_2*, *PB_2* with size $L^2 \times L^2$ from *PR*, *PG* and *PB* respectively, *PR_2*, *PG_2*, *PB_2* are divided into L^2 blocks in turn, and the size of each block is $L \times L$. L is obtained by

$$L = \min\left\{\left\lfloor\sqrt{M}\right\rfloor,\ \left\lfloor\sqrt{N}\right\rfloor\right\}, \tag{14}$$

where $\lfloor a \rfloor$ is the floor operation to obtain the largest integer that is not greater than a.

Step 2 The first L^2 data in the sequences *O*, *P*, *Q* are selected and denoted as *O_3*, *P_3*, *Q_3*, respectively. Four new alignments *T*, *H*, *Y* and *Z* were acquired by

$$\begin{cases} T_i = O\_3_i + P\_3_i \\ H_i = P\_3_i + Q\_3_i \\ Y_i = O\_3_i - Q\_3_i \\ Z_i = O\_3_i + P\_3_i - Q\_3_i \end{cases}, \tag{15}$$

where T_i, H_i, Y_i, Z_i are the ith numerical elements in the sequences *T*, *H*, *Y*, *Z* separately, and i is an integer, $i \in [1,\ L^2]$. The corresponding index vectors *IT*, *IH*, *IY* and *IZ* are obtained by arranging *T*, *H*, *Y* and *Z* in ascending order, respectively.

Step 3 Initialize two matrices *S* and *W* with size $L^2 \times L^2$, set the values of *IT* to each row of *S*, and the values of *IH* to each row of *W*. Each row in *S* is cyclically shifted using the elements in *IY*, and each row in *W* is cyclically moved

Algorithm 1 The pseudocode of PSLS.

Input: Plaintext images *I* of size *M*×*N* and external keys.

Output: Three sequences *CR*, *CG*, *CB*.

1: Generate initial parameters of three chaotic system by hash value of original image, iterate Eq. (1) to obtain three random sequences *O*, *P*, *Q* with length *MN*.

2: Convert sequences *O*, *P*, *Q* into integer series by Eq. (6).

3: Transform the red, green, blue components of the plaintext images into sequences *R_1*, *G_1*, *B_1*.

4: Sort arrays *O*, *P*, *Q*, and get the corresponding index vectors *O_1*, *P_1*, *Q_1*. Arrange arrays *R_1*, *G_1*, *B_1* by *O_1*, *P_1*, *Q_1* to get *R_2*, *G_2*, *B_2*.

5: *R_2* is operated on by Eqs. (8) and (9) to obtain the sequence *CR*.

6: Manipulate *B_2* to generate sequence *CB* according to Eqs. (11) and (12).

7: Process array *G_2* using Eqs. (10) and (13) to get sequence *CG*.

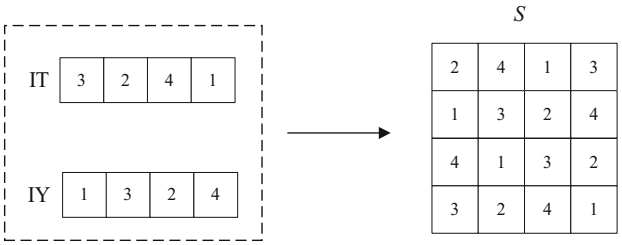

Fig. 3 Illustration of generating matrix S

according to the elements in IZ, finally two matrices S and W are obtained.

An example of using the sequences IT and IY to generate matrix S is shown in Fig. 3. Initialize each row in matrix S to $\{3, 2, 4, 1\}$, the first element in sequence IY is 1, one shifts the first row of matrix S to the left by 1 pixel, the first row of the altered matrix S is obtained, and it is $\{2, 4, 1, 3\}$. The second element in sequence IY is 3, one shifts the second row of matrix S to the left by 3 pixels, the second row of the altered matrix S is obtained and denoted as $\{1, 3, 2, 4\}$. The third element in sequence IY is 2, one moves the third row of matrix S to the left by 2 pixels, the third row of the modified matrix S is obtained and represented as $\{4, 1, 3, 2\}$. The last value in sequence IY is 4, one shifts the last row of matrix S to the left by 4 pixels, then the last row of the altered matrix S is gotten and shown as $\{3, 2, 4, 1\}$.

Step 4 An integer matrix E of size $L^2 \times L^2$ is generated by

$$E(i,\ j) = \mathrm{mod}(S(i,\ j) + j,\ 3), \tag{16}$$

where $i \in [1,\ L^2]$, $j \in [1,\ L^2]$, $E(i,\ j)$ is the element of the ith row and jth column in matrix E, and $E(i,\ j) \in [0,\ 2]$.

Step 5 Set column $j = 1$. Initialize three matrices VR, VG and VB with size $M \times N$, and set VR as PR, VG as PG, VB as PB.

Step 6 Determine the target position of the pixels to move in the matrix PR according to the values in the matrix E. For the matrix PR, move the pixel located at the ith row and jth column to the $W(i,\ S(i,\ j))th$ position of the $S(i,\ j)th$ block of the matrix VR when $E(i,\ j) = 0$; shift the pixel located at the ith row and jth column to the $W(i,\ S(i,\ j))th$ position of the $S(i,\ j)th$ block of the matrix VG when $E(i,\ j) = 1$; move the pixel located at the ith row and jth column to the $W(i,\ S(i,\ j))th$ position of the $S(i,\ j)th$ block of the matrix VB when $E(i,\ j) = 2$. The position of the blocks and the pixels within the block are arranged in row priority, this detail is as shown in Fig. 4.

Step 7 Compute the target position of the elements to move in the matrix PG according to the values in the matrix E. For the matrix PG, move the pixel located at the ith row and jth column to the $W(i,\ S(i,\ j))th$ position of the $S(i,\ j)th$ block of the matrix VR when $\mathrm{mod}(E(i,\ j) + 1,\ 3) = 0$;

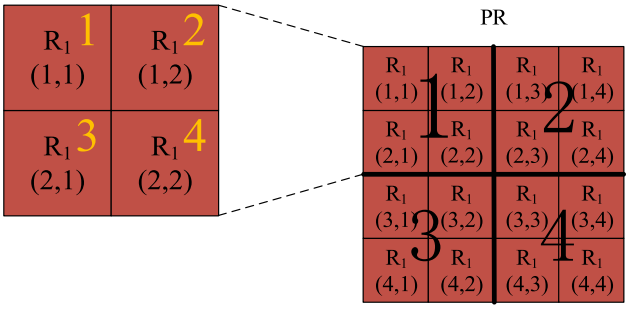

Fig. 4 Arrangement of the pixels within the block

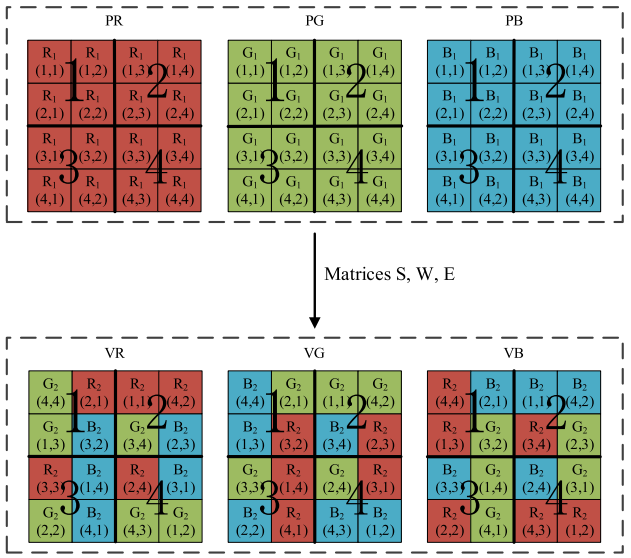

Fig. 5 Illustration of determining the target position of each pixel movement according to three matrices S, W, E

shift the pixel located at the ith row and jth column to the $W(i,\ S(i,\ j))th$ position of the $S(i,\ j)th$ block of the matrix VG when $\mathrm{mod}(E(i,\ j) + 1,\ 3) = 1$; move the pixel located at the ith row and jth column to the $W(i,\ S(i,\ j))th$ position of the $S(i,\ j)th$ block of the matrix VB when $\mathrm{mod}\,(E(i,\ j) + 1,\ 3) = 2$. The position of the block and pixels within the block are arranged in line priority (Fig. 4).

Step 8 Calculate the target position of the pixels to move in the matrix PB according to the values in the matrix E. For the matrix PB, move the pixel located at the ith row and jth column to the $W(i,\ S(i,\ j))th$ position of the $S(i,\ j)th$ block of the matrix VR when $\mathrm{mod}(E(i,\ j) + 2,\ 3) = 0$; shift the pixel at the ith row and jth column to the $W(i,\ S(i,\ j))th$ position of the $S(i,\ j)th$ block of the matrix VG when $\mathrm{mod}(E(i,\ j) + 2,\ 3) = 1$; move the pixel at the ith row and jth column to the $W(i,\ S(i,\ j))th$ position of the $S(i,\ j)th$ block of the matrix VB when $\mathrm{mod}\,(E(i,\ j) + 2,\ 3) = 2$. The position of the block and pixels within the block are arranged in row priority.

In steps 6 ~ 8, where i is an integer, $i \in [1,\ L^2]$, $j \in [1,\ L^2]$, $E(i,\ j)$ is the element at the ith row and jth column in

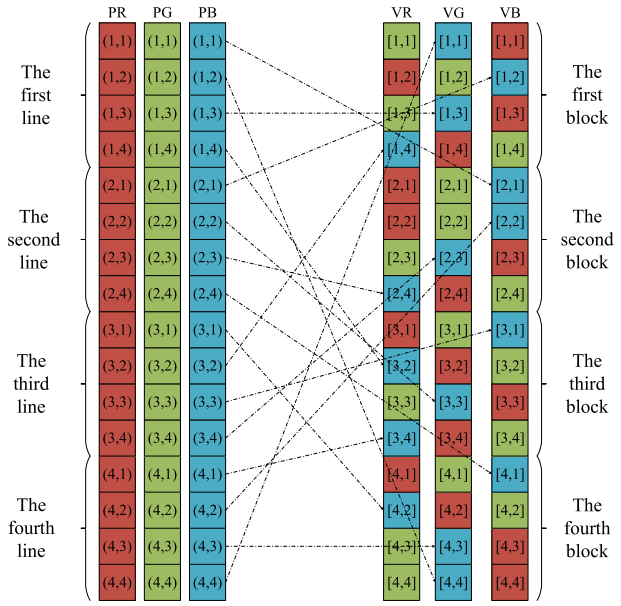

Fig. 6 Illustration of the permutation process

matrix E, $E(i,\ j) \in [0,\ 2]$, $S(i,\ j) \in \left[1,\ L^2\right]$, and $W(i,\ j) \in \left[1,\ L^2\right]$.

Step 9 Repeat steps 6 ~ 8 for $j = 2 \sim L^2$ to achieve the permutation operation. Finally, three matrices *VR*, *VG*, *VB* with size $M \times N$ are obtained. To make the reader understand the BCPIV well, an example shown in Fig. 5 is presented to determine the target position of each pixel movement according to three matrices S, W, E. Figure 6 gives another expression of the permutation process, are red, green, blue components before the permutation are given on the left, expanded by the row, components after the permutation are put on the right, expanded by blocks. The pseudocode of BCPIV is described in Algorithm 2.

BCPIV may achieve the cross-component permutation between the red, green, blue components of the plaintext images. The random sequences are sorted in ascending order to get the corresponding index vectors which are employed to generate three new matrices, and then three matrices are utilized to determine the target position of the current pixel to be shifted, which allows pixels to be scrambled across components. It makes the process of changing the position of pixels more random and improves resistance of the proposed CIE-LSCP to statistical attacks.

Algorithm 2 The pseudocode of BCPIV.

Input: Three sequences *CR*, *CG*, *CB* of length *MN*, three random sequences *O*, *P*, *Q*.

Output: Scrambled results *VR*, *VG*, *VB* with size *M*×*N*.

1: Calculate L, $L \leftarrow \min\left\{\left\lfloor\sqrt{M}\right\rfloor, \left\lfloor\sqrt{N}\right\rfloor\right\}$.

2: Generate four chaotic sequences: *T*, *H*, *Y*, *Z* according to three sequences *O*, *P*, *Q*;

3: Sort *T*, *H*, *Y*, *Z*, and get the corresponding index vectors *IT*, *IH*, *IY*, *IZ*.

4: Initialize *S*, *W*, *E* based on index vectors *IT*, *IH*, *IY*, *IZ*, $S \in N^{L^2 \times L^2}$, $W \in N^{L^2 \times L^2}$, $E \in N^{L^2 \times L^2}$.

5: for $k = 0$ to 2 do

for $j = 1$ to L^2 do

for $i = 1$ to L^2 do

$a \leftarrow S(i,j)$; $b \leftarrow W(i,j)$;

$m_1 \leftarrow \text{floor}((a-1)/L)$; $n_1 \leftarrow \text{mod}((a-1), L)$;

$m_2 \leftarrow \text{floor}((b-1)/L)+1$; $n_2 \leftarrow \text{mod}((b-1), L)+1$;

$x \leftarrow m_1 \times L + m_2$; $y \leftarrow n_1 \times L + n_2$;

$z \leftarrow \text{mod}(E(i,j)+k, 3)$;

$VR(x,y), VG(x,y), VB(x,y) \leftarrow PR(i,j), PG(i,j), PB(i,j)$;

/* When moving pixels, the target component is determined by *z*, the target row by *x*, and the target column by *y*, the starting component is decided by *k*, the starting row by *i*, the starting column by *j*. */

end for

end for

end for

Multi stochastic diffusion based on random sequences (MSDRS)

The permutated images are diffused by MSDRS to upgrade the encryption effect. The detailed diffusion process is described as follows:

Step 1 Three random sequences CO, CP and CQ with the length of MN are generated from the chaotic sequences O, P and Q by

$$\begin{cases} CO_i = \mathrm{mod}\left(floor\left(|O_i + P_i - Q_i| \times 10^{10}\right), 256\right) \\ CP_i = \mathrm{mod}\left(floor\left(|P_i + Q_i - CO_i| \times 10^{10}\right), 256\right) \\ CQ_i = \mathrm{mod}\left(floor\left(|CO_i + Q_i - CP_i| \times 10^{10}\right), 256\right) \end{cases}, \tag{17}$$

where $|x|$ is the absolute value of x, CO_i is the ith numerical element in the sequence CO, and i is an integer, $i \in [1, M \times N]$.

Three corresponding index vectors XD, SD and CD are generated by arranging CO, CP and CQ in ascending order. Meanwhile, the sequence K is generated by

$$K_i = \mathrm{mod}(|CO_i + CQ_i + CP_i|, 256), \tag{18}$$

where K_i is the ith numerical element in the sequence K, and i is an integer, $i \in [1, M \times N]$.

Step 2 The three matrices VR, VG and VB are converted into one-dimensional arrays VRR, VGG and VBB with length MN, respectively. Three sequences VRR, VGG and VBB were diffused using XD, SD, CD and K, and sequences RA, GA and BA with the length of MN are obtained by

$$\begin{cases} RA(XD(i)) = VRR(SD(i)) \oplus K(CD(i)) \oplus VGG(SD(i)) \\ GA(XD(i)) = VGG(SD(i)) \oplus K(CD(i)) \oplus VBB(SD(i)), \\ BA(XD(i)) = VBB(SD(i)) \oplus K(CD(i)) \end{cases} \tag{19}$$

where $XD(i)$ is the ith numerical element in the sequence XD, and i is an integer, $i \in [1, M \times N]$.

Step 3 Three matrices CIR, CIG and CIB of size $M \times N$ are gotten by transforming three sequences RA, GA and BA, respectively. Subsequently, combine CIR, CIG and CIB to obtain ciphertext image, and the encryption process is completed.

In the MSDRS, three index vectors corresponding to random sequences are generated. According to the index vectors, one pixel in the current component, one pixel in other component and one element in a random sequence are selected to participate in XOR operation, and then the obtained pixel is placed at a random position among the current component. MSDRS can achieve a more stochastic process that enhances the security of CIE-LSCP. The pseudocode of MSDRS is described in Algorithm 3.

Algorithm 3 The pseudocode of MSDRS.

Input: Three matrices VR, VG, VB with size $M \times N$, three random sequences O, P, Q with length MN.

Output: Three matrices CIR, CIG, CIB of size $M \times N$.

1: Generate four random sequences CO, CP, CQ, K according to Eqs. (17) and (18);

2: Sort CO, CP, CQ, and get the corresponding index vectors XD, SD, CD.

3: for $i = 1$ to MN do

Sequences RA, GA, BA are obtained by applying Eq. (19);

end for

4: Matrices CIR, CIG, CIB are produced by recombining RA, GA, BA.

Decryption algorithm

The decryption steps of this proposed algorithm are the reverse process of images encryption, the ciphertext image and keys are used as the input of the decryption algorithm, and the decrypted image is output. The specific flowchart shown in Fig. 7 are described as follows. The decryption is completed by three inverse processes, namely, the inverse MSDRS, the inverse BCPIV, and the inverse PSLS. The RGB three-component in the ciphertext image is transformed into three sequences RA, GA and BA, then three sequences VRR, VGG and VBB are obtained by Eq. (20), and next three sequences CR,CG and CB are generated by the inverse BCPIV process, subsequently, three sequences R_2, G_2, B_2 are produced by Eqs. (21–26), finally the three sequences obtained are recombined to obtain the color decrypted image.

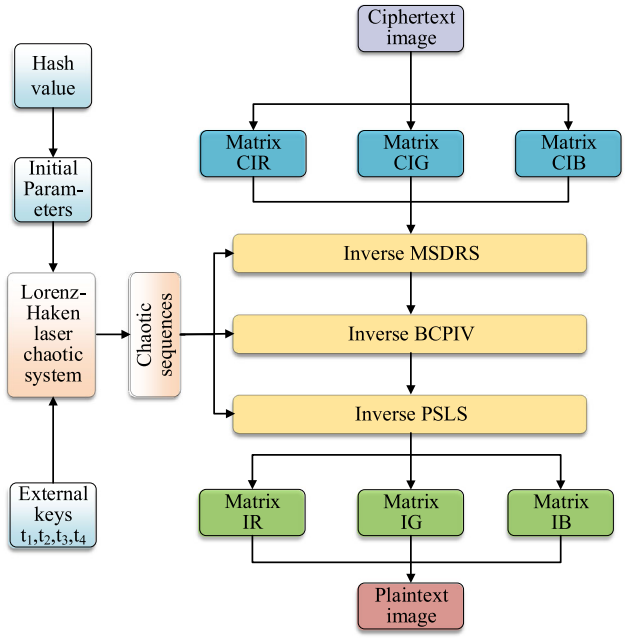

Fig. 7 The flowchart of the proposed decryption process

$$\begin{cases} VBB(SD(i)) = BA(XD(i)) \oplus K(CD(i)) \\ VGG(SD(i)) = GA(XD(i)) \oplus K(CD(i)) \oplus VBB(SD(i)), \\ VRR(SD(i)) = RA(XD(i)) \oplus K(CD(i)) \oplus VGG(SD(i)) \end{cases} \tag{20}$$

$$NG_i = \mathrm{mod}(CG_i + 256 - CB_{i-1} \oplus CB_i, 256), \tag{21}$$

$$NB_i = \mathrm{mod}\left(floor\left(\frac{1}{2}(CR_i + \mathrm{mod}(Q_i, 256))\right), 256\right) \oplus CB_i, \tag{22}$$

$$B\_2_i = \mathrm{mod}\left(NB_i + 256 - floor\left(\frac{1}{2}(CR_{i-1} + CR_i)\right), 256\right), \tag{23}$$

$$G\_2_i = \mathrm{mod}(NG_i + 256 - CR_{i-1} \oplus CR_i, 256), \tag{24}$$

$$NR_i = \mathrm{mod}\left(floor\left(\frac{1}{2}(G\_2_i + \mathrm{mod}(P_i, 256))\right), 256\right) \oplus CR_i, \tag{25}$$

$$R\_2_i = \mathrm{mod}\left(NR_i + 256 - floor\left(\frac{1}{2}(G\_2_{i-1} + G\_2_i)\right), 256\right), \tag{26}$$

Simulation results and performance analyses

Simulation results are given to verify the security and effectiveness of the proposed CIE-LSCP in this section. The experiments are executed on a computer, the environments are as follows: Intel(R) Core (TM) i7-6700 CPU @ 3.40 GHz (8 CPUs). All the experiments are tested on Windows 10 operating system with MATLAB 2018a. Four color images (Lena, Baboon, Sailboat, Peppers) in Table 1 are selected to perform the encryption and decryption processes. Figure 8 illustrates the plaintext images (a–d), the homologous ciphertext images (e–h), and the decrypted images (i–l). Besides, to fully demonstrate the availability of CIE-LSCP, eight plaintext images in Fig. 9a–h taken from the USC-SIPI (http://sipi.usc.edu/database/) images database are adopted as test images for the following analyses. Figure 9i–p and q–x are the corresponding encrypted and decrypted images.

Table 1 Images for testing

Image	Size ($h \times w \times c$)	Image	Size ($h \times w \times c$)
Lena	$256 \times 256 \times 3$	Sailboat	$256 \times 256 \times 3$
Baboon	$256 \times 256 \times 3$	Peppers	$256 \times 256 \times 3$

As shown in Figs. 8 and 9, the encrypted images are absolutely dissimilar to plaintext images, we cannot obtain any information related to the original images from them, the decrypted images are indistinguishable from primitive images. Then we can conclude that the proposed CIE-LSCP possesses satisfactory image cryptosystem effect.

Key space

For image encryption algorithms, the size of the key space has a direct impact on their security, the brute-force attacks can be withstood proficiently when the key space reaches 2^{100}. The more the number of the key, the larger its key space is. In CIE-LSCP, the keys consist of: (1) 256-bit hash value generated from original image; (2) Parameters of chaotic system σ, γ, b and the external keys t_1, t_2, t_3, t_4. Assuming the computation precision of the computer is 10^{-14}, the key space of CIE-LSCP is $2^{256} + (10^{14})^7 \approx 2^{325} > 2^{100}$, so our proposed scheme can resist brute-force attack effectively. Besides, comparison results with other algorithms are listed in Table 2. It is evident that our method has the largest key space than algorithms in Refs. [15, 17, 24, 27, 31].

Quality metrics analysis

Mean squared error (MSE) and Peak signal to noise ratio (PSNR) are usually applied to measure the pixel difference between two images. The MSE value represents the average error value of two images, wherein the larger the MSE of the encrypted image, the greater the pixel difference between the ciphertext image and the plaintext image, and the smaller the MSE value of the decrypted image, the more superior the decryption of the encrypted image. There is an inverse relationship between the values of MSE and

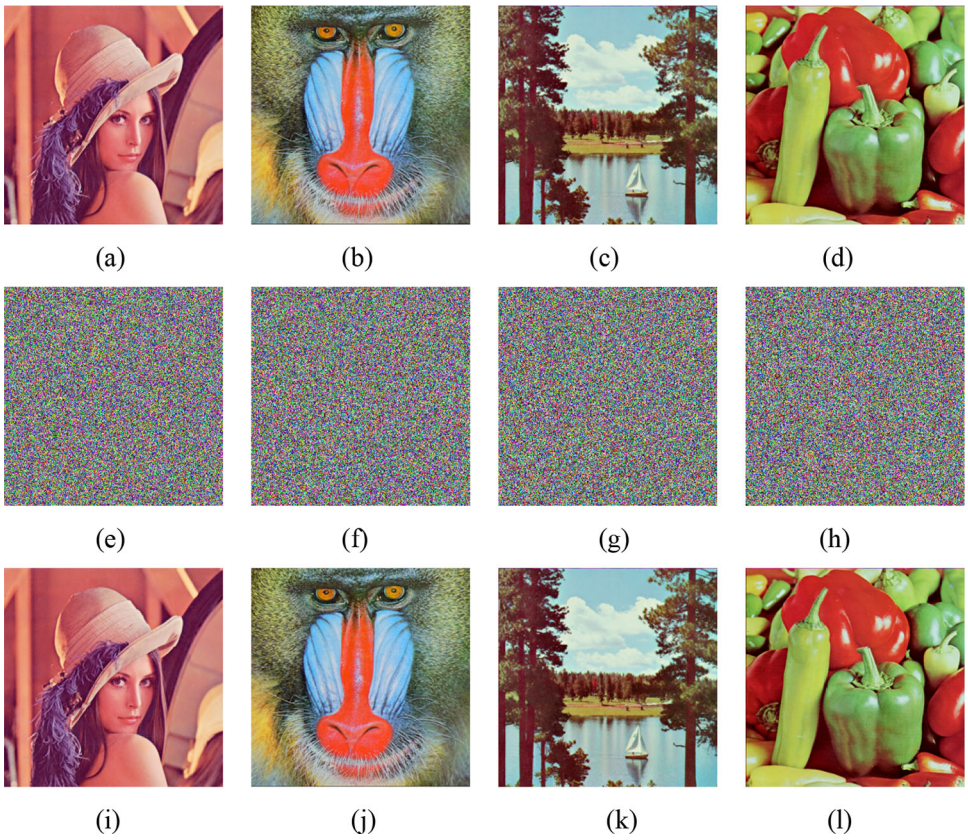

Fig. 8 Some results of encryption algorithm. **a**–**d** Original images; **e**–**f** Ciphertext images; **i**–**l** Decryption images

PSNR, that is, the smaller the MSE value, the larger the PSNR value. We tested the MSE and the PSNR metrics [41] for the encrypted and decrypted images. Since we can decrypt the plaintext images completely from the encrypted images, making the decrypted images and the plaintext images identical, the MSE results of the decrypted images obtained from the test are all 0 and the PSNR results are all $+\infty$. We present the outcomes of MSE and PSNR metrics for the encrypted images in Table 3.

$$MSE = \begin{cases} \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [P(i,\ j) - I(i,\ j)]^2,\ for\ gray - scale\ image \\ \frac{1}{M \times N \times 3} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [P(i,\ j) - I(i,\ j)]^2,\ for\ color\ image \end{cases}, \tag{27}$$

$$PSNR = 10 \log \frac{255^2}{\left(\frac{1}{MN}\right) \sum_{i=1}^{M} \sum_{j=1}^{N} [D(i,\ j) - I(i,\ j)]^2}, \tag{28}$$

where P is a plaintext image and I is a decrypted image, $P(i, j)$ is the element of the ith row and jth column in matrix P, M is the row number of the plaintext images, and N is the column number of the plaintext images.

Histogram analysis

The histogram of the image shows how pixels of different channels are distributed, and the information quantity in the image can be evaluated to determine the resistance of the encryption algorithm to statistical attacks. Generally, the histograms of the ciphertext image need to be completely distinct from those of plaintext image. The histograms of original and ciphertext images are illustrated in Fig. 10. The histogram of original images shown in Fig. 10a–d, their distribution is not uniform. The rest show the histogram of corresponding ciphertext images, their distribution is flat. The histogram distribution shows that the proposed CIE-LSCP can be effective in hiding the pixel distribution information of plaintext image and withstand statistical attack well.

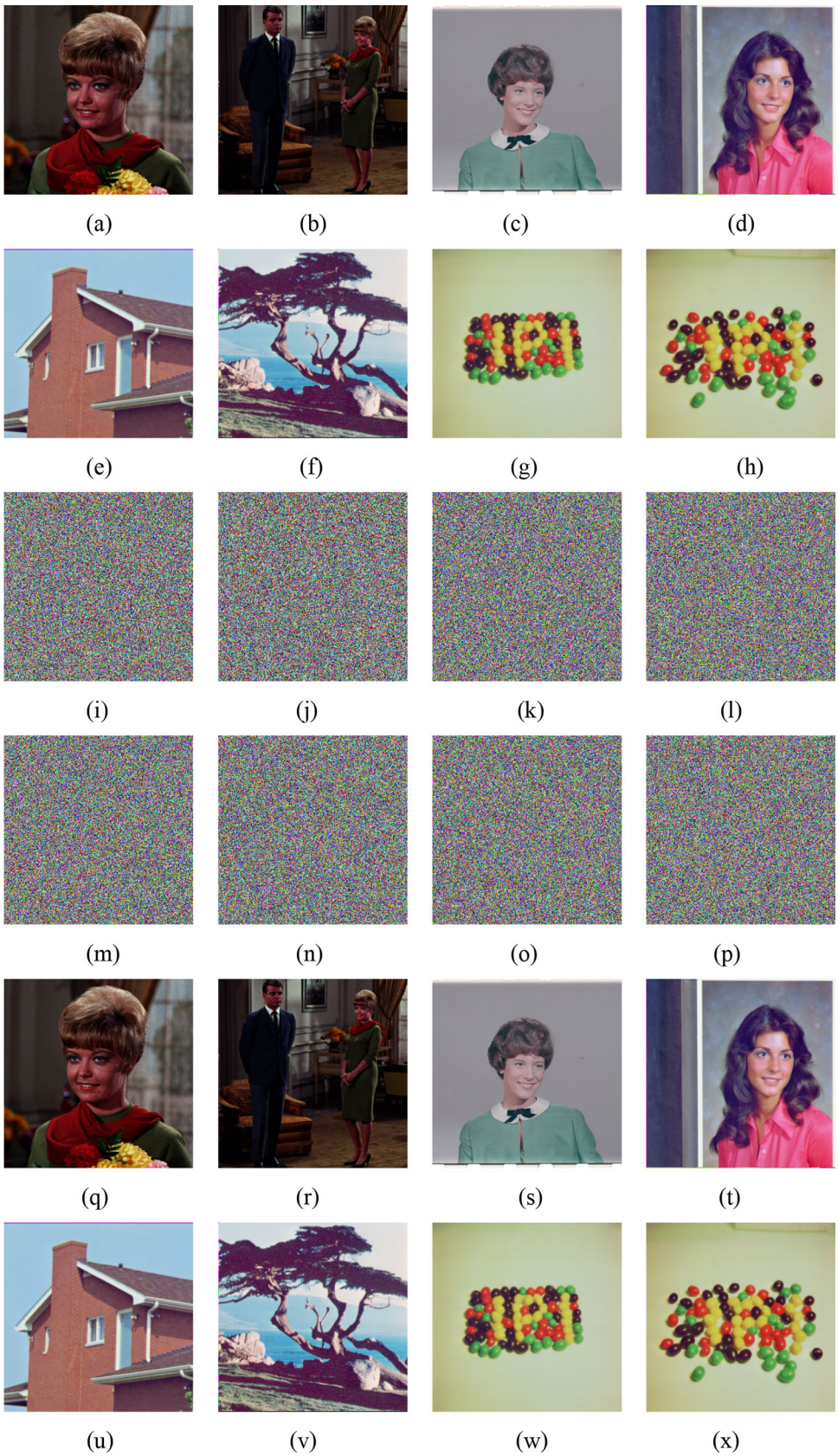

Fig. 9 Encryption and decryption results of eight test images. **a–h** Original images; **i–p** Ciphertext images; **q–x** Decryption images

Table 2 Comparison of key space with other algorithms

Algorithm	Ours	Ref. [15]	Ref. [17]	Ref. [24]	Ref. [27]	Ref. [31]
Key space	2^{325}	2^{318}	2^{203}	2^{300}	2^{138}	2^{312}

Table 3 MSE and PSNR results

Image	Component	MSE	PSNR (dB)	Image	Component	MSE	PSNR (dB)
Lena	R	10,652	7.8566	Sailboat	R	7239	9.5339
	G	9032	8.5729		G	11,343	7.5834
	B	7059	9.6436		B	11,404	7.5603
Baboon	R	8383	8.8967	Peppers	R	1975	9.1136
	G	7409	9.4330		G	11,183	7.6452
	B	9154	8.5149		B	11,102	7.6768
Ref. [23]	R	10,697	7.8378	Ref. [35]	R	7927	9.1395
	G	8968	8.6035		G	7307	9.4931
	B	7042	9.6540		B	9712	8.2573

The variances of histogram are used to assess the uniformity of the plaintext and encrypted images, whereby a smaller variance value indicates a more uniform image [42]. The test results of histogram variance are illuminated in Table 4. By contrasting the variance of plaintext and ciphertext image, we can find that the variance of ciphertext image is decreased significantly relative to that of original image, and the result of variance of ciphertext image is less than 285. The chi-square test outcomes are revealed in Table 5, from which we can see that the data results of each component of the color image pass the chi-square test, and the maximum test value is not more than 285. It is proved that the proposed CIE-LSCP could withstand statistical attacks significantly. The chi-square test [43] can be calculated by

$$x^2_{\text{test}} = \sum_{k=1}^{n} \frac{(c_k - v_k)}{v_k}, \tag{29}$$

where n is gray level ($n = 256$ in this test), c_k and v_k are the occurrence frequency and expected value of gray level k, respectively.

Correlation analysis of adjacent pixels

Neighboring pixels in a plaintext image usually have high correlation, and a feasible image encryption method can reduce the correlation coefficient of adjacent pixels in a ciphertext image to an admissible range. The absolute value between adjacent pixels of the ciphertext image should be as close to 0 as possible. The values of the correlation coefficient of original images is greater than 0.78 in Table 6, which indicates that there is strong correlation between adjacent pixels of plaintext images in all directions, while the correlation coefficient of ciphertext image is less than 0.008, indicating that the correlation of ciphertext image is very weak. Compared to some other approaches, CIE-LSCP leads to reduced correlation, thus CIE-LSCP has a better mess effect.

The correlation test results of adjacent pixels of three components of plaintext image and ciphertext image are elucidated in Fig. 11. As can be seen from it that the adjacent pixels of plaintext images in the horizontal, vertical and diagonal directions are mostly concentrated, this is due to the fact that the plaintext image carries a large amount of information, resulting in strong correlation between adjacent pixels, while our proposed encryption scheme can shift a pixel in the image to an arbitrary target location, which can increase the difference between adjacent pixel values and weaken the correlation between adjacent pixels and R, G, and B components, making the pixel distribution of the ciphertext image more uniform.

The correlation coefficients of adjacent pixels in plaintext image and the corresponding ciphertext image are calculated by Eqs. (30, 31, 32) [44],

$$CC = \frac{E((x - E(x))(y - E(y)))}{\sqrt{D(x)D(y)}}, \tag{30}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{31}$$

$$D(x) = \frac{1}{N}\sum_{i-1}^{N} (x_i - E(x))^2, \tag{32}$$

where x, y are the values of two neighboring pixels in the image, N is the pixel count chosen from the image, $E(x)$ and $D(x)$ are the expectation and variance of x, respectively.

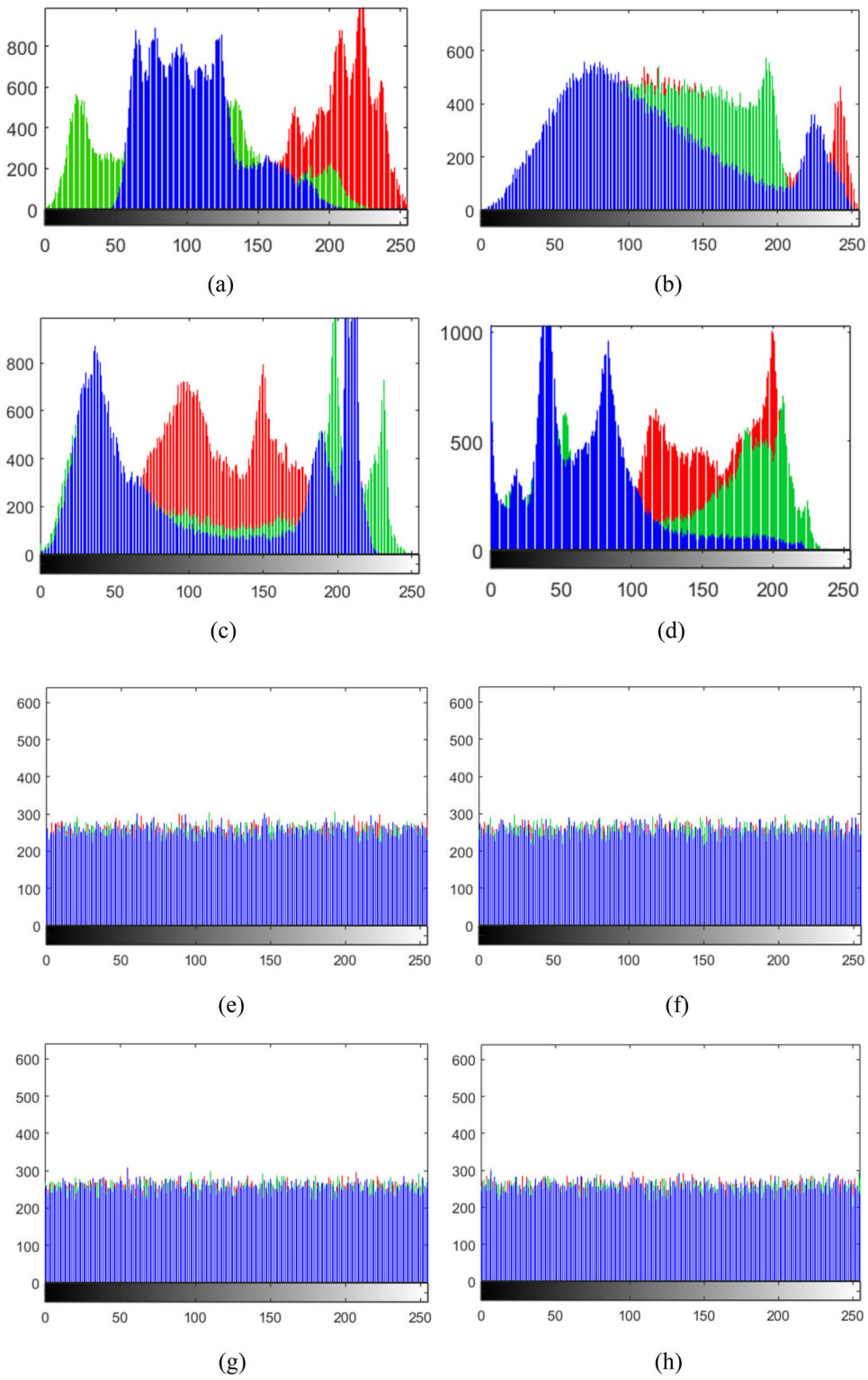

Fig. 10 Histograms of plaintext and ciphertext images

Table 4 Histogram variance of plaintext and ciphertext images

Image	Plaintext images			Ciphertext image		
	R	G	B	R	G	B
Lena	6.4731e+04	2.9669e+04	9.1621e+04	260.0784	271.2471	278.6824
Peppers	5.6977e+04	5.4691e+04	1.0462e+05	269.0039	279.8824	280.3294
Baboon	2.8194e+04	4.2050e+04	2.5960e+04	280.9412	284.6510	253.1373
Sailboat	5.3726e+04	3.5834e+04	9.1551e+04	215.2157	256.8863	215.3412
4.1.01	1.6962e+05	1.5857e+05	1.5866e+05	253.8745	254.8314	260.0549
4.1.02	2.1119e+05	3.3918e+05	2.9077e+05	253.6627	233.3490	250.1020
4.1.03	7.9393e+05	8.6434e+05	6.2281e+05	268.0000	246.9098	252.6196
4.1.04	6.7232e+04	6.4687e+04	1.1349e+05	274.4235	276.1412	277.8902
4.1.05	2.5959e+05	3.0033e+05	3.9558e+05	249.3725	242.2118	270.1176
4.1.06	8.1690e+04	5.7233e+04	1.3033e+05	267.2157	255.7647	266.8627
4.1.07	7.9309e+05	5.0363e+05	1.6874e+05	241.6078	279.9843	268.8627
4.1.08	5.3961e+05	3.5062e+05	1.3027e+05	246.0314	254.5569	244.0706

Table 5 Results of chi-square test of ciphertext images

Image	x^2_{test}			$x^2_{256,0.05}$	Result
	R	G	B		
Lena	259.0625	270.1875	277.5938	293	Pass
Peppers	267.9531	278.7891	279.2344	293	Pass
Baboon	279.8438	283.5391	252.1484	293	Pass
Sailboat	214.3750	255.8828	214.5000	293	Pass
4.1.01	252.8828	253.8359	259.0391	293	Pass
4.1.02	252.6719	232.4375	249.1250	293	Pass
4.1.03	266.9531	245.9453	251.6328	293	Pass
4.1.04	273.3516	275.0625	276.8047	293	Pass
4.1.05	248.3984	241.2656	269.0625	293	Pass
4.1.06	266.1719	254.7656	265.8203	293	Pass
4.1.07	240.6641	278.8906	267.8125	293	Pass
4.1.08	245.0703	253.5625	243.1172	293	Pass

Information entropy analysis

Information entropy can be applied to evaluate the randomness and unpredictability of an image. The value of information entropy is calculated by [44]

$$H(m) = \sum_{i=0}^{2^N-1} p(m_i) \log \frac{1}{p(m_i)}, \tag{33}$$

where m represents an information source, $p(m_i)$ indicates potentiality of m_i. In our scheme, the random sequences resulting from the 3-D chaotic system are utilized to permute the preprocessed matrix. The conversion of pixel positions across the three components is effectively implemented, and also enhances the uncertainty and disorder of pixels in ciphertext images. From Table 7, we can get that the information entropy of the red, green, blue components in ciphertext image is above 7.99, close to the theoretical value of 8, which indicates that the ciphertext image gotten by our algorithm has remarkable random attribution and can resist strong statistical attacks and entropy attacks.

Key sensitivity analysis

The encryption key sensitivity analysis

This part uses a key that is subtly dissimilar to the correct key to encrypt the original image, then tests the sensitivity of the key in the encryption process. The correct key set K_0 contains the 256-bit hash value of the plaintext images, the parameters σ, γ, b and external keys t_1, t_2, t_3, t_4. Change the external key t_1, t_2, t_3, t_4 in key set K_0 by the amount of Δ

Table 6 Correlation coefficients of adjacent pixels in plaintext images and ciphertext image

Image		Horizontal	Vertical	Diagonal
Plaintext images (Lena)	R	0.9760	0.9535	0.9281
	G	0.9687	0.9376	0.9089
	B	0.9516	0.9217	0.9006
Ciphertext image (Lena)	R	− 0.0008	− 0.0004	− 0.0073
	G	0.0009	0.0025	− 0.0046
	B	− 0.0012	− 0.0018	0.0001
Ciphertext image (Baboon)	R	− 0.0006	0.0014	0.0029
	G	0.0022	− 0.0017	0.0007
	B	0.0058	0.0051	− 0.0009
Ciphertext image (Sailboat)	R	− 0.0042	0.0010	0.0045
	G	0.0007	− 0.0062	0.0024
	B	0.0036	− 0.0003	− 0.0038
Ciphertext image (Peppers)	R	− 0.0039	− 0.0068	0.0051
	G	− 0.0039	− 0.0008	0.0005
	B	0.0043	− 0.0007	0.0021
Ref. [17]	R	0.0137	− 0.0237	0.0109
	G	− 0.0246	− 0.0170	− 0.0133
	B	− 0.0137	0.0023	− 0.0013
Ref. [18]	R	− 0.0047	0.0028	− 0.0043
	G	− 0.0023	− 0.0060	− 0.0069
	B	− 0.0038	− 0.0057	− 0.0112
Ref. [24]	R	0.0083	− 0.0054	− 0.0010
	G	0.0049	0.0100	0.0124
	B	− 0.0017	0.0095	− 0.0042
Ref. [30]	R	− 0.0141	− 0.0036	0.0006
	G	− 0.0091	− 0.0074	0.0037
	B	− 0.0094	− 0.0282	− 0.0049

($\Delta = 10^{-14}$), only one of the parameters is altered at a time, other parameters remain unchanged, and then new key sets K_1-K_4 are obtained.

The NPCR between ciphertext image encrypted by K_0 and those encrypted by K_1–K_4 is tested and listed in Table 8. According to Table 8, when the key has been modified just a tiny amount (Δ), the pixel difference rate of different ciphertext images is close to 99.60. It can be found from the values in Table 8 that the proposed CIE-LSCP is very sensitive to the key in encryption stage.

The decryption key sensitivity analysis

This part uses K_1–K_4, which is marginally dissimilar to the correct decryption K_0, to decrypt the ciphertext image and tests the sensitivity of the key in the decryption procedure.

We test the NPCR between the image decrypted with key set K_0 and the image decrypted with the changed key K_1-K_4 quantitatively. From Table 9, one may watch that when there is a minor alteration in the key (Δ), the pixel difference rate of different decrypted images is about 99.60. The data in the table show that the proposed algorithm is sensitive to the key in decryption process.

Noise attack analysis

In the process of images transmission, they are highly likely to be attacked by the noise, which increases the difficulty to restore the plaintext images. During the simulation, we add different salt & pepper noise (SPN), speckle noise (SN) and white Gaussian noise (GN) respectively to the ciphertext image, and then decrypt them. The corresponding decryption results are shown in Fig. 12. We can still derive most of the significant information in the plaintext image from the decrypted image. Here PSNR is employed to appraise the connection between plaintext image and decrypted image. The PSNR value of two images can be computed by Eq. (28). The PSNR values of the plaintext

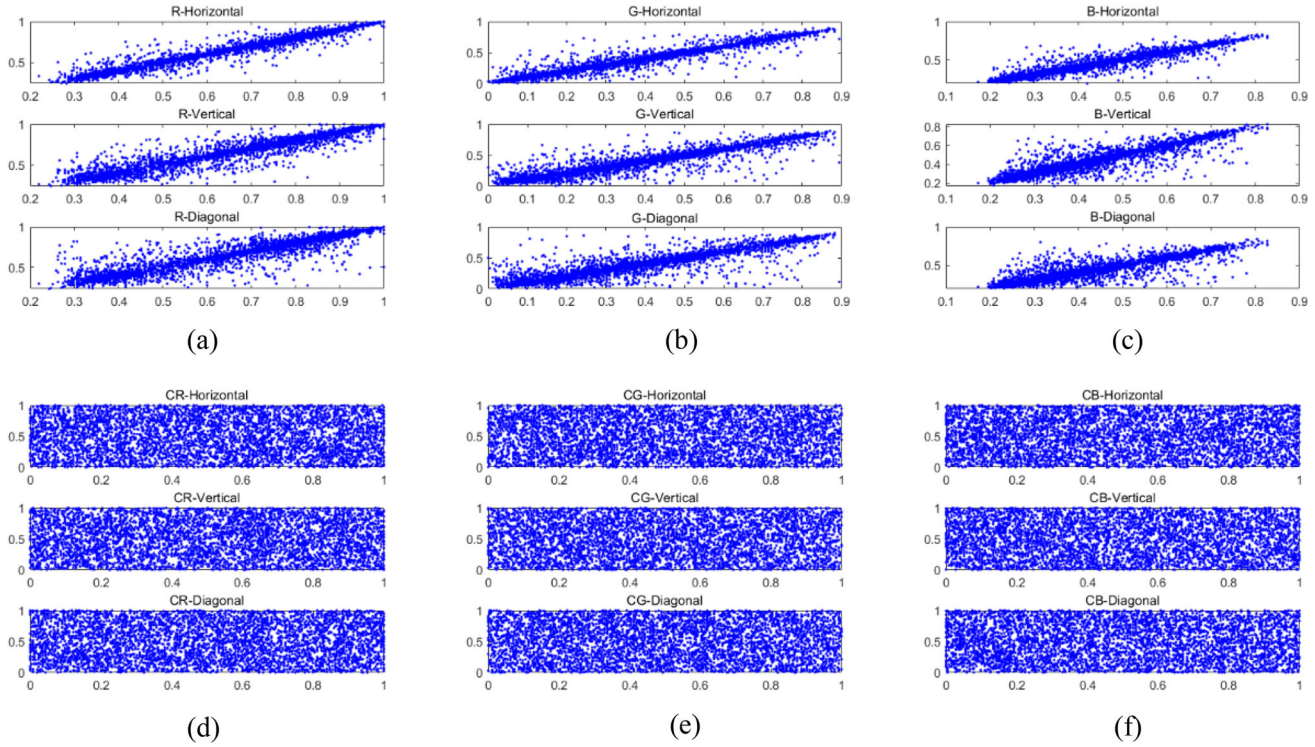

Fig. 11 Adjacent pixel distribution. **a–c** Adjacent pixel distribution of red, green and blue components of plaintext images; **d–f** Adjacent pixel distribution of red, green and blue components of ciphertext image

Table 7 Information entropy test results

Image	Red	Green	Blue
Lena	7.2509	7.5906	6.9284
Baboon	7.6160	7.3754	7.6804
Sailboat	7.2673	7.6263	7.2006
Peppers	7.3063	7.5557	7.0912
Enc_Lena	7.9969	7.9973	7.9972
Enc_Baboon	7.9973	7.9965	7.9967
Enc_Sailboat	7.9976	7.9972	7.9976
Enc_Peppers	7.9972	7.9974	7.9972
Ref. [17]	7.9892	7.9898	7.9899
Ref. [18]	7.9973	7.9965	7.9969
Ref. [24]	7.9972	7.9972	7.9975
Ref. [30]	7.9973	7.9970	7.9972

images and the decrypted images with noise are listed in Table 10, where the greater the PSNR value, the smaller the difference between plaintext image and decrypted image. With SPN attached to the ciphertext image, the noise intensity varies from 0.00001, to 0.00003, to 0.00005, to 0.00007, respectively, the PSNR value is as low as 43.1644 dB and as high as 61.5080 dB. While we add GN with noise intensity of 0.00001, 0.00003, 0.00005, 0.00007 to the ciphertext image, the PSNR value varies from a minimum of 13.8054 dB to a maximum of 16.9962 dB. We can still largely recognize the contents of the plaintext image, this suggests that CIE-LSCP has the ability to resist noise attacks.

Cropping attack analysis

The ciphertext images with different cropping attacks (a–c) and the corresponding decrypted images (d–f) are displayed in Fig. 13. Obviously, the deciphered images still contain some details of the original image, and one may get the plaintext information from them. When the ciphertext image is cropped by 1/64, the PSNR value between the decrypted image and plaintext image reaches 20.1234 dB; when the ciphertext image is cut out by 1/16, the PSNR value decreases by 5.5925 dB; when the ciphertext image is cut out by 1/4, the PSNR value drops by another 4.3789 dB. Even though some information in encrypted image is dropped, the participants are still able to roughly recover the information in the original image by the decryption algorithm. The simulation results demonstrate that the proposed CIE-LSCP has excellent robustness against cropping attack.

Running time

The time of the encryption and decryption is also an important factor to evaluate the performance of an image encryption algorithm. If the encryption and decryption speed is

Table 8 The NPCR between two ciphertext images

Key	NPCR (%)			
	Lena	Baboon	Sailboat	Peppers
$K_1(t_1' = t_1 + \Delta)$	99.57	99.62	99.64	99.59
$K_2(t_2' = t_2 + \Delta)$	99.59	99.62	99.64	99.59
$K_3(t_3' = t_3 + \Delta)$	99.64	99.60	99.62	99.57
$K_4(t_4' = t_4 + \Delta)$	99.63	99.58	99.58	99.61

Table 9 The NPCR between two deciphered images

Key	NPCR (%)			
	Lena	Baboon	Sailboat	Peppers
$K_1(t_1' = t_1 + \Delta)$	99.59	99.59	99.59	99.61
$K_2(t_2' = t_2 + \Delta)$	99.64	99.59	99.57	99.64
$K_3(t_3' = t_3 + \Delta)$	99.59	99.55	99.65	99.60
$K_4(t_4' = t_4 + \Delta)$	99.59	99.61	99.61	99.61

too slow, the image cryptosystem lacks practicality. The time to be consumed is obtained by the simulation experiment. The proposed CIE-LSCP is divided into four major parts: chaotic sequences generation, preprocessing-PSLS, scrambling-BCPIV and diffusion-MSDRS. Besides, the proposed image decryption method also includes four stages: generating chaotic sequences, PSLS, inverse BCPIV and inverse MSDRS. The specific encryption and decryption times of our CIE-LSCP for 256 × 256 color images are displayed in Table 11. As is evident from Table 11 that the encryption time and decryption time are almost equal for the proposed CIE-LSCP is a symmetric encryption method, and the encryption time is a bit more than decryption time, this may be because that some time is cost since some key parameters are produced in encryption stage and transmitted to the receiver, thus, they may be directly utilized for decryption.

Differential attack analysis

For image encryption schemes, differential attack is a common kind of attack strategy. Attackers are able to get the ciphertext image, and by changing a few pixels in the plaintext image, they attempt to retrieve the information needed to obtain new ciphertext image, attack the encryption algorithm through comparing the changes of two ciphertext images. The number of pixel change rate (NPCR) and unified average change in intensity (UACI) are often used to perform quantitative evaluation of encryption schemes. NPCR measures the pixel change rate in ciphertext image and UACI is the average of the intensity difference between ciphertext images.

The values of NPCR and UACI are determined by [16]

$$D(i,\ j) = \begin{cases} C_1(i,\ j) = C_2(i,\ j),\ \text{if } D(i,\ j) = 0 \\ \text{Otherwise, if } D(i,\ j) = 1 \end{cases}, \tag{34}$$

$$\text{NPCR} = \frac{\sum D(i,\ j)}{\text{rows} \times \text{cols}} \times 100\%, \tag{35}$$

$$\text{UACI} = \frac{1}{\text{rows} \times \text{cols}} \left(\sum_{i,\ j} \frac{C_1(i,\ j) - C_2(i,\ j)}{255} \right), \tag{36}$$

where C_1 refers to the ciphertext image obtained from the plaintext image, and C_2 refers to the ciphertext image obtained by encrypting the image that has minor differences from the plaintext image.

From the data in Table 12, it can be obtained that our encryption scheme is sensitive to the plaintext image, because the SHA256 hash values of the plaintext image are computed during the encryption process, which are applied to get the initial parameters of the chaotic system and generate chaotic sequences, and these sequences are extremely crucial to the preprocessing, permutation and diffusion processes.

Classical attack analysis

Cryptanalysis is the decryption of a ciphertext without knowing the secret information, and it is usually needed for decryption through finding the keys. When performing cryptanalysis, there are four types of attacks, depending on the level of information available to the analyst [45].

1. Ciphertext only attack: The attacker performs an exhaustive attack when only the ciphertext is known.

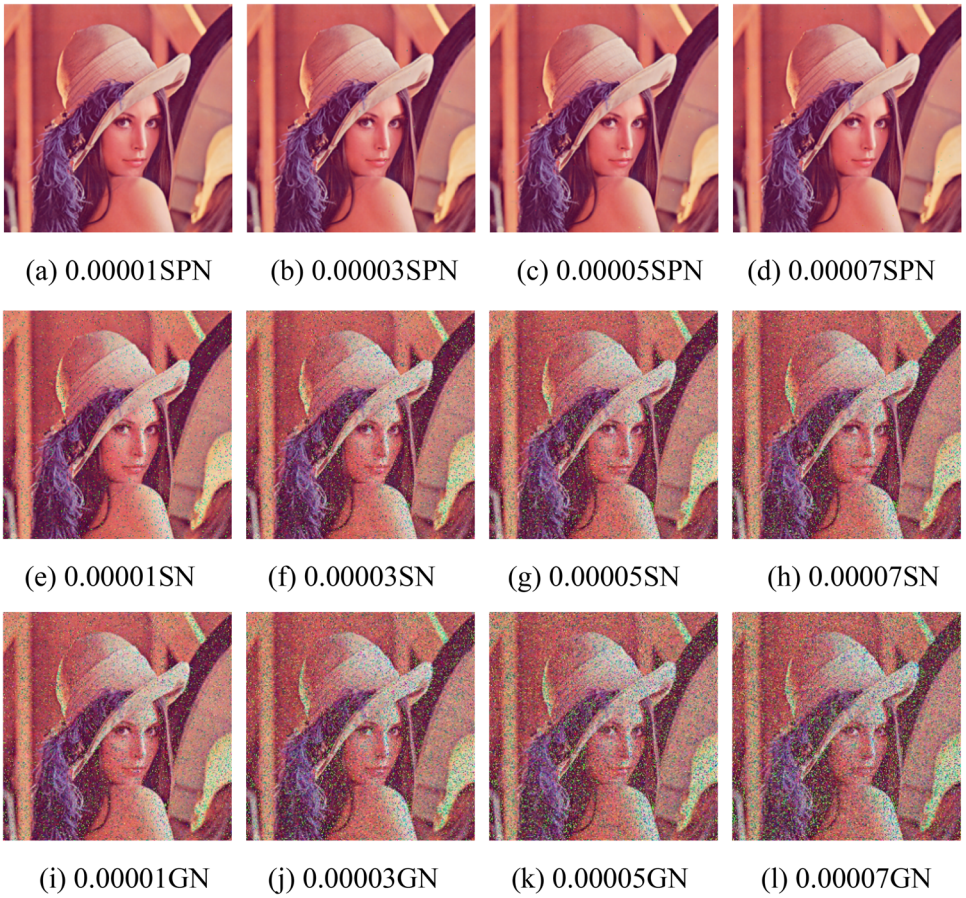

Fig. 12 The deciphered images with different noise

Table 10 The PSNR values of the plaintext images and decrypted images with noise

Noise type	Noise intensity	PSNR (dB)
SPN	0.00001	61.5080
	0.00003	43.1644
	0.00005	41.5922
	0.00007	40.3752
SN	0.00001	19.8482
	0.00003	17.2376
	0.00005	16.3590
	0.00007	15.7123
GN	0.00001	16.9962
	0.00003	15.1184
	0.00005	14.2924
	0.00007	13.8054

2. Known plaintext attack: The attacker has some given plaintext-ciphertext pairs and uses them to derive the encryption algorithm and keys.
3. Chosen plaintext attack: The attacker can select some plaintexts and obtain the corresponding ciphertexts with the goal of launching the key.
4. Chosen ciphertext attack: An attacker can construct the plaintext corresponding to any ciphertext with the goal of launching the key.

Among them, the chosen-plaintext attack is the most powerful type of attack, and as long as the encryption algorithm can resist the chosen-plaintext attack, it can resist other types of attacks. In the encryption scheme proposed in this paper, three random sequences are generated based on plaintext information and an external key for the encryption process. Any error in one of the initial parameters will produce a completely different encrypted image, so this encryption scheme can resist the chosen-plaintext attack and also the other three types of attacks mentioned above.

NIST test

In many cryptographic applications, random and pseudo-random numbers are required. Here the randomness test of the sequences is discussed. First, the sequences to be tested are transformed into a sequence of binary numbers and then

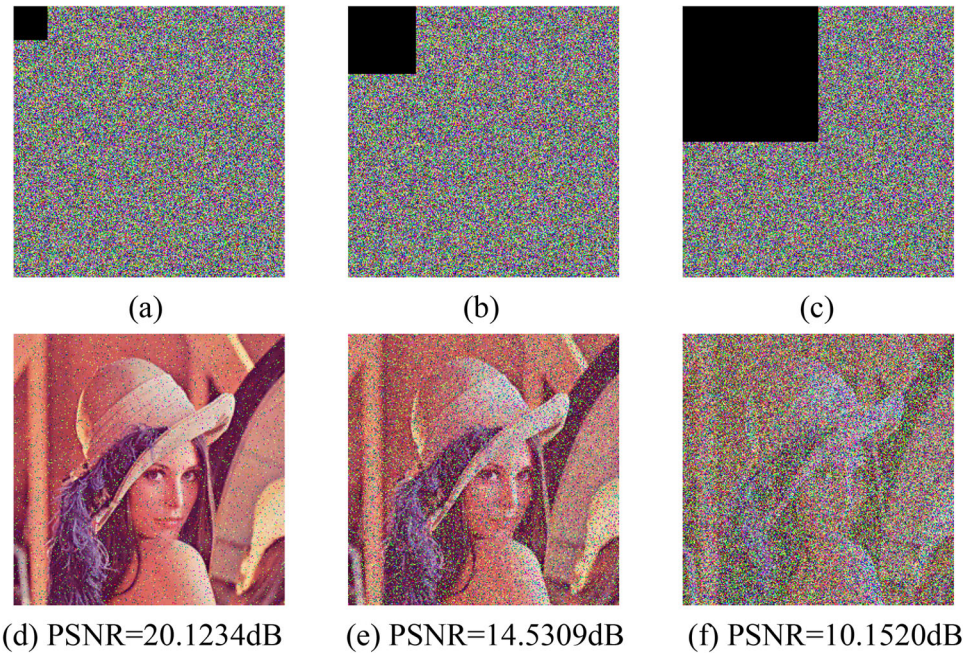

Fig. 13 The results of cropping attack

Table 11 Encryption and decryption time for 256 × 256 color images

Images	Lena	Baboon	Sailboat	Peppers	Average
Encryption	1.7715 s	1.7836 s	1.7229 s	1.7310 s	1.7523 s
Decryption	1.5159 s	1.5403 s	1.5075 s	1.4754 s	1.5098 s

Table 12 The NPCR and UACI between two ciphered images

Image		NPCR (%)	UACI (%)
Ciphertext image (Lena)	R	99.64	33.55
	G	99.62	33.43
	B	99.62	33.44
Ciphertext image (Baboon)	R	99.59	33.52
	G	99.59	33.25
	B	99.62	33.36
Ciphertext image (Sailboat)	R	99.56	33.52
	G	99.63	33.49
	B	99.57	33.42
Ciphertext image (Peppers)	R	99.62	33.54
	G	99.57	33.65
	B	99.61	33.36

the testing software of National Institute of Standards and Technology (NIST) is employed to conduct randomness tests on the sequences, containing a total of 15 items of measurement indexes, the probability is used to characterize the properties of the random sequences, specifically denoted as P-value. For each test indicator, a P-value is generated, where if the P-value is greater than or equal to 0.001, the test passes, otherwise the test does not pass.

We tested the randomness of the three random sequences which were generated by the Lorenz–Haken laser chaotic system as well as the sequences of encrypted images. The specific results are shown in Tables 13 and 14, among them, the P-values are all greater than 0.001 for 15 items, indicating that the used chaotic sequences and the obtained cipher images have good randomness.

Table 13 NIST analyses for chaotic sequences

Number	Statistical test	x P value	y P value	z P value	Pass/Fail
1	Frequency	0.350485	0.739918	0.911413	Pass
2	Block frequency	0.739918	0.122325	0.350485	Pass
3	Cumulative sums	0.911413	0.534146	0.739918	Pass
4	Runs	0.739918	0.350485	0.534146	Pass
5	Longest run	0.739918	0.534146	0.534146	Pass
6	Rank	0.911413	0.213309	0.350485	Pass
7	FFT	0.350485	0.213309	0.534146	Pass
8	Non-overlapping template	0.534146	0.534146	0.350485	Pass
9	Overlapping template	0.739918	0.534146	0.122325	Pass
10	Universal	0.625716	0.161025	0.431301	Pass
11	Approximate entropy	0.911413	0.739918	0.350485	Pass
12	Random excursions	0.739210	0.345419	0.123169	Pass
13	Random excursions Variant	0.446049	0.259659	0.040016	Pass
14	Serial	0.213309	0.739918	0.350485	Pass
15	Linear complexity	0.739918	0.911413	0.991468	Pass

Table 14 NIST analyses for the sequences of encrypted images

Number	Statistical test	Lena P value	Peppers P value	Sailboat P value	Pass/Fail
1	Frequency	0.162606	0.906060	0.909852	Pass
2	Block frequency	0.017912	0.437274	0.131644	Pass
3	Cumulative sums	0.213309	0.534146	0.577721	Pass
4	Runs	0.275709	0.066882	0.685442	Pass
5	Longest run	0.834308	0.637119	0.056513	Pass
6	Rank	0.275709	0.350485	0.378119	Pass
7	FFT	0.739918	0.637119	0.852559	Pass
8	Non-overlapping template	0.833562	0.090936	0.883000	Pass
9	Overlapping template	0.350485	0.025193	0.012041	Pass
10	Universal	0.159092	0.711317	0.829665	Pass
11	Approximate entropy	0.739918	0.534146	0.211803	Pass
12	Random excursions	0.372069	0.103181	0.892352	Pass
13	Random excursions Variant	0.294020	0.298914	0.948317	Pass
14	Serial	0.739918	0.637119	0.178024	Pass
15	Linear complexity	0.350485	0.534146	0.814771	Pass

Comparative analysis

To quantitatively analyze the performance of our algorithm compared with some state-of-the-art schemes, we choose Lena image of size $256 \times 256 \times 3$ as the test image and calculate its adjacent pixels correlation, information entropy, NPCR, UACI, MSE and PSNR, we list the data in Table 15 and the maximum values are marked in bold. From Table 15, one may conclude that in contrast to the encryption schemes in Refs. [19, 23–25], which all process the three RGB components separately, our image encryption scheme implements a scrambling operation across the three RGB components, which effectively reduces the correlation of pixels among the components and the correlation between different components. And because our encryption scheme can be completely decrypted, the decrypted PSNR value can reach $+\infty$.

Table 15 Comparing results of key space, correlation coefficients and information entropy

Algorithm		Correlation coefficients			Information entropy	NPCR (%)	UACI (%)	MSE	PSNR (dB)
		H	V	D					
Ours	R	− 0.0008	**− 0.0004**	− 0.0073	7.9969	**99.64**	33.55	0	+ ∞
	G	**0.0009**	**0.0025**	− 0.0046	7.9973	99.62	33.43	0	+ ∞
	B	**− 0.0012**	0.0018	**0.0001**	7.9972	**99.62**	33.44	0	+ ∞
Ref. [19]	R	− 0.0266	− 0.0303	− 0.0031	7.9972	99.62	**32.98**	15.398	36.256
	G	0.0163	− 0.0169	0.0242	**7.9976**	99.63	30.41	10.716	37.047
	B	− 0.0462	− 0.0469	− 0.0046	7.9971	99.6	27.7	5.717	39.433
Ref. [23]	R	0.0021	0.0061	0.0012	**7.9974**	–	–	–	–
	G	0.0055	− 0.0053	− 0.0007	7.9973	–	–	–	–
	B	0.0035	**− 0.001**	− 0.001	7.997	–	–	–	–
Ref. [24]	R	0.0083	− 0.001	**− 0.001**	7.9972	99.61	33.56	0	+ ∞
	G	0.0049	0.0124	0.0124	7.9972	**99.67**	33.45	0	+ ∞
	B	0.0095	− 0.0042	− 0.0042	**7.9975**	99.61	**33.51**	**0**	+ ∞
Ref. [25]	R	**0.0004**	0.0062	0.0062	7.9972	99.61	33.44	–	–
	G	− 0.0018	0.0067	0.0067	7.9966	99.62	**33.48**	–	–
	B	0.0026	0.0044	0.0044	7.9968	99.6	33.45	–	–

The bold fonts indicate the best values in each element

Conclusion

Based on the lifting scheme and cross-component permutation, a new color image encryption algorithm named as CIE-LSCP is proposed. First, the initial values of Lorenz–Haken laser chaotic system are calculated using SHA256 hash values of plaintext images and external keys. The chaotic sequences used in encryption process are generated by this chaotic system. Second, PSLS is proposed to operate the plaintext images, the application of PSLS addresses the problem that some permutation algorithms are not effective for particular images (all-black and all-white images), as well as broadens the scope of application of image encryption method. Moreover, the preprocessed red, green and blue components of the plaintext images are performed by BCPIV to change the pixel position, the index vectors obtained from the chaotic sequences are applied in the permutation process, three random matrices are generated to determine the target location of the pixel movement, which reinforces the randomness and indestructibility of the permutation effect. Additionally, MSDRS is employed to change the value of the scrambled pixels to generate the final ciphertext image, after MSDRS, the pixel values in the ciphertext image are evenly distributed and can effectively resist statistical attacks. Simulation results and performance analyses show that the proposed CIE-LSCP has the ability to resist brute-force attacks, cropping attacks, noise attacks, and it may be applied in the secure transmission and storage of digital images.

Acknowledgements All the authors are deeply grateful to the editors for smooth and fast handling of the manuscript. The authors would also like to thank the anonymous referees for their valuable suggestions to improve the quality of this paper. This work is supported by the National Natural Science Foundation of China (Grant No. 61802111, 61872125, 62171114) and the Key Science and Technology Project of Henan Province (Grant No. 201300210400, 212102210094).

Author contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Xiuhui Chen, Mengxin Gong, Zhihua Gan, Yang Lu, Xiuli Chai and Xin He. The first draft of the manuscript was written by Xiuhui Chen and Xiuli Chai. All authors commented on this version of the manuscript. All authors read and approved the final manuscript.

Funding This work is supported by the National Natural Science Foundation of China (Grant No. 61802111, 61872125, 62171114), and the Key Science and Technology Project of Henan Province (Grant No. 201300210400, 212102210094).

Availability of data and material The authors declare that data and material will be made available on reasonable request.

Code availability The authors declare that the code will be made available on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethics approval Our research is not involved in human participants and animals.

Consent to participate All the listed authors have participated actively in the study.

Consent for publication All the listed authors have agreed to publish our article in Complex & Intelligent Systems.

References

1. Liu H, Wang X (2010) Color image encryption based on one-time keys and robust chaotic maps. Comput Math Appl 59(10):3320–3327. https://doi.org/10.1016/j.camwa.2010.03.017
2. Liu H, Wang X (2011) Color image encryption using spatial bit-level permutation and high-dimension chaotic system. Opt Commun 284(16–17):3895–3903. https://doi.org/10.1016/j.optcom.2011.04.001
3. Chen J, Zhu Z, Zhang L, Zhang Y, Yang B (2018) Exploiting self-adaptive permutation-diffusion and DNA random encoding for secure and efficient image encryption. Signal Process 142:340–353. https://doi.org/10.1016/j.sigpro.2017.07.034
4. Wang X, Gao S (2020) Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. Inf Sci 507:16–36. https://doi.org/10.1016/j.ins.2019.08.041
5. Chai X, Fu J, Gan Z, Lu Y, Zhang Y (2022) An image encryption scheme based on multi-objective optimization and block compressed sensing. Nonlinear Dyn. https://doi.org/10.1007/s11071-022-07328-3
6. Chai X, Wu H, Gan Z, Han D, Zhang Y, Chen Y (2021) An efficient approach for encrypting double color images into a visually meaningful cipher image using 2D compressive sensing. Inf Sci 556:305–340. https://doi.org/10.1016/j.ins.2020.10.007
7. Fridrich J (1998) Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurcation Chaos 8(6):1259–1284
8. Li S, Zhao L, Zhang H (2021) Image grey level encryption based on cat map. J Comput Appl 41(4):1148–1152
9. Zhang J, Chen N, Li Y (2019) Image encryption algorithm based on chaotic map and dynamic s-box. J China Acad Electron Inf Technol 14(11):1129–1135
10. Wang X, Feng L, Zhao H (2019) Fast image encryption algorithm based on parallel computing system. Inf Sci 486:340–358. https://doi.org/10.1016/j.ins.2019.02.049
11. Raza SF, Satpute V (2019) A novel bit permutation-based image encryption algorithm. Nonlinear Dyn 95(2):859–873. https://doi.org/10.1007/s11071-018-4600-8
12. Wang M, Wang X, Zhao T, Zhang C, Xia Z, Yao N (2021) Spatiotemporal chaos in improved cross coupled map lattice and its application in a bit-level image encryption scheme. Inf Sci 544:1–24. https://doi.org/10.1016/j.ins.2020.07.051
13. Wen W, Zhang Y, Su M, Zhang R, Chen J, Li M (2017) Differential attack on a hyper-chaos-based image cryptosystem with a classic bi-modular architecture. Nonlinear Dyn 87(1):383–390. https://doi.org/10.1007/s11071-016-3049-x
14. Wang X, Zhang Y, Bao X (2015) A novel chaotic image encryption scheme using DNA sequence operations. Opt Lasers Eng 73:53–61. https://doi.org/10.1016/j.optlaseng.2015.03.022
15. Wang X, Liu L, Zhang Y (2015) A novel chaotic block image encryption algorithm based on dynamic random growth technique. Opt Lasers Eng 66:10–18. https://doi.org/10.1016/j.optlaseng.2014.08.005
16. Pankaj S, Dua M (2021) A novel ToCC map and two-level scrambling-based medical image encryption technique. Netw Model Anal Heal Inform Bioinform 10:1–19. https://doi.org/10.1007/s13721-021-00324-4
17. Wu X, Wang K, Wang X, Kan H, Kurths J (2018) Color image DNA encryption using NCA map-based CML and one-time keys. Signal Process 148:272–287. https://doi.org/10.1016/j.sigpro.2018.02.028
18. Rehman A, Liao X (2019) A novel robust dual diffusion/confusion encryption technique for color image based on Chaos, DNA and SHA-2. Multimed Tools Appl 78(2):2105–2133. https://doi.org/10.1007/s11042-018-6346-1
19. Dua M, Suthar A, Garg A, Garg V (2021) An ILM-cosine transform-based improved approach to image encryption. Complex Intell Syst 7(1):327–343. https://doi.org/10.1007/s40747-020-00201-z
20. Wang X, Gao S (2020) Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network. Inf Sci 539:195–214. https://doi.org/10.1016/j.ins.2020.06.030
21. Xian Y, Wang X (2021) Fractal sorting matrix and its application on chaotic image encryption. Inf Sci 547:1154–1169. https://doi.org/10.1016/j.ins.2020.09.055
22. Xian Y, Wang X, Teng L (2021) Double parameters fractal sorting matrix and its application in image encryption. IEEE T Circ Syst Vid. https://doi.org/10.1109/TCSVT.2021.3108767
23. Khan M, Alanazi AS, Khan LS, Hussain I (2021) An efficient image encryption scheme based on fractal Tromino and Chebyshev polynomial. Complex Intell Syst 7(5):2751–2764. https://doi.org/10.1007/s40747-021-00460-4
24. Zhou J, Zhou N, Gong L (2020) Fast color image encryption scheme based on 3D orthogonal Latin squares and matching matrix. Opt Laser Technol 131:106437. https://doi.org/10.1016/j.optlastec.2020.106437
25. Ma K, Teng L, Wang X, Meng J (2021) Color image encryption scheme based on the combination of the fisher-yates scrambling algorithm and chaos theory. Multimed Tools Appl. https://doi.org/10.1007/s11042-021-10847-7
26. Kumar A, Dua M (2021) Novel pseudo random key & cosine transformed chaotic maps based satellite image encryption. Multimed Tools Appl 80:27785–27805. https://doi.org/10.1007/s11042-021-10970-5
27. Pak C, Huang L (2017) A new color image encryption using combination of the 1D chaotic map. Signal Process 138:129–137. https://doi.org/10.1016/j.sigpro.2017.03.011
28. Mansouri A, Wang X (2021) A novel one-dimensional chaotic map generator and its application in a new index representation-based image encryption scheme. Inf Sci 563:91–110. https://doi.org/10.1016/j.ins.2021.02.022
29. Talhaoui MZ, Wang X, Talhaoui A (2021) A new one-dimensional chaotic map and its application in a novel permutation-less image encryption scheme. Visual Comput 37(7):1757–1768. https://doi.org/10.1007/s00371-020-01936-z
30. Abd El-latif AA, Abd-El-Atty B, Venegas-Andraca SE (2020) Controlled alternate quantum walk-based pseudo-random number

generator and its application to quantum color image encryption. Physica A 547:123869. https://doi.org/10.1016/j.physa.2019.123869
31. Rehman AU, Liao X, Ashraf R, Ullah S, Wang H (2018) A color image encryption technique using exclusive-OR with DNA complementary rules based on chaos theory and SHA-2. Optik 159:348–367. https://doi.org/10.1016/j.ijleo.2018.01.064
32. Wang X, Liu P (2021) A new full chaos coupled mapping lattice and its application in privacy image encryption. IEEE T Circuits-I 69:3. https://doi.org/10.1109/TCSI.2021.3133318
33. Yang F, Mou J, Ma C, Cao Y (2020) Dynamic analysis of an improper fractional-order laser chaotic system and its image encryption application. Opt Lasers Eng 129:106031. https://doi.org/10.1016/j.optlaseng.2020.106031
34. Wang X, Yang J (2021) A privacy image encryption algorithm based on piecewise coupled map lattice with multi dynamic coupling coefficient. Inf Sci 569:217–240. https://doi.org/10.1016/j.ins.2021.04.013
35. Bisht A, Dua M, Dua S, Jaroli P (2020) A color image encryption technique based on bit-level permutation and alternate logistic maps. J Intell Syst 29(1):1246–1260. https://doi.org/10.1515/jisys-2018-0365
36. Yang F, Mou J, Liu J, Ma C, Yan H (2020) Characteristic analysis of the fractional-order hyperchaotic complex system and its image encryption application. Signal Process 169:107373. https://doi.org/10.1016/j.sigpro.2019.107373
37. Yang F, Mou J, Cao Y, Chu R (2020) An image encryption algorithm based on BP neural network and hyperchaotic system. China Commun 17(5):21–28. https://doi.org/10.23919/JCC.2020.05.003
38. Zhao J, Wang J, Wang H (2012) The study of finite-time stability active control method for Lorenz–Haken laser chaotic system. Acta Phys Sin 61(11):110209
39. Equation L (2012) Faculty of Mathematics Table of Contents List of Figures.
40. Sweldens W (1996) The lifting scheme: a custom-design construction of biorthogonal wavelets. Appl Comput Harmon Anal 3:186–200
41. Wang X, Liu C, Jiang D (2021) A novel triple-image encryption and hiding algorithm based on chaos, compressive sensing and 3D DCT. Inf Sci 574:505–527. https://doi.org/10.1016/j.ins.2021.06.032
42. Zhang Y, Wang X (2014) A symmetric image encryption algorithm based on mixed linear–nonlinear coupled map lattice. Inf Sci 273:329–351. https://doi.org/10.1016/j.ins.2014.02.156
43. Chai X, Zhi X, Gan Z, Zhang Y, Chen Y, Fu J (2021) Combining improved genetic algorithm and matrix semi-tensor product (STP) in color image encryption. Signal Process 183:108041. https://doi.org/10.1016/j.sigpro.2021.108041
44. Chai X, Bi J, Gan Z, Liu X, Zhang Y, Chen Y (2020) Color image compression and encryption scheme based on compressive sensing and double random encryption strategy. Signal Process 176:107684. https://doi.org/10.1016/j.sigpro.2020.107684
45. Wang X, Teng L, Qin X (2012) A novel colour image encryption algorithm based on chaos. Signal Process 92(4):1101–1108. https://doi.org/10.1016/j.sigpro.2011.10.023