Nonlinear Dyn (2021) 105:1859–1876
https://doi.org/10.1007/s11071-021-06663-1

ORIGINAL PAPER

# Color image encryption based on cross 2D hyperchaotic map using combined cycle shift scrambling and selecting diffusion

Lin Teng · Xingyuan Wang · Feifei Yang · Yongjin Xian

Received: 16 February 2021 / Accepted: 22 June 2021 / Published online: 5 July 2021

**Abstract** A novel color image encryption algorithm based on a cross 2D hyperchaotic map is proposed in this paper. The cross 2D hyperchaotic map is constructed using one nonlinear function and two chaotic maps with a cross structure. Chaotic behaviors are illustrated using bifurcation diagrams, Lyapunov exponent spectra, phase portraits, etc. In the color image encryption algorithm, the keys are generated using hash function SHA-512 and the information of the plain color image. First, the color plain image is converted to a combined bit-level matrix and permuted by the chaos-based row and column combined cycle shift scrambling method. Then, the scrambled integer matrix is diffused according to the selecting sequence which depends on the chaotic sequence. Last, decompose the diffusion matrix to get the encrypted color image. Simulation experiments and security evaluations show that the algorithm can encrypt the color image effectively and has good security to resist various kinds of attacks.

L. Teng · X. Wang (✉) · F. Yang · Y. Xian
School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China
e-mail: Wangxy@dlmu.edu.cn

X. Wang
Guangxi Key Lab of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China

**Keywords** Cross 2D hyperchaotic map · Color image encryption · Combined cycle shift scrambling · Selecting diffusion

## 1 Introduction

In modern society, along as the fast growth of the Internet, big data, artificial intelligence, and 5G communications, a large amount of information has been digitized and transmitted over the network. As an important information carrier, the security of digital images attracts more and more attention. Because color images contain richer information than gray-level images, the related research in encryption of color images has been a hot research topic [1–11].

The method to encrypt image is different from the way to encrypt text because the image has characteristics of massive data volume and highly relevant contents between pixels. Thus, traditional encryption technologies include DES, IDES, and RSA are not any more appropriate for encrypting image. The chaotic system has the features of sensitiveness of control parameters and initial conditions, ergodicity, random-like behavior, and unpredictable orbit. It corresponds to the concepts of key design, confusion, diffusion, and round-robin in cryptography, which makes chaos theory have great potential in the field of cryptography.

In the last decades, many methods of image cryptography are introduced based on chaos theory [12–20]. Because of the limitation of computer precision, the dynamical behavior of most low-dimensional chaotic systems degenerates, which leads to the defects of small keyspace and weak security performance. The image encryption algorithms design of using a high-dimensional continuous chaotic system still have the defect that the encrypted image interruptible by known-plaintext attack or selected plaintext attack [21–24]. Besides, the computational complexity and time consumption of the encryption algorithm are increased.

Compared with chaotic systems, hyperchaotic maps having more than one positive Lyapunov exponent have more complicated and abundant behaviors of dynamics, which enhance the stochasticity and unpredictability of the relevant systems [25]. Therefore, when applied to encryption, the hyperchaotic system can generate larger keyspace and more complex random sequences. Using the hyperchaotic system to design algorithms for encrypting color image will greatly improve the security of the algorithm [26–32].

How to design an encryption algorithm according to the characteristics of the color image and chaotic system still has a large research value. In the last several years, some new chaotic maps have been applied to image encryption algorithms [33–37]. Some of the new chaotic maps still have defects that trajectory is not distributed in the whole phase space or has no complex dynamic behavior.

Because of the above shortcomings, we design a nonlinear discrete cross 2D hyperchaotic map and propose a color image cryptography technique based on this hyperchaotic map. The 2D hyperchaotic map is constructed using one nonlinear function and two chaotic maps with cross structure. Chaotic behaviors are illustrated using bifurcation diagrams, Lyapunov exponent spectra, phase portraits, etc. The simulation results prove that the cross 2D hyperchaotic map has good chaotic performance. In the color image encryption algorithm, the keys are associated with the plain color image, that is, distinct plain color images produce different keys, thus enhancing security against selected plaintext/ciphertext attacks. The color plain image is converted to a combined bit-level matrix and permuted by the chaos-based row and column combined cycle shift scrambling method. Then, the scrambled integer matrix is diffused according to the selecting sequence which depends on the chaotic sequence. The cipher color image is obtained by decomposed the diffused matrix. The proposed encryption algorithm makes the three color components of the color image influence each other to eliminate the correlations between them. Simulation results show that the algorithm can encrypt the color image effectively and has good security.

The remainder of this paper has been structured in the following ways. Section 2 introduces the model of the nonlinear cross 2D hyperchaotic map. In Sect. 3, we analyze the dynamic behaviors of the proposed cross 2D hyperchaotic map. Section 4 introduces algorithm for encryption and decryption of color images. Section 5 evaluates the results of the experiments and the security of the method. Section 6 provides the conclusion.

## 2 Cross 2D hyperchaotic map

In this paper, a cross 2D hyperchaotic map is proposed, and the structure is shown in Fig. 1. The proposed model has two inputs variables and two cross outputs, which is when input is $x_n$ the output is $y_{n+1}$, and when input is $y_n$ the output is $x_{n+1}$. Function $f$ is a nonlinear function, functions $F$ and $G$ are two chaotic maps. The $+$ sign indicates the addition of two inputs. The mathematical expression of the model is shown in Eq. (1).

$$\begin{cases} x_{n+1} = F(f(y_n)) \\ y_{n+1} = G(x_n + y_n) \end{cases} \tag{1}$$

where $F$ and $G$ can be selected as any one-dimensional chaotic map.

The chaotic map $F$ is chosen as the infinite collapse map [38] defined as

$$x_{i+1} = \sin\left(\frac{\alpha}{x_i}\right) \tag{2}$$

where its control parameter $\alpha \neq 0$. And the chaotic map $G$ is chosen as the Sine map in this paper. The Sine map is given as

$$x_{i+1} = \beta \sin(\pi x_i) \tag{3}$$

where $\beta$ is a control parameter and it has an interval of (0,1]. The nonlinear function $f$ is set to sin function,

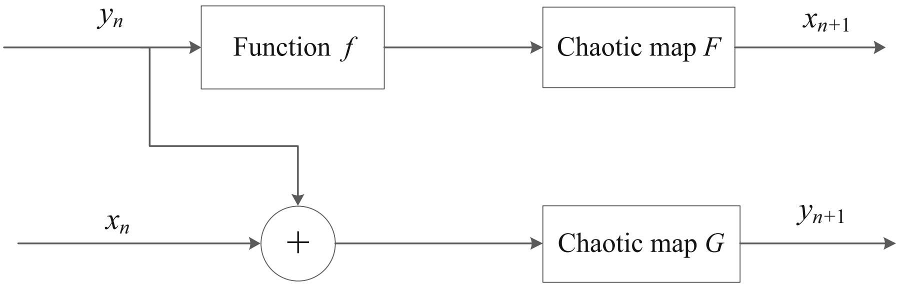


**Fig. 1** Diagram of Nonlinear cross 2D hyperchaotic map

that is $f(x) = sin(x)$. So the mathematical expression of the modified 2D coupled chaotic map model is set to

$$\begin{cases} x_{i+1} = \sin\left(\dfrac{\alpha}{\sin(y_i)}\right) \\ y_{i+1} = \beta \sin(\pi(x_i + y_i)) \end{cases} \tag{4}$$

where its control parameter $\alpha \neq 0$, $\beta \in (0, 1]$, the initial value $y_0 \neq 0$.

## 3 Dynamics analysis of the cross 2D hyperchaotic map

### 3.1 Bifurcation diagram

The dynamical behaviors of a chaotic system can be evaluated by its bifurcation diagram. A bifurcation diagram shows the changes in the system's motion state along with the control parameters. The evolution process of the system can be directly observed by the bifurcation diagram. Set the initial conditions $x_0 = 0.3$and $y_0 = 0.6$. Fixing $\beta = 1$, when the control parameter $\alpha$ in the range [0.25, 2], the bifurcation diagram is illustrated in Fig. 2a. From Fig. 2a, we can see that the system goes through a periodic state to chaotic orbit. When $0.55 < \alpha \leq 2$, the system exhibits chaotic behavior. Fixing $\alpha = 1$ illustrates the bifurcation diagram when control parameter $\beta$ in the range [0.1, 1] in Fig. 2b. The results show that the system exhibits the periodic behavior between (0.34, 0.353), and produces chaotic attractors throughout the remaining range.

### 3.2 Lyapunov exponent spectrum

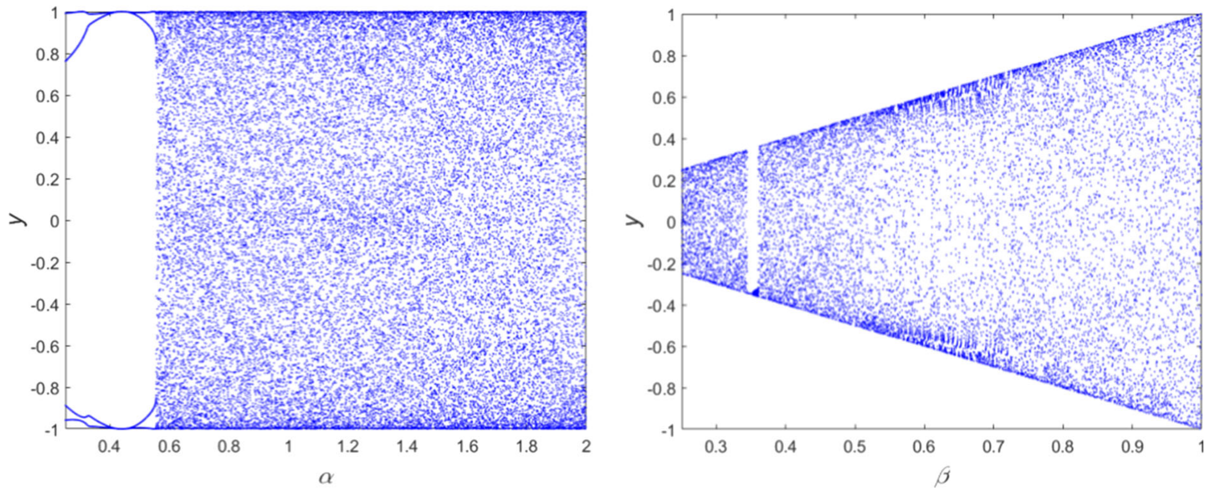
The Lyapunov exponent is one of the characteristics used to identify the chaotic characteristics of dynamic systems. For a high-order dynamic system, due to the different directions of the initial separation vector, the exponential divergence rate will be different, so there are multiple Lyapunov exponents. The system has the same number of Lyapunov exponents and order. Consequently, the two-dimensional system with two Lyapunov exponents.

The Lyapunov exponents of the system are two negative numbers, indicating that the system is at a fixed point. The system has a negative and a zero Lyapunov exponent when it is in a periodic orbit. The Lyapunov exponent of the system is one positive and one negative when it is in a chaotic orbit. The Lyapunov exponent of the system is two positive numbers when the system is in hyperchaotic state. Hyperchaotic systems normally possess more sophisticated and abundant dynamic behaviors compared to chaotic systems, which enhances the stochastic and unpredictable nature of the systems.

In the proposed map, the Lyapunov exponents spectrum is shown in Fig. 3a when fixing $\beta = 1$ and varying $\alpha$. It can be seen that when $\alpha \in (1.55,\ 1.47)$, the largest Lyapunov exponent is positive, so the system is in the chaotic state. When $\alpha \in [1.47,\ 2]$, the two Lyapunov exponents are both positive, so the system can generate hyperchaotic attractors.

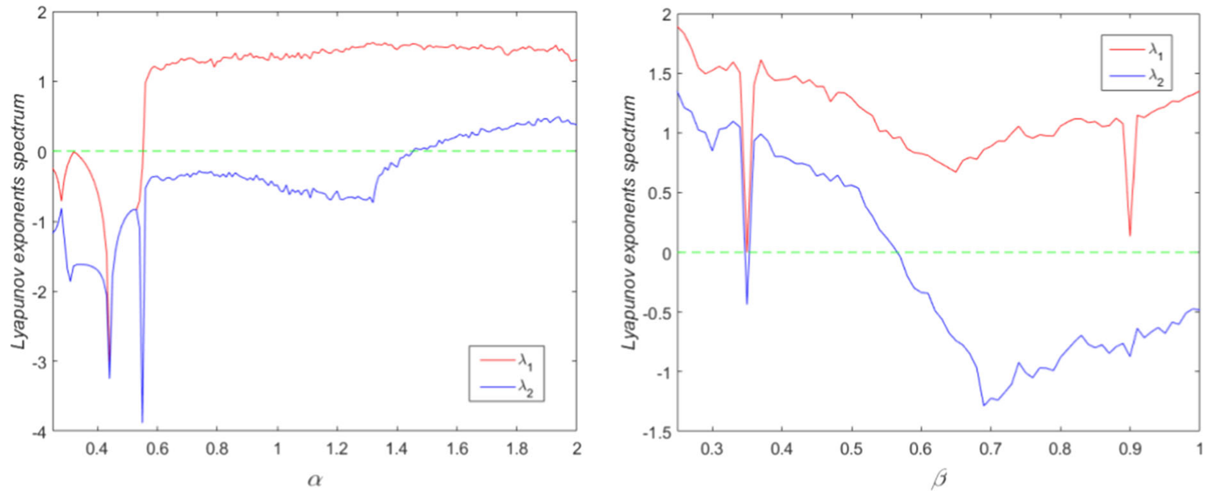
Figure 3b shows the Lyapunov exponents spectrum varying $\beta$ when $\alpha = 1$. We can see that the system exhibits the periodic behavior when $\beta \in (0.34,\ 0.353)$. There are two positive Lyapunov exponents when $\beta \in [0.25,\ 0.34] \cup [0.354,\ 0.56]$, and the map exhibits the hyperchaotic state. The system is in chaotic attractors in the range of $\beta \in (0.56,\ 1]$.

It can be seen that there is a one-to-one correspondence between the Lyapunov exponent spectrum and the bifurcation diagrams.



**(a)** Bifurcation diagram for $y$ against $\alpha$ **(b)** Bifurcation diagram for $y$ against $\beta$

**Fig. 2** Bifurcation diagrams of the proposed system



**(a)** Lyapunov exponents spectrum varying $\alpha$ **(b)** Lyapunov exponents spectrum varying $\beta$

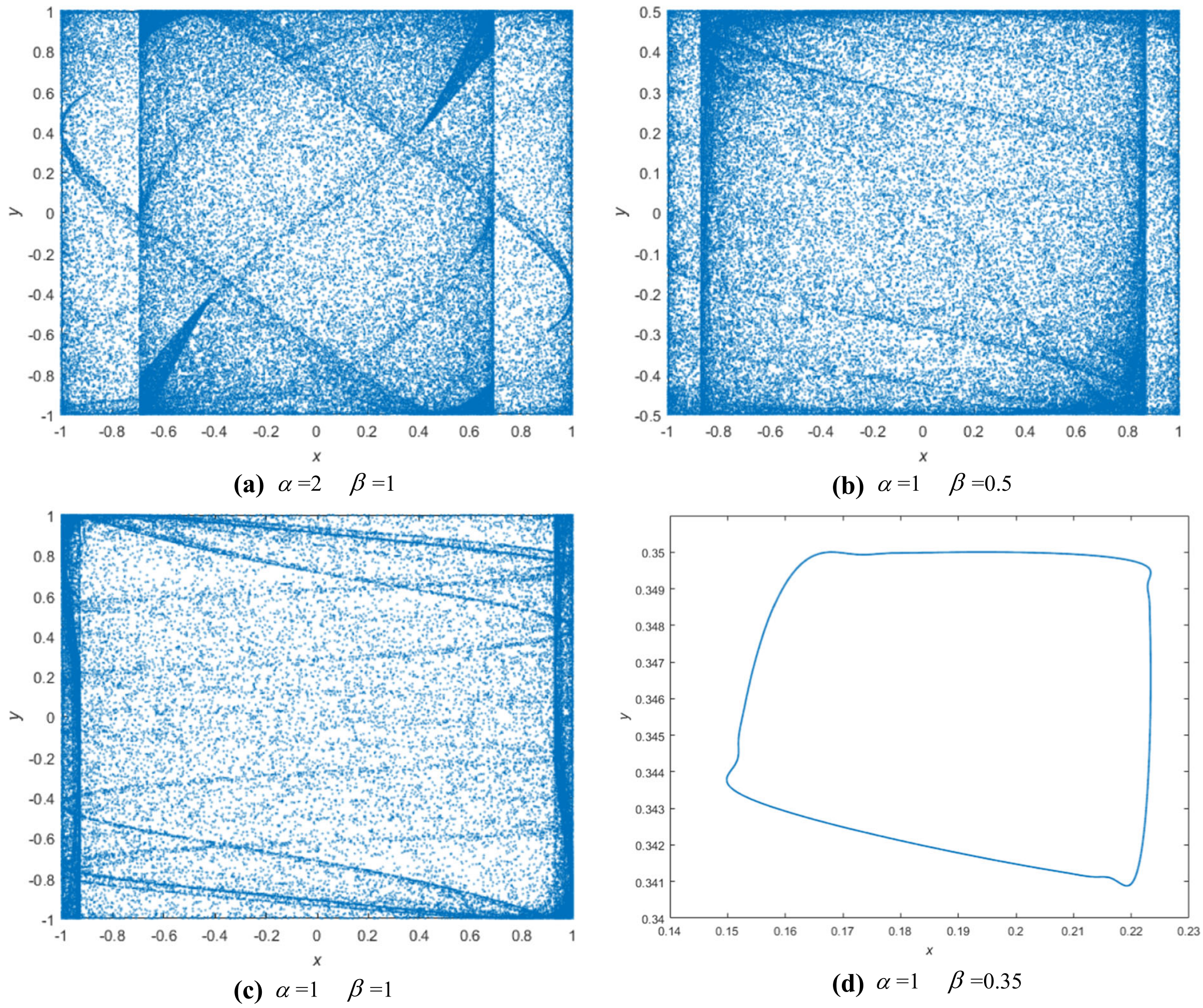
**Fig. 3** Lyapunov exponents spectrum

### 3.3 Attractor phase diagram

A chaotic system with good chaotic performance usually has complex attractors which occupy a large area in the phase diagram. Set the initial conditions $x_0 = 0.3$ and $y_0 = 0.6$; the attractor phase diagrams are generated in Fig. 4. The system generates the hyperchaotic attractors as shown in phase diagrams (Fig. 4a and b) when parameters $\alpha = 2$, $\beta = 1$ and $\alpha = 1$, $\beta = 0.5$. When parameters $\alpha = 1$ and $\beta = 1$, the attractor phase diagram is shown in Fig. 4c; the system generates the chaotic attractor. When parameters $\alpha = 1$ and $\beta = 0.35$, the system demonstrates the periodic behavior displayed in Fig. 4d.

### 3.4 Sensitivity analysis of initial value

A well-performing chaotic system can be extremely sensible to its initial values. A slight difference of initial value can produce a completely diverse chaotic trajectory. In order to analyze the initial sensitivity of the proposed hyperchaotic map, the initial value is varied $10^{-16}$, and the experimental outcomes are shown in Fig. 5. It is observed that the proposed



**(a)** $\alpha=2$ $\beta=1$

**(b)** $\alpha=1$ $\beta=0.5$

**(c)** $\alpha=1$ $\beta=1$

**(d)** $\alpha=1$ $\beta=0.35$

**Fig. 4** Attractor phase diagrams

hyperchaotic system has great sensitivity to the initial value.

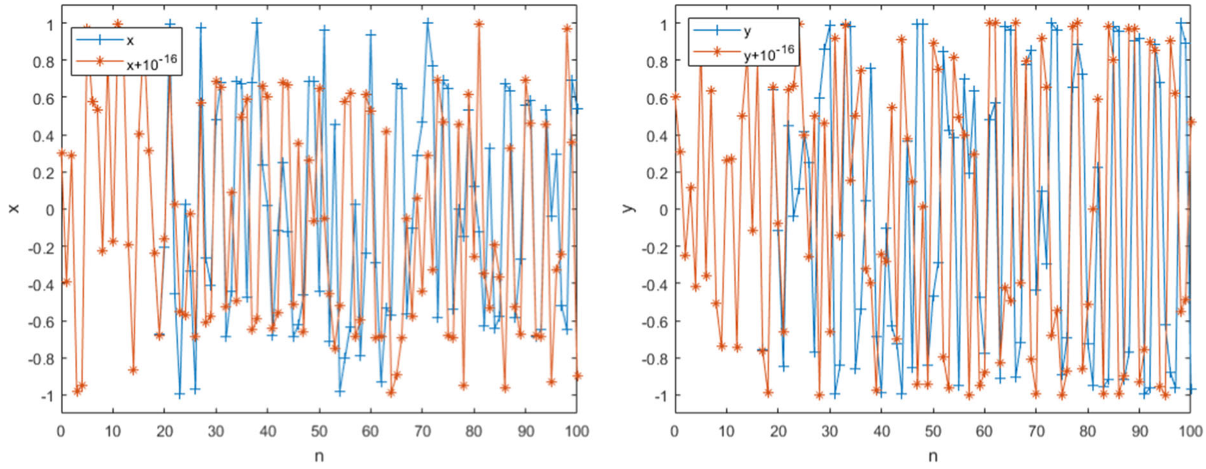
The simulation results indicate that, compared with other two-dimensional chaotic maps, the proposed 2D hyperchaotic map has better chaotic property as shown in Table 1.

## 4 Cross 2D hyperchaotic map-based color image encryption and decryption algorithm

Since the proposed cross 2D hyperchaotic map has good chaotic performance, this section proposes a color image cryptographic method based on this hyperchaotic map. Figure 6 displays the flowchart for the proposed encryption scheme, which includes the substitution process and the diffusion process.

### 4.1 Key generation

Due to the irreversibility and strong security of the hash algorithm, we use it to generate the key of the color image encryption scheme. A 512-bit secret key $K$ is produced by the SHA-512 hash function $K = SHA_{512}(\ )$, in which the input value of this hash function is related to the plain color image to increase the security and able to resist select plaintext/ cipher-text attack. Divide the 512-bit key $K$ into 8-bit blocks, which can be expressed as $K = k_1, k_2, \ldots, k_{64}$. $K$ is processed and grouped in sub-keys as follows:



**(a)** $x_0$ changed $10^{-16}$ **(b)** $y_0$ changed $10^{-16}$

**Fig. 5** Initial value sensitivity

**Table 1** Chaotic property compared with other 2D chaotic maps

| System | The proposed 2D hyperchaotic map | |
|---|---|---|
| | Better ergodicity property | Larger hyperchaotic range |
| 2D-SLMM [33] | √ | √ |
| 2D-LASM [34] | √ | √ |
| 2D-SIMM [35] | - | √ |

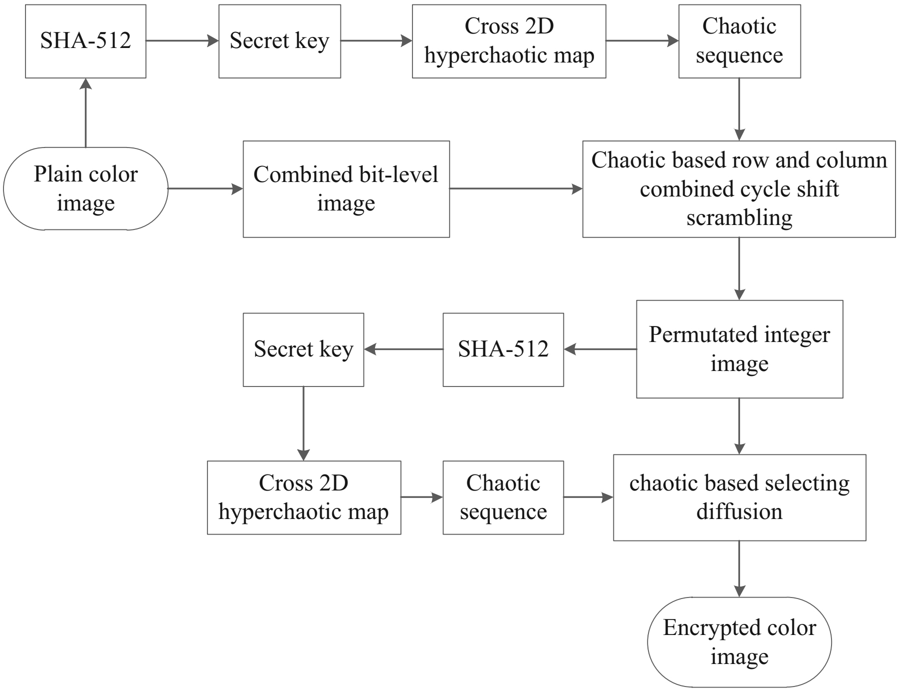


**Fig. 6** Flowchart of proposed encryption algorithm

$$\begin{cases} K1 = \dfrac{k_1 \oplus k_2 \oplus \ldots \oplus k_{16}}{256} \\ K2 = \dfrac{k_{17} \oplus k_{18} \oplus \ldots \oplus k_{32}}{256} \\ K3 = \dfrac{k_{33} \oplus k_{34} \oplus \ldots \oplus k_{48}}{256} \\ K4 = \dfrac{k_{49} \oplus k_{50} \oplus \ldots \oplus k_{64}}{256} \end{cases}. \tag{5}$$

### 4.2 Chaos-based row and column combined cycle shift scrambling

In the scrambling process, average the statistical information of the image by varying the position of the image pixels to make the image energy uniform. A chaos-based row and column combined cycle shift scrambling method is proposed:

Step 1. The image matrix is assumed to be of size $M \times N$, that is $M$ rows and $N$ columns. Set the row vectors $PR$ and column vectors $PC$. Process rows and columns together, there are $M + N$ vectors.

Step 2. A chaotic sequence $X_1$ of length $M + N$, a chaotic sequence $X_2$ of length $M$ and a chaotic sequence $X_3$ of length $N$ are selected. The chaotic sequences are further processed by

$$X_1'(i) = \mathrm{mod}(ceil(X_1(i) \times 10^{15}), M + N), \quad 1 \le i \le M + N \tag{6}$$

$$X_2'(j) = \mathrm{mod}(ceil(X_2(j) \times 10^{15}), N), \quad 1 \le j \le M \tag{7}$$

$$X_3'(k) = \mathrm{mod}(ceil(X_3(k) \times 10^{15}), M), \quad 1 \le k \le N \tag{8}$$

where $ceil(x)$ gives back the smallest integer greater than or equal to $x$, and mod() represents the modular action.

Step 3. Sort the sequence $X_1'$, record the transform position $TP$ of each element in the chaotic series, thus the length of vector $TP$ is $M + N$ and the elements of $TP$ are all non-repeating integers between 1 and $M + N$.

Step 4. Exchange the locations of image elements using row and column combined cycle shift by

$$\begin{cases} PR(TP(i)) = circshift(PR(TP(i)), X_2'(j)), \ j = j + 1 \\ PC(TP(i) - M) = circshift(PC(TP(i) - M), X_3'(k)), \ k = k + 1 \end{cases} \tag{9}$$

where $circshift(A, SHIFTSIZE)$ circularly move the data in array $A$ using the $SHIFTSIZE$ elements. When $TP(i) \le M$, circularly shift the $TP(i)$ th row by $X_2'(j)$ elements to the right. When $TP(i) > M$, circularly shift the $TP(i) - M$ th column by $X_3'(k)$ elements down.

Take an image matrix with the size of 4 × 4 as an example shown in Fig. 7, where $TP$ = {8, 1, 4, 7, 3, 2, 5, 6}, $X_2'$ = {3, 2, 3, 1}, $X_3'$ = {2, 0, 1, 3}. It can be seen that the position of every pixel varies only after a single permutation.

### 4.3 Permutation process

Step 1. In general, the size of the color plain image $P$ is assumed to be $M \times N$. And there are three components in the color image: the red ($R$) part, the green ($G$) part, and the blue ($B$) part. Each part pixel's value in the range of 0 to 255. Hence, one pixel can be converted into 8-bit binary value. Therefore, an $M \times N$ size color image $P$ can be expended to three binary image matrixes $R_b$, $G_{b,}$ and $B_b$ with size $M \times 8\,N$. Combine the $R_b$, $G_b$, and $B_b$ matrixes vertically and get a combined image matrix $P_b$ with $3 \times M$ rows, $8 \times N$ columns. Thus, the three color image components can influence each other.

Step 2. Set the permute key $K_{permute} = SHA_{512}(R, G, B)$, which $R$, $G$, and $B$ are all the pixels in each color component of plain image so that different plain images will get a distinct key. Obtain the sub-keys $K1$, $K2$, $K3$, $K4$ according to Eq. (5), and the chaotic system (4) parameters and initial conditions are given:

$$\begin{cases} \alpha = \mathrm{mod}(K1, 0.5) + 1 \\ \beta = \mathrm{mod}(K2, 0.5) + 0.6 \\ x_0 = \mathrm{mod}(K3, 1) \\ y_0 = \mathrm{mod}(K4, 1) \end{cases}. \tag{10}$$

Step 3. Iterate the chaotic system (4) 3 $M$ + 8 $N$ + $k$ times, the length of the sequence discarded is $k$ for better chaos, where $k = \mathrm{mod}(sum(R + G + B), 100) + 500$, $sum(R +$

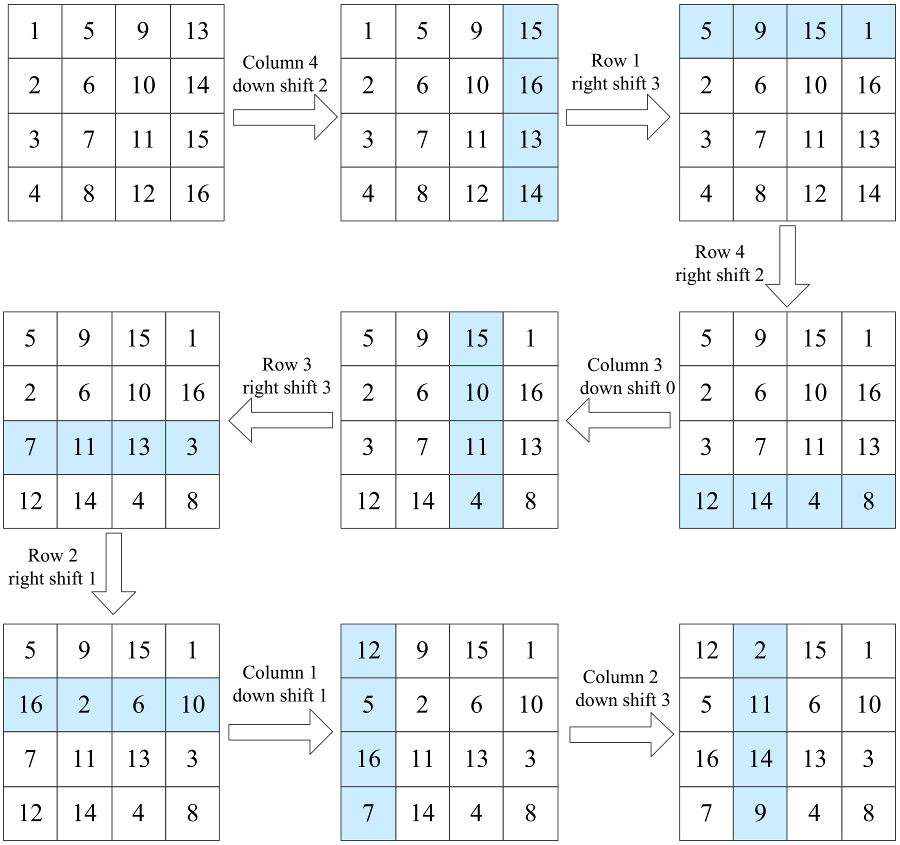


**Fig. 7** 4 × 4 matrix scrambled by row and column combined cycle shift

$G + B)$ means the sum of all these elements in three components. Get the chaotic sequences $x(i) = \{x_1, x_2, \ldots, x_{3M+8N}\}$ and $y(i) = \{y_1, y_2, \ldots, y_{3M+8N}\}$.

Step 4. Use the chaos-based row and column combined cycle shift scrambling method described in Sect. 4.2 to permute the bit-level image matrix $P_b$ with the size of 3 $M$ × 8 $N$, where the chaotic sequence $X_1 = x(i)$ ($i$ = 1, 2,…, 3 $M$ + 8 $N$), $X_2 = y(i)$ ($i$ = 1, 2,…, 3 $M$), $X_3 = y(i)$ ($i$ = 3 $M$ + 1,…, 3 $M$ + 8 $N$). A permutated bit-level image matrix $P_b'$ is generated.

Step 5. Transform $P_b'$ to an integer image matrix denoted as $P'$ with the size of 3 $M$ × $N$. The position of pixels is changed as well as the value of the pixels in the permutated image $P'$.

## 4.4 Diffusion process

During the diffusion process, the pixel values of the image are altered so that small differences in a single pixel spread over a maximum number of pixels. The proposed diffusion equation is chosen according to the selecting sequence which depends on the chaotic sequence.

Step 1. Set the diffuse key $K_{diffuse} = SHA_{512}(P'([i\ j\ k], :))$, where $P'([i\ j\ k], :)$ is the $i$th row, $j$th row, and $k$th row of the permutated image matrix $P'$. Obtain the sub-keys $K1^{'}$, $K2^{'}$, $K3^{'}$, $K4^{'}$ according to Eq. (5), and set the initial conditions and parameters of the chaotic system (4) using Eq. (10).

Step 2. The hyperchaotic map (4) is iterated $3\ M \times N + k_1$ times, where $k_1 = \mathrm{mod}(sum(sum(P'(:, i : j))), 100) + 500$, $P'(:, i : j)$ is the $i$th to $j$th columns of $P'$. Discard the former $k$ values of the chaotic sequences. Get the chaotic series $x'(i) = \{x_1', x_2', \ldots, x_{3M\times N}'\}$ and $y'(i) = \{y_1', y_2', \ldots, y_{3M\times N}'\}$.

Step 3. The selecting sequence $S(i) = \{s_1, s_2, \ldots, s_{3M\times N}\}$ and the diffusion sequence $D(i) =$

$\{d_1, d_2, ..., d_{3M\times N}\}$ can be obtained using Eqs. (11) and (12)

$$S(i) = \mathrm{mod}(ceil(x'(i) \times 10^{15}), 3) \tag{11}$$

$$D(i) = \mathrm{mod}(ceil(y'(i) \times 10^{15}), 256). \tag{12}$$

Step 4. According to the chaotic-based selecting diffusion equations below, the encrypted combined image pixel matrix $C'(i) = \{c'_1, c'_2, ..., c'_{3M\times N}\}$ can be acquired out of the diffusion matrix $D$ and the permutation image $P'$.

$$\begin{cases} C'(i) = \mathrm{mod}(P'(i) + D(i), 256) \ S(i) = 0 \\ C'(i) = \mathrm{mod}(P'(i) - D(i), 256) \ S(i) = 1 \\ C'(i) = P'(i) \oplus D(i) \ S(i) = 2 \end{cases} \tag{13}$$

where $\oplus$ denotes bit-level *XOR* operator.

Step 5. Vertical transformation $C'$ into the $R$, $G$, and $B$ color matrix to get its cipher color image $C$ in size of $M \times N$.

### 4.5 Decryption algorithm

The decryption method is the reverse of the encryption method using the permute key $K_{permute}$ and the diffuse key $K_{diffuse}$ provided in the encryption algorithm, which means $K_{permute}$ and $K_{diffuse}$ should be provided and known in the decryption method.

Step 1. Obtain the cipher image $C$ and convert it to the $R$, $G$, and $B$ color components' matrix. Combine the color matrixes vertically and get a combined encrypted image matrix $C'$ with $3 \times M$ rows, $N$ columns.

Step 2. Use the same diffuse key $K_{diffuse}$ as the encryption algorithm to obtain the same selecting sequence $S(i)$ and the diffusion sequence $D(i)$.

Step 3. The matrix $C_D$ can be attained using the inverse diffusion formula equation below from the diffusion matrix $D$ and the encrypted image matrix $C'$.

$$\begin{cases} C_D(i) = \mathrm{mod}(C'(i) - D(i), 256) \ S(i) = 0 \\ C_D(i) = \mathrm{mod}(C'(i) + D(i), 256) \ S(i) = 1 \\ C_D(i) = C'(i) \oplus D(i) \ S(i) = 2 \\ \end{cases}. \tag{14}$$

Step 4. Extended the matrix $C_D$ to its binary image matrixes $C_{Db}$. Use the same permute key $K_{permute}$ as the encryption algorithm to obtain the same chaotic sequences $X_1$, $X_2$, and $X_3$ in the encryption permutation process.

Step 5. The matrix $C_P$ can be acquired using the reverse row and column combined cycle shift scrambling Eq. (15) to reverse permute the bit-level image matrix $C_{Db}$.

$$\begin{cases} PR(TP(i)) = circshift(PR(TP(i)), -X'_2(j)), \ j = j + 1 \ TP(i) \le M \\ PC(TP(i) - M) = circshift(PC(TP(i) - M), -X'_3(k)), k = k + 1 TP(i) > M \end{cases}. \tag{15}$$

Step 6. Transform $C_P$ to an integer image matrix denoted as $C_I$ which its size is 3 $M \times N$. Convert $C_I$ into the $R$, $G$, and $B$ color matrix vertically to obtain the decrypted image.

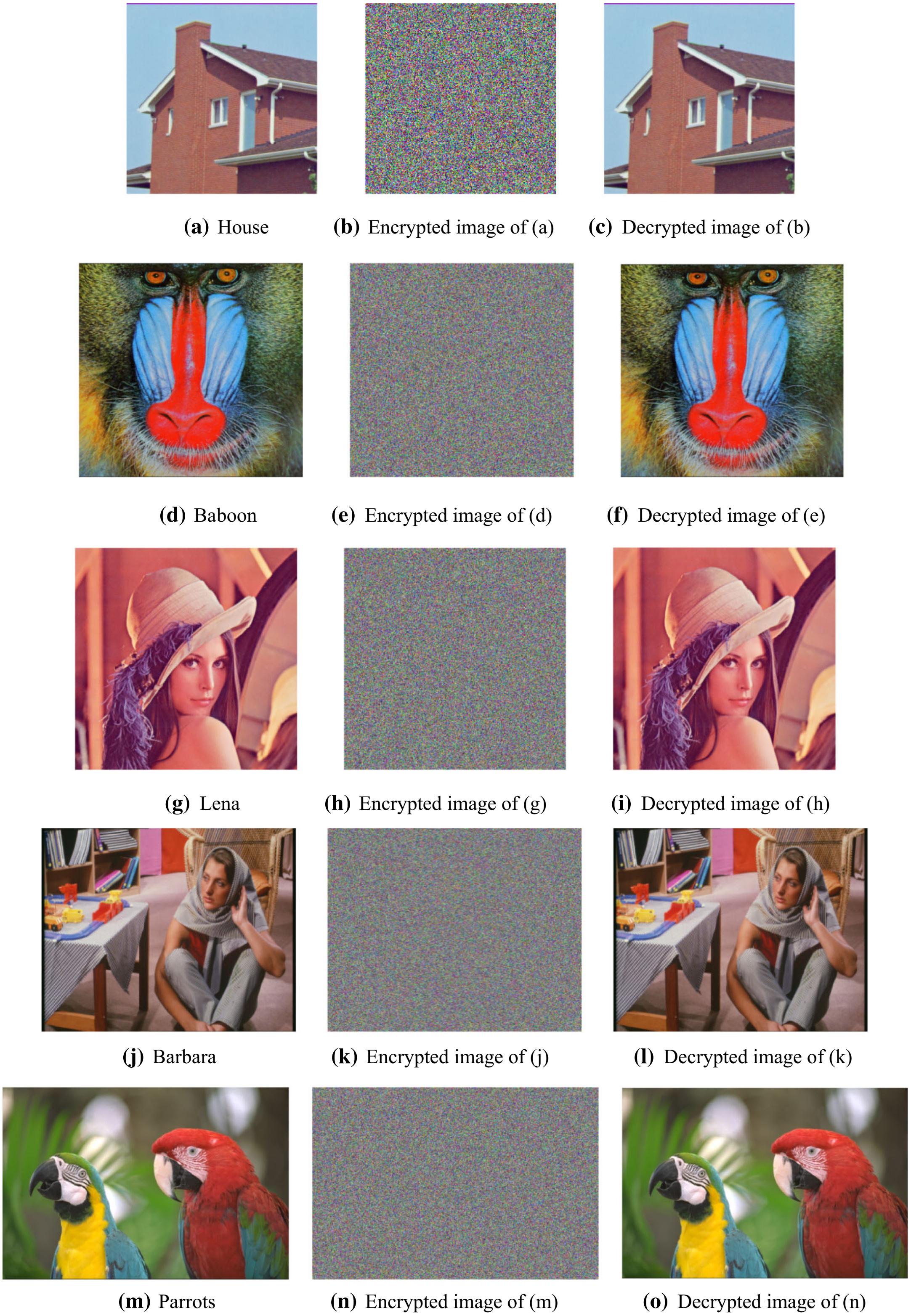
## 5 Experimental results and performance analysis

The experimental results and performance of the algorithm are analyzed. To verify the effectiveness of the cryptographic system, a number of images were simulated numerically.

The chosen sample images are the different sizes color images of 256 × 256 House, 500 × 480 Baboon, 512 × 512 Lena, 512 × 512 Peppers, 720 × 576 Barbara, and 768 × 512 Parrots. Figure 8 illustrates the experimental simulation performance of the House, Baboon, Lena, Barbara, and Parrots images. The encrypted images are comparable to the image of noise with no visual leakage of information.

### 5.1 Keyspace analysis

The encryption algorithm with larger keyspace are better able to resist brute-force assaults and have higher degree of security. In this method, the key contains the initial conditions $x_0$, $y_0$, $x'_0$, $y'_0$, parameters $\alpha$, $\beta$, $\alpha'$, $\beta'$ and the discard length $k$, $k_1$ of the hyperchaotic system in the permutation and diffusion processes. Because the computational precision is $10^{-16}$, the size of the keyspace for one round encryption is $10^{16\times 8} > 2^{425}$ which is much bigger than $2^{100}$ to guarantee the security according to [39]. So the keyspace is sufficiently large to defend against any violent attack.



**(a)** House **(b)** Encrypted image of (a) **(c)** Decrypted image of (b)

**(d)** Baboon **(e)** Encrypted image of (d) **(f)** Decrypted image of (e)

**(g)** Lena **(h)** Encrypted image of (g) **(i)** Decrypted image of (h)

**(j)** Barbara **(k)** Encrypted image of (j) **(l)** Decrypted image of (k)

**(m)** Parrots **(n)** Encrypted image of (m) **(o)** Decrypted image of (n)

**Fig. 8** Encryption and decryption results

### 5.2 Key sensitivity analysis

For the high-security image encryption algorithm, key sensitivity is an essential feature. A slightly different key used in encryption will generate a radically altered cryptographic image. And minor changes in the key used for decryption will cause decryption failure. The encryption and decryption keys are changed $10^{-16}$ to analyze the key sensitivity of the proposed scheme. Figure 9 displays the experimental performance of the key sensitivity. Any small modification in the encryption key will lead to a totally distinct encrypted image. A minor alteration in a key will produce an entirely different image, and the correct decrypted image will not be available.

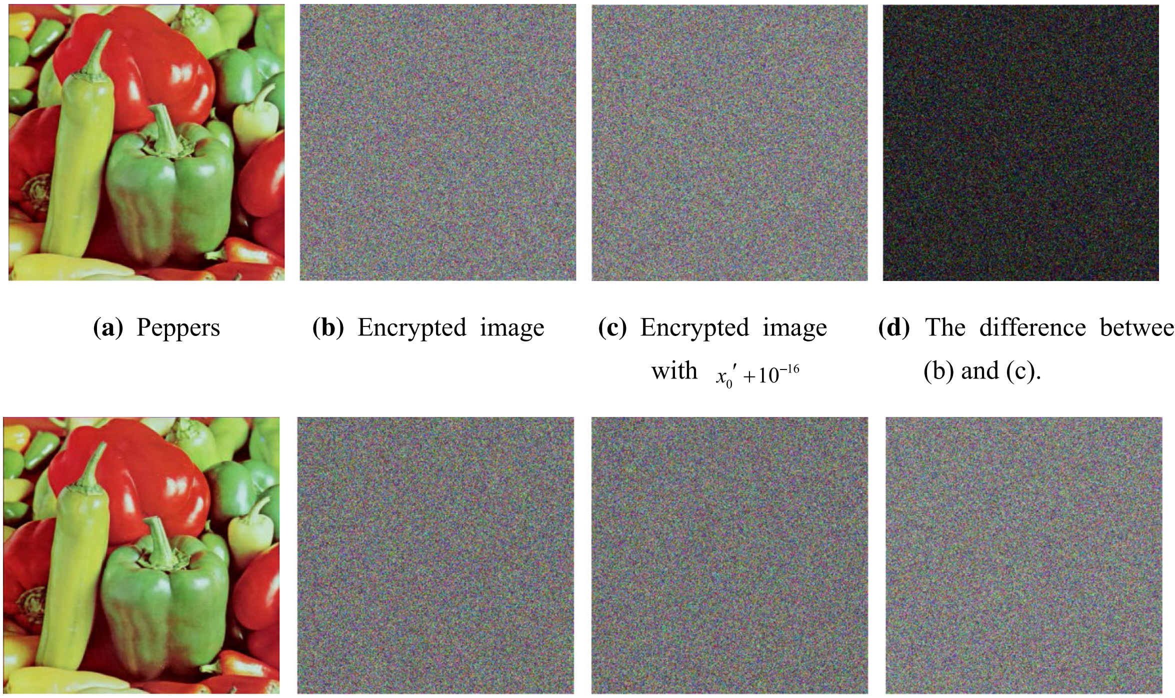
### 5.3 Histogram

The image histogram reveals information about the distribution of pixel values. To prevent statistical analysis attacks from recovering any meaningful message from the histogram of a cryptographic image, the cryptographic image needs to be uniformly distributed. Figure 10 shows the histograms of the plain images and the relevant cryptographic images. As shown in Fig. 10, the histogram allocation of the cryptographic images is uniform, which makes it very hard for any attacker to analyze the cryptographic image through a statistically analysis attack.

### 5.4 Correlation of adjacent pixels

The correlation between the neighboring pixels is at a high degree in a plain image, which may reveal the attacker's statistical information. Therefore, encrypted images should minimize the correlation between the adjacent pixels. For each pair, the correction factor is computed using the equation below.

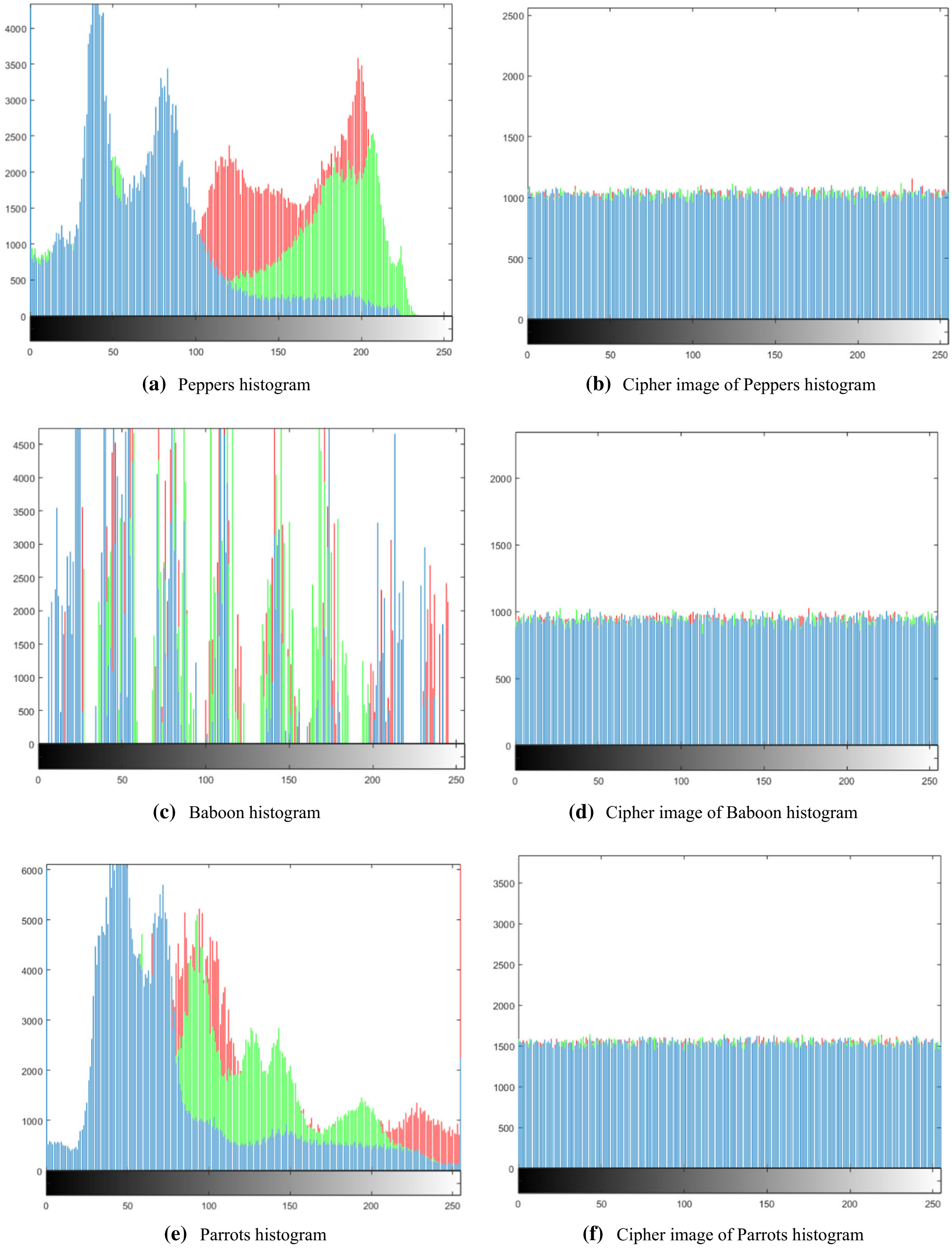
$$r_{xy} = \frac{\operatorname{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{16}$$



**(a)** Peppers **(b)** Encrypted image **(c)** Encrypted image with $x_0' + 10^{-16}$ **(d)** The difference between (b) and (c).

**(e)** Decrypted with correct key **(f)** Decrypted with $\beta + 10^{-16}$ **(g)** Decrypted with $x_0 + 10^{-16}$ **(h)** Decrypted with $y_0' + 10^{-16}$

**Fig. 9** Key sensitivity



**(a)** Peppers histogram

**(b)** Cipher image of Peppers histogram

**(c)** Baboon histogram

**(d)** Cipher image of Baboon histogram

**(e)** Parrots histogram

**(f)** Cipher image of Parrots histogram

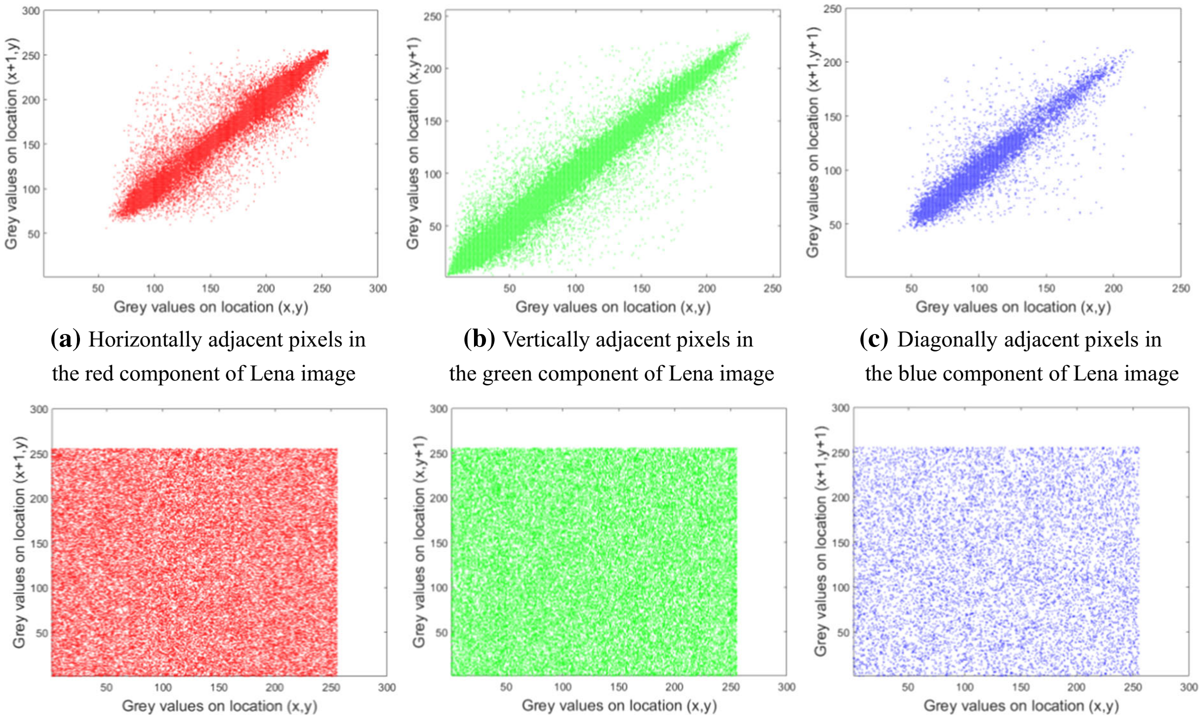
**Fig. 10** Histogram of different plain images and cipher images

where $E(x)=\frac{1}{N}\sum_{i=1}^{N}x_i$, $D(x)=\frac{1}{N}\sum_{i=1}^{N}(x_i-E(x))^2$, $\mathrm{cov}(x,y)=\frac{1}{N}\sum_{i=1}^{N}(x_i-E(x))(y_i-E(y))$, $x$ and $y$ are gray-scale values of two neighboring pixels in the image.

The distribution of the correlation between the color components of the original Lena image and the encrypted image in horizontal, vertical, and diagonal adjacent pixels are shown in Fig. 11. The correlation results between the adjacent pixels of the test images are provided in Table 2. From the figure and table, it can be observed that the correlation between neighboring pixels of the encrypted image satisfies zero correlation by effectively encrypt the image, indicating that the encryption method provides promising security.

**Table 2** Correlation coefficients of plain image and ciphered image

| Image | Direction | | |
|---|---|---|---|
| | Horizontal | Vertical | Diagonal |
| House | 0.976027 | 0.957512 | 0.941427 |
| Encrypted house | 0.002455 | 0.002587 | 0.000420 |
| Baboon | 0.904796 | 0.868943 | 0.836901 |
| Encrypted baboon | 0.000553 | -0.000029 | -0.000420 |
| Lena | 0.9774 | 0.962438 | 0.972488 |
| Encrypted Lena | 0.000617 | -0.000535 | -0.000411 |
| Peppers | 0.978912 | 0.968976 | 0.970080 |
| Encrypted Peppers | 0.002505 | 0.001836 | -0.000575 |
| Barbara | 0.916137 | 0.889445 | 0.906666 |
| Encrypted Barbara | 0.001323 | 0.001164 | 0.000961 |
| Parrots | 0.988276 | 0.978517 | 0.977365 |
| Encrypted Parrots | -0.001189 | -0.000545 | 0.000062 |



**(a)** Horizontally adjacent pixels in the red component of Lena image

**(b)** Vertically adjacent pixels in the green component of Lena image

**(c)** Diagonally adjacent pixels in the blue component of Lena image

**(d)** Horizontally adjacent pixels in red component of encrypted image

**(e)** Vertically adjacent pixels in green component of encrypted image

**(f)** Diagonally adjacent pixels in blue component of encrypted image

**Fig. 11** Correlation of adjacent pixels of plain image and cipher image

### 5.5 Information entropy analysis

Information entropy can be applied to quantitatively quantify and calculate the information source's randomness. The entropy $H(m)$ of a signal source $m$ could be calculated using the following equation:

$$H(m) = \sum_{i=0}^{2^N-1} p(m_i) \log \frac{1}{p(m_i)}, \tag{17}$$

where $p(m_i)$ denotes the symbol $m_i$'s probability represents the probability of symbol and the entropy is represented in bits. The information entropy is 8 for an ideal completely random image.

Table 3 calculates the information entropy of the encrypted images and compares them with those of other methods. The information entropy of the encrypted image is quite near to the theory value 8, which implies that the information disclosure during encryption is ignorable and that the cryptosystem is secure.

### 5.6 Differential attack

A safe image cryptography method is supposed to be extremely sensible to any small changes in the plain image, implying that a single pixel variation in the plain image will result in a totally distinct encrypted image.

The differential attack is tested using the uniform average change intensity (UACI) and the number of pixels change rate (NPCR). The following formulas are used to calculate NPCR and UACI:

$$\text{NPCR} = \frac{\sum_{ij} D(i,j)}{M \times N} \times 100\% \tag{18}$$

$$\text{UACI} = \frac{1}{M \times N} [\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}] \times 100\% \tag{19}$$

where $M$ and $N$ denote the rows and columns of the image, $C_1$ and $C_2$ are the encrypted images before and after a pixel change of the plain image, respectively.

We randomly vary the one-bit value of the pixel among any color component, and then evaluate NPCR and UACI in the red, green, and blue components. Table 3 lists the NPCR and UACI performance of this algorithm and compares them with other methods. It can be observed from Table 4 that the NPCR and UACI results are quite near to the theoretical values, and the cryptosystem is effective against both plaintext and differential attacks.

**Table 3** The results of information entropy

| Image | *R* | *G* | *B* |
|---|---|---|---|
| House | 7.9892 | 7.9896 | 7.9892 |
| Baboon | 7.9913 | 7.9916 | 7.9917 |
| Lena | 7.9912 | 7.9913 | 7.9914 |
| Peppers | 7.9915 | 7.9914 | 7.9912 |
| Barbara | 7.9919 | 7.9916 | 7.9918 |
| Parrots | 7.9916 | 7.9915 | 7.9916 |
| Ref. [6] | 7.9973 | 7.9974 | 7.9974 |
| Ref. [9] | 7.99171 | 7.9912 | 7.9917 |
| Ref [10] | 7.9997 | 7.9997 | 7.9997 |
| Ref. [11] | 7.9980 | 7.9979 | 7.9978 |

### 5.7 Noise and data loss attacks

Image data are vulnerable to disturbance and data damage during the transmission process. A highly secure image cryptography algorithm is supposed to be resistant to disturbance and data loss attacks. Different densities of salt and pepper noise are added to the encrypted Lena images. From Fig. 12, it is visible that the decrypted image can be still restored with success and the noise cannot stop us to identify the decrypted image content visually.

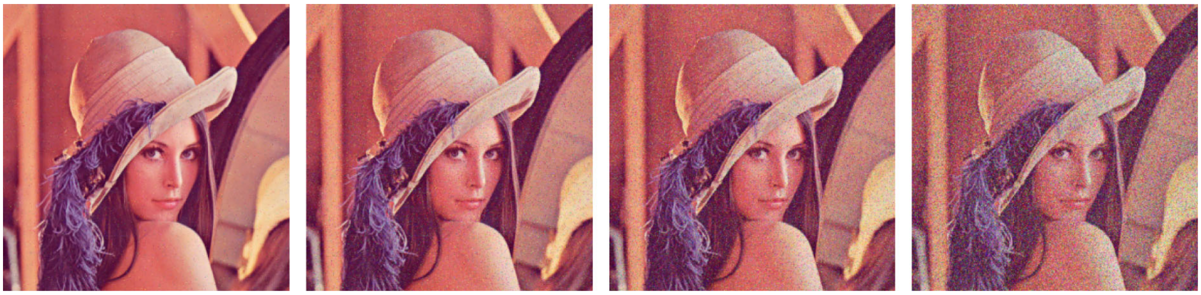
The encrypted image of the lost data and the corresponding decrypted image are illustrated in Fig. 13. Results of the experiments show that the more the image is lost, the more unclear the image is decrypted. However, the image can still be decrypted and recognized although the image is half lost. The algorithm is highly robust to noise attacks and information loss attacks.

### 5.8 Running Performance

To satisfy the fast increase in image data capacity, image cryptography methods should be fast and efficient. In the proposed algorithm, all pixels in the three color components can be completely shuffled through one operation of row and column combined cycle shift scrambling. And it requires only a single

**Table 4** NPCR and UACI values (%)

| Encrypted images | *NPCR (%)* | | *B* | *UACI (%)* | | *B* |
|---|---|---|---|---|---|---|
| | *R* | *G* | | *R* | *G* | |
| House | 99.6567 | 99.5956 | 99.6277 | 33.5193 | 33.3888 | 33.4707 |
| Baboon | 99.5863 | 99.6063 | 99.61 | 33.4855 | 33.4542 | 33.4989 |
| Lena | 99.6243 | 99.6433 | 99.6029 | 33.4686 | 33.5020 | 33.4155 |
| Peppers | 99.6071 | 99.6044 | 99.6143 | 33.4829 | 33.4417 | 33.4553 |
| Barbara | 99.6209 | 99.6091 | 99.6067 | 33.4316 | 33.5033 | 33.4907 |
| Parrots | 99.6137 | 99.6075 | 99.6070 | 33.4772 | 33.4705 | 33.4074 |
| **Average** | **99.6182** | **99.611** | **99.6114** | **33.4775** | **33.46** | **33.4564** |
| Ref [5] | 99.6124 | 99.6277 | 99.6399 | 33.5715 | 33.3356 | 33.4044 |
| Ref [6] | 99.6353 | 99.6309 | 99.6175 | 33.4395 | 33.5495 | 33.5061 |
| Ref [8] | 99.6052 | 99.6120 | 99.6303 | 33.4025 | 33.4428 | 33.5029 |
| Ref [9] | 99.6243 | 99.6218 | 99.6280 | 33.4224 | 33.4361 | 33.4603 |
| Ref [10] | 99.61 | 99.61 | 99.62 | 31.12 | 30.23 | 29.74 |
| Ref [11] | 99.6531 | 99.6522 | 99.6518 | 33.4572 | 33.4715 | 33.4384 |



**(a)** Noise strength 0.01 **(b)** Noise strength 0.05 **(c)** Noise strength 0.1 **(d)** Noise strength 0.25

**Fig. 12** The decryption results with different density noises

turn of permutation and diffusion to realize high security. So our proposed algorithm can exhibit rapid encryption speed.

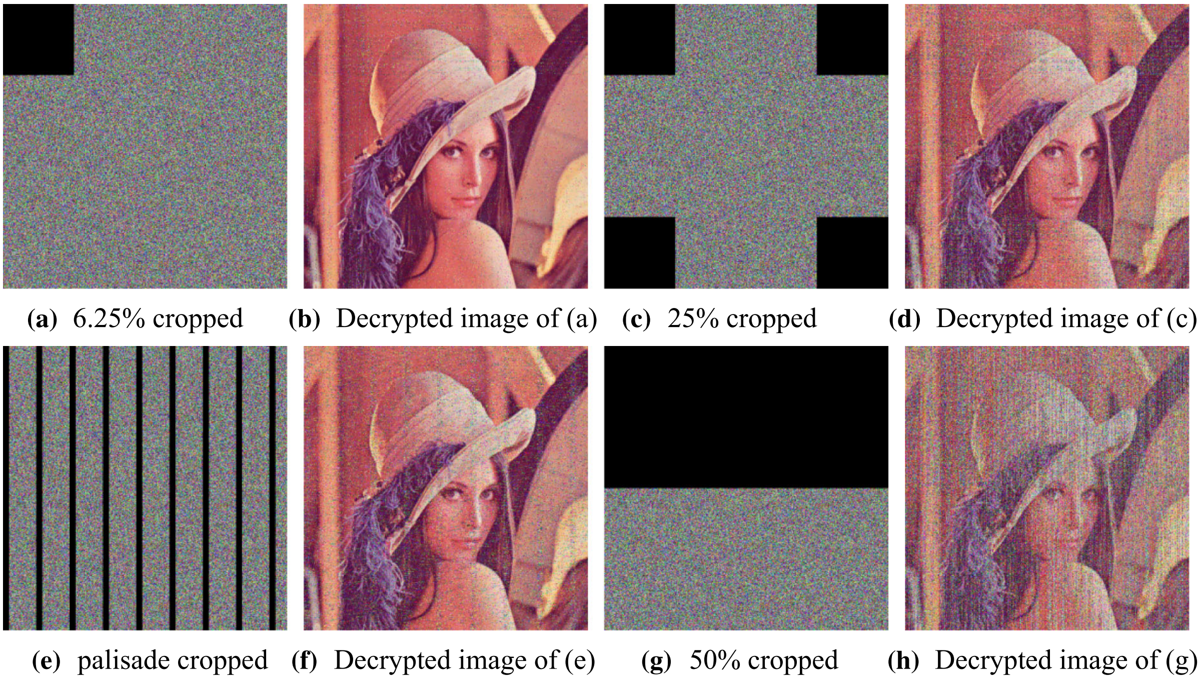
To analyze the calculation complexity of the proposed scheme, assume that the size of the color plain image is $M \times N$. In the permutation process, the time complexity is O(3 $M$ + 8 $N$). During the diffusion process, the complexity of time is O($M \times N \times$ 3). Therefore, the time complexity of the proposed method is O($M \times N$).

To calculate the properties of this method, MATLAB R2017a with Intel(R) Core (TM) i5-8300H CPU@2.30 GHz and 4.0G RAM on Windows 10 OS were utilized. Table 5 lists the running time of the proposed encrypting images. When encryption and decryption of different color images with different sizes, the program can achieve good running performance within a few seconds.

Table 6 indicates the running time of the proposed algorithm compares with other methods when encrypt and decrypt color images of size 512 × 512. Although the experimental environments are different, our proposed algorithm shows relatively good efficiency compared with the similar environment algorithms. It shows that our proposed algorithm is preferred for real-time applications due to its performance in encryption and decryption speed.

## 6 Conclusion

This paper designed a cross 2D hyperchaotic map, which is constructed using one nonlinear function and



(a) 6.25% cropped (b) Decrypted image of (a) (c) 25% cropped (d) Decrypted image of (c)

(e) palisade cropped (f) Decrypted image of (e) (g) 50% cropped (h) Decrypted image of (g)

**Fig. 13** Data loss attack

**Table 5** Execution time analysis (seconds)

| Image | Size | Encryption time | Decryption Time |
|---|---|---|---|
| House | 256 × 256 | 0.459837 | 0.212294 |
| Baboon | 500 × 480 | 1.636521 | 0.771797 |
| Lena | 512 × 512 | 1.769703 | 0.837978 |
| Peppers | 512 × 512 | 1.796105 | 0.847575 |
| Barbara | 720 × 576 | 2.842928 | 1.324155 |
| Parrots | 768 × 512 | 2.700164 | 1.397047 |

**Table 6** Compared with other methods for execution time (seconds)

| Methods | Encryption time | Decryption time | Experiment environment |
|---|---|---|---|
| Proposed | 1.769703 | 0.837978 | 2.30 GHz CPU, 4.0 G RAM |
| Ref [1] | 20.915818 | 18.802360 | 1.90 GHz CPU, 4.0 G RAM |
| Ref [3] | 1.3408 | 1.0158 | 3.45 GHz CPU, 8.0 G RAM |
| Ref [4] | 3.9593 | 13.1574 | 3.30 GHz CPU, 4.0 G RAM |
| Ref [8] | 0.3239 | - | 2.4 GHz CPU, 2.0 G RAM |
| Ref [11] | 9.0016 | 9.1095 | 2.7 GHz CPU, 8.0 G RAM |

two chaotic maps with cross structure. Numerous experiments such as bifurcation diagrams, Lyapunov exponent spectra, and phase portraits are carried out to illustrate the complex chaotic behavior of the proposed hyperchaotic system. The simulation outcomes show that this nonlinear crossed two-dimensional hyperchaotic system provides good chaotic performance. In the color image encryption algorithm, the keys are generated using hash function SHA-512 and the information of the plain color image. First, the color plain image is converted to a combined bit-level matrix and permutated by the chaotic-based row and column combined cycle shift scrambling method. Then, the scrambled integer matrix is diffused according to the selecting sequence. The cipher color image is obtained by decomposed the diffused matrix. The proposed encryption algorithm makes the three color components of color image affect one another to eliminate the correlations between them. Through the comparison of encryption performance, this proposed algorithm has been proven to have enough keyspace, a sensitive key, resistance to statistical attacks and differential attacks, and is applicable for real-time implementations.

**Funding** The work is funded by the Postdoctoral Research Foundation of China (No: 2020M680933), the National Natural Science Foundation of China (Nos: 61701070, 61672124), the Doctoral Start-up Foundation of Liaoning Province (No: 2018540090), Key R&D Projects of Liaoning Province (No: 2019020105-JH2/103), Liaoning Province Science and Technology Innovation Leading Talents Program Project (No: XLYC1802013), Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (No: MIMS20-M-02), Jinan City '20 universities' Funding Projects Introducing Innovation Team Program (No: 2019GXRC031).

**Data availability** Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

**Declarations**

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

## References

1. Xiong, Z.G., Wu, Y., Ye, C.H., Zhang, X.M., Xu, F.: Color image chaos encryption algorithm combining CRC and nine palace map. Multimed. Tools Appl. **78**(22), 31035–31055 (2019)
2. Sneha, P.S., Sankar, S., Kumar, A.S.: A chaotic colour image encryption scheme combining Walsh-Hadamard transform and Arnold-Tent maps. J. Ambient. Intell. Humaniz. Comput. **11**(3), 1289–1308 (2020)
3. Kaur, G., Agarwal, R., Patidar, V.: Color image encryption scheme based on fractional Hartley transform and chaotic substitution-permutation. Vis. Comput., (3):1–24, (2021).
4. Chai, X., Bi, J., Gan, Z., et al.: Color image compression and encryption scheme based on compressive sensing and double random encryption strategy. Signal Process. **176**, 107684 (2020)
5. Pak, C., An, K., Jang, P., et al.: A novel bit-level color image encryption using improved 1D chaotic map. Multimed. Tools Appl. **78**, 12027–12042 (2019)
6. Zhou, J., Zhou, N.R., Gong, L.H.: Fast color image encryption scheme based on 3D orthogonal Latin squares and matching matrix. Opt. Laser Technol. **131**, 106437 (2020)
7. Wang, L., Ran, Q., Ma, J.: Double quantum color images encryption scheme based on DQRCI. Multimed. Tools Appl. **79**(9–10), 6661–6687 (2020)
8. Li, Z., Peng, C., Tan, W.: A novel chaos-based color image encryption scheme using bit-level permutation. Symmetry **12**(9), 1497 (2020)
9. Zhang, Y.Q., He, Y., Li, P., et al.: A new color image encryption scheme based on 2DNLCML system and genetic operations. Opt. Lasers Eng. **128**, 106040 (2020)
10. Wen, W., Wei, K., Zhang, Y., et al.: Colour light field image encryption based on DNA sequences and chaotic systems. Nonlinear Dyn. **99**(2), 1587–1600 (2020)
11. Kang, X., Guo, Z.: A new color image encryption scheme based on DNA encoding and spatiotemporal chaotic system. Signal Process. Image Commun. **80**, 115670 (2020)
12. Wang, S.C., Wang, C.H., Xu, C.: An image encryption algorithm based on a hidden attractor chaos system and the Knuth-Durstenfeld algorithm. Opt. Lasers Eng. **128**, 105995 (2020)
13. Nestor, T., Kengne, J., Abd-El-Atty, B., et al.: Design and implementation of a simple dynamical 4-D chaotic circuit with applications in image encryption. Inf. Sci. **515**, 191–217 (2020)
14. Wang, X.Y., Gao, S.: Image encryption algorithm for synchronously updating Boolean networks based on matrix semi-tensor product theory. Inf. Sci. **507**, 16–36 (2020)
15. Farah, M.A.B., Guesmi, R., Kachouri, A., et al.: A Novel Chaos Based Optical Image Encryption Using fractional Fourier transform and DNA Sequence Operation. Opt. Laser Technol. **121**, 105777 (2019)
16. Xian, Y.J., Wang, X.Y., Yan, X.P., Li, Q., Wang, X.Y.: Image encryption based on chaotic sub-block scrambling and chaotic digits selection diffusion. Opt. Lasers Eng. **134**, 106202 (2020)
17. He, Y., Zhang, Y.Q., Wang, X.Y.: A new image encryption algorithm based on two-dimensional spatiotemporal chaotic system. Neural Comput. Appl. **32**, 247–260 (2020)
18. Liu, H., Zhang, Y., Kadir, A., et al.: Image encryption using complex hyper chaotic system by injecting impulse into parameters. Appl. Math. Comput. **360**, 83–93 (2019)
19. Wang, X.Y., Feng, L., Li, R., Zhang, F.C.: A fast image encryption algorithm based on non-adjacent dynamically coupled map lattice model. Nonlinear Dyn. **95**(4), 2797–2824 (2019)
20. Wang, X.Y., Sun, H.H.: A chaotic image encryption algorithm based on improved Joseph traversal and cyclic shift function. Opt. Laser Technol. **122**, 105854 (2020)
21. Zhu, C., Sun, K.: Cryptanalyzing and Improving a novel color image encryption algorithm using rt-enhanced chaotic tent maps. IEEE Access **6**, 18759–18770 (2018)
22. Farajallah, M., Assad, S.E., Deforges, O.: Cryptanalyzing an image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion. Multimed. Tools Appl. **77**(21), 28225–28248 (2018)
23. Ge, X., Lu, B., Liu, F., et al.: Cryptanalyzing an image encryption algorithm with compound chaotic stream cipher based on perturbation. Nonlinear Dyn. **90**(2), 1141–1150 (2017)
24. Wen, H., Yu, S., Lü, J.H.: Breaking an image encryption algorithm based on DNA encoding and spatiotemporal chaos. Entropy **21**(3), 246 (2019)
25. Rössler, O.E.: An equation for hyperchaos. Phys. Lett. A **71**, 155–157 (1979)

26. Zhu, S.Q., Zhu, C.X.: Secure image encryption algorithm based on hyperchaos and dynamic DNA coding. Entropy **22**(7), 772 (2020)
27. Xu, C., Sun, J., Wang, C.: An image encryption algorithm based on random walk and hyperchaotic systems. Int J Bifurcation Chaos **30**(4), 2129–2151 (2020)
28. Bouslehi, H., Seddik, H.: Innovative image encryption scheme based on a new rapid hyperchaotic system and random iterative permutation. Multimed. Tools Appl. **77**(23), 1–23 (2018)
29. Cheng, G., Wang, C., Chen, H.: A novel color image encryption algorithm based on hyperchaotic system and permutation-diffusion architecture. Int. J. Bifurcation Chaos **29**(09), 1950115 (2019)
30. Kaur, M., Singh, D., Sun, K., et al.: Color image encryption using non-dominated sorting genetic algorithm with local chaotic search based 5D chaotic map. Futur. Gener. Comput. Syst. **107**, 333–350 (2020)
31. Zhou, M., Wang, C.: A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks. Signal Process. **171**, 107484 (2020)
32. Ouyang, X., Luo, Y., Liu, J., et al.: A color image encryption method based on memristive hyperchaotic system and DNA encryption. Int. J. Mod. Phys. B **34**(4), 2050014 (2020)
33. Hua, Z.Y., Zhou, Y.C., Pun, C.M., Chen, C.L.P.: 2D sine logistic modulation map for image encryption. Inf. Sci. **297**, 80–94 (2015)
34. Hua, Z.Y., Zhou, Y.C.: Image encryption using 2D Logistic-adjusted-Sine map. Inf. Sci. **339**, 237–253 (2016)
35. Liu, W., Sun, K., et al.: A fast image encryption algorithm based on chaotic map. Opt. Lasers Eng. **84**, 26–36 (2016)
36. Cao, W., Mao, Y., Zhou, Y.: Designing a 2D infinite collapse map for image encryption. Signal Process. **171**, 107457 (2020)
37. Wang, M.X., Wang, X.Y., Zhao, T.T., Zhang, C., Xia, Z.Q., Yao, N.M.: Spatiotemporal chaos in improved Cross Coupled Map Lattice and its application in a bit-level image encryption scheme. Inf. Sci. **544**, 1–24 (2021)
38. He, D., He, C., Jiang, L.G., et al.: Chaotic characteristics of a one-dimensional iterative map with infinite collapses. IEEE Trans. Circuits Syst. I Fundament. Theory Appl. **48**(7), 900–906 (2001)
39. Alvarez, G., Li, S.: Some basic cryptographic requirements for chaos-based cryptosystems. Int. J. Bifurcation Chaos **16**(8), 2129–2151 (2006)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.