Multimedia Tools and Applications (2022) 81:24757–24776
https://doi.org/10.1007/s11042-022-12479-x

Multi-image encryption algorithm based on wavelet transform and 3D shuffling scrambling

Huiyan Zhong[1] · Guodong Li[1]

Received: 9 April 2021 / Revised: 6 July 2021 / Accepted: 25 January 2022 /
Published online: 21 March 2022

Abstract
To solve the problems of low efficiency and poor resistance to attack, this paper proposes the multi-image encryption algorithm based on Haar wavelet transform and 3D shuffling scrambling. Taking advantage of the high computational efficiency of low-dimensional chaotic maps, this paper designs the dynamic pseudo-random sequence generator based on a dynamic chaotic library with three one-dimension chaotic maps and the roulette algorithm, which is highly associated with the plaintext image. To better scramble the image cube, this paper proposes a 3D shuffling scrambling algorithm, which divides the cube into one-dimensional vectors and scrambles the order of the one-dimensional vectors, then reorganizes the cube. To encrypt multiple images, first, reconstruct the images into an image cube and perform wavelet transformation on each layer of the cube. Then, use the 3D shuffling algorithm to scramble the low-frequency coefficient and reconstruct the cube with the scrambled low-frequency coefficient and high-frequency parts. Last, the chaotic matrix is XOR with each layer of the image cube. The algorithm can encrypt grayscale or color images of any size, which is flexible. In the simulation experiments, the algorithm has ideal ciphertext statistical characteristics, high running speed, and the ability of anti-attack, which is better than encryption algorithm in other references.

Keywords Multi-image encryption · Wavelet transform · 3D shuffling scrambling · Dynamic chaos library

1 Introduction

With the development of computer techniques, massive amounts of information are being transmitted, stored and shared every day. In recent years, people have paid more and more

✉ Guodong Li
lgdzhy@126.com

[1] School of Mathematics and Computational Science, Guilin University of Electronic Technology, Guilin 541004 Guangxi, China

attention to information security. Many fields produce a large number of images carrying confidential information every day. To ensure the security of image content, the mainstream methods are image encryption and digital steganography. Image encryption uses pixel scrambling and diffusion to turn images into meaningless snowflake images [2–8, 10–12, 14–30], while digital steganography protects image security by hiding images [1, 9, 13]. This article mainly studies image encryption.

In image encryption algorithms, chaotic systems are usually used for encryption due to their unpredictability and sensitivity [2, 3, 5–8, 10–12, 14–27]. Some scholars study the encryption of images in the time domain [2, 4, 7, 16, 17, 19–21, 25–27]. Guodong Li et al. used 5-D Cellular Neural Networks and Logistics chaos to scramble and diffuse the image [16]. Some scholars study image encryption in the frequency domain [3, 5, 12, 14, 15, 18, 22, 24]. Fourier transform, wavelet transform, Fresnel transform and other transformations are used to transform the image and the encryption is carried out in the frequency domain. Feng et al. proposed the encryption algorithm based on adaptive wavelet chaos, which used the convex optimization-particle swarm optimisation algorithm to enhance the adaptive ability of wavelet transform and scramble low-frequency coefficients. This algorithm has a stronger ability to resist attacks, but the decrypted image is a little distorted [3]. Guodong Li et al. used fractional Fourier transform to transform the image to the frequency domain, and used double chaos systems to scramble and diffuse pixels in the frequency domain [15]. Shakir H. R et al. proposed an encryption algorithm based on selective AES and wavelet transform. First, the plaintext image was decomposed by wavelet, and then the low-frequency coefficients were encrypted by the AES algorithm. After the image was reconstructed, the image was scrambled by chaos [22]. However, the encryption algorithms only aim at a single image, which is low efficient when encrypting multiple images.

With the increase of image data, the traditional single-image encryption algorithm has the problem of low efficiency in practice. Multi-image encryption has attracted the attention of scholars [11, 12, 17, 20, 21, 24, 25]. Sahasrabuddh A et al. proposed a multi-image encryption algorithm based on 3D scrambling and hyperchaotic systems, which let multiple ordinary images generate a 3D image, and use chaos theory and the concept of elliptic curve Elgamar cryptographic system to generate cryptographic images and shared keys. This algorithm has ideal cryptographic statistics and is highly efficient [21]. Lei Zhang et al. proposed a multi-image encryption algorithm based on bit plane. The algorithm decomposed k images into 8 k bit plane, scrambled the high pixel bit and performed XOR on the low pixel bit [25]. K. Patro et al. proposed an algorithm for multiple grey image encryption using cross-coupled chaotic mapping. This algorithm uses the cross-coupled chaotic sequence to improve the security level and has high efficiency [20]. Some scholars have studied multi-image encryption in the frequency domain. Chun-Lai Li proposed a robust chaotic mapping multi-image encryption algorithm in the wavelet domain. Multiple ciphertexts are pieced together into an image, then the image is decomposed by the discrete wavelet. The low-frequency coefficients are scrambled by cat mapping, and the image is reconstructed. The algorithm has a certain degree of anti-attack ability, but the article does not test the efficiency of the algorithm [17]. However, the existing multi-image encryption algorithms rarely transform 3D images into the frequency domain for encryption.

Inspired by the work of predecessors, for the problems of low efficiency of multi-image encryption algorithm, poor anti-attack, distortion of decrypted images, limited image shape and so on, this paper proposes an efficient multi-image encryption algorithm based on wavelet transform and three-dimensional shuffling scrambling. First, design the dynamic pseudo-random sequence generator based on a dynamic chaotic library with three one-dimension chaotic maps and the roulette algorithm, which is highly associated with the plaintext image. Then, the multiple images are divided into blocks and then reconstructed into an image cube. The wavelet transform is performed on each layer of the image cube and the low-frequency coefficients are scrambled by the 3D shuffle algorithm. Finally, use the scrambled low-frequency coefficients and high-frequency parts to reconstruct the cube. After reconstruction, the chaotic matrix is used to perform the bitwise XOR operation with each layer of the image cube to obtain the final encrypted image. The algorithm has ideal ciphertext statistical characteristics, high running speed, and anti-attack.

Section 2 of this paper will mainly explain the design of the multi-image encryption algorithm based on wavelet transform and how to construct an image cube. Section 3 explains the design of the dynamic pseudo-random sequence generator based on roulette algorithm. Section 4 explains the design of the 3D scrambling algorithm based on the shuffling algorithm. Section 5 gives the specific steps of the encryption algorithm. Section 6 performs simulation experiments, analyzes the proposed image encryption algorithm and compares it with the mainstream encryption algorithms. Finally, the last section summarizes and discusses the full text.

2 Preliminaries

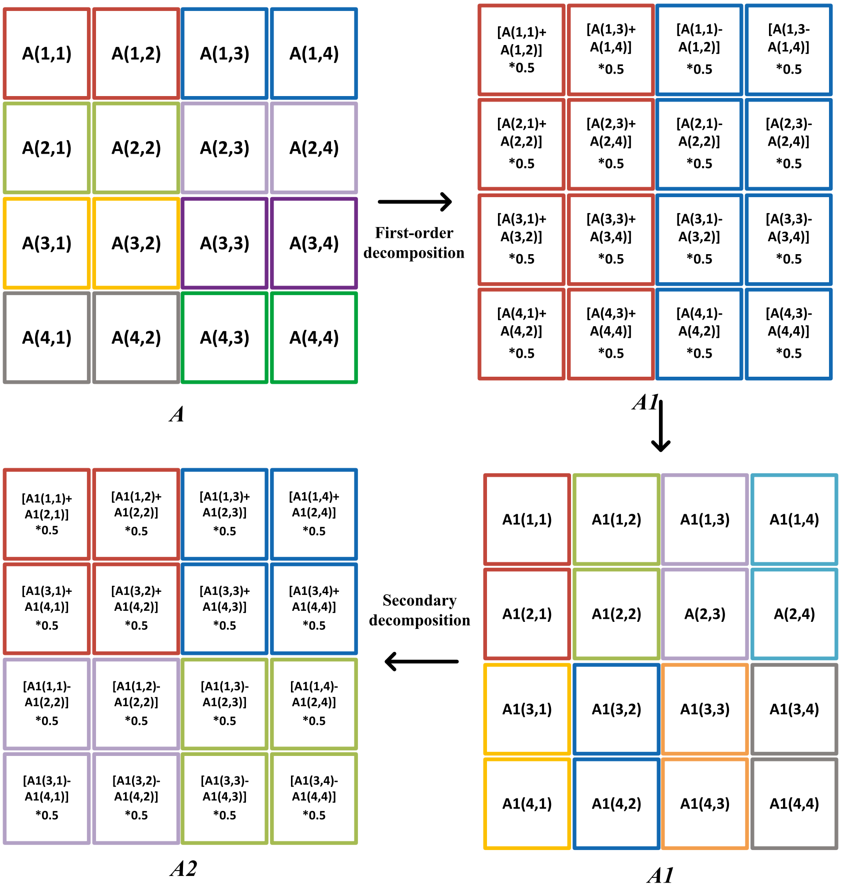
2.1 Haar wavelet transform

Wavelet transform is a transform analysis method, which is developed from Fourier transform. It improves the limitation that Fourier transform cannot change the window size and has no adaptive ability. Therefore, it is favoured by many scholars and is widely used in various fields. After the image is subjected to the two-dimensional discrete wavelet transform, four sub-images are obtained. LL is the low-frequency part that contains the most information of the original image, while HL, LH and HH are detail parts in horizontal, vertical and diagonal directions respectively.

The proposed encryption algorithm uses discrete two-dimensional Haar wavelet transform on the image. As shown in Fig. 1, the 4×4 matrix $\boldsymbol{A}$ is decomposed by Haar wavelet. First, the average between the adjacent elements in each row is placed to the left of each row in order. Then, the adjacent elements in each row are differentiated and divided by two and the resulting values are placed on the left side of each row in order. Each row is treated in this way to obtain matrix $\boldsymbol{A1}$. The red part on the left side of $\boldsymbol{A1}$ is the low-frequency part L, and the blue part on the right side is the high-frequency detail part H. Perform corresponding calculations on each column of $\boldsymbol{A1}$ to obtain the wavelet two-level decomposition matrix $\boldsymbol{A2}$. The red part in the upper left corner of $\boldsymbol{A2}$ is the low-frequency part LL and the remaining parts are the high-frequency detail parts LH, HL, LL.

2.2 Construct image cube

The proposed algorithm requires preprocessing of the image shape. Namely, it reconstitutes multiple images to be encrypted into an image cube. In this step, the images need to be divided into blocks and overlapped. Suppose the images to be encrypted are $P_1, P_2, \ldots, P_n$, and the size are $m_1 \times n_1, m_2 \times n_2, \ldots, m_n \times n_n$ respectively. The amount of data is $M = (m_1 \times n_1)(m_2 \times n_2)\cdots(m_n \times n_n)$. Set the size of the cube is $a \times a \times b$. Calculate a, b such that $M \approx a \times a \times b$. Since the proposed encryption algorithm encrypts the low-frequency coefficients of the image



A

A(1,1)	A(1,2)	A(1,3)	A(1,4)
A(2,1)	A(2,2)	A(2,3)	A(2,4)
A(3,1)	A(3,2)	A(3,3)	A(3,4)
A(4,1)	A(4,2)	A(4,3)	A(4,4)

A1

[A(1,1)+ A(1,2)] *0.5	[A(1,3)+ A(1,4)] *0.5	[A(1,1)- A(1,2)] *0.5	[A(1,3- A(1,4)] *0.5
[A(2,1)+ A(2,2)] *0.5	[A(2,3)+ A(2,4)] *0.5	[A(2,1)- A(2,2)] *0.5	[A(2,3)- A(2,4)] *0.5
[A(3,1)+ A(3,2)] *0.5	[A(3,3)+ A(3,4)] *0.5	[A(3,1)- A(3,2)] *0.5	[A(3,3)- A(3,4)] *0.5
[A(4,1)+ A(4,2)] *0.5	[A(4,3)+ A(4,4)] *0.5	[A(4,1)- A(4,2)] *0.5	[A(4,3)- A(4,4)] *0.5

A1

A1(1,1)	A1(1,2)	A1(1,3)	A1(1,4)
A1(2,1)	A1(2,2)	A(2,3)	A(2,4)
A1(3,1)	A1(3,2)	A1(3,3)	A1(3,4)
A1(4,1)	A1(4,2)	A1(4,3)	A1(4,4)

A2

[A1(1,1)+ A1(2,1)] *0.5	[A1(1,2)+ A1(2,2)] *0.5	[A1(1,3)+ A1(2,3)] *0.5	[A1(1,4)+ A1(2,4)] *0.5
[A1(3,1)+ A1(4,1)] *0.5	[A1(3,2)+ A1(4,2)] *0.5	[A1(3,3)+ A1(4,3)] *0.5	[A1(3,4)+ A1(4,4)] *0.5
[A1(1,1)- A1(2,2)] *0.5	[A1(1,2)- A1(2,2)] *0.5	[A1(1,3)- A1(2,3)] *0.5	[A1(1,4)- A1(2,4)] *0.5
[A1(3,1)- A1(4,1)] *0.5	[A1(3,2)- A1(4,2)] *0.5	[A1(3,3)- A1(4,3)] *0.5	[A1(3,4)- A1(4,4)] *0.5

Fig. 1 Discrete two-dimensional Haar wavelet transform

cube, if the image block is too small, it will cause image information loss, so set a, b to make a as large as possible and $|\frac{a}{4}-b|$ as small as possible. If $a \times a \times b \neq M$, it can be filled with 0 to construct a cube. The constructed image cube is shown in Fig. 2.

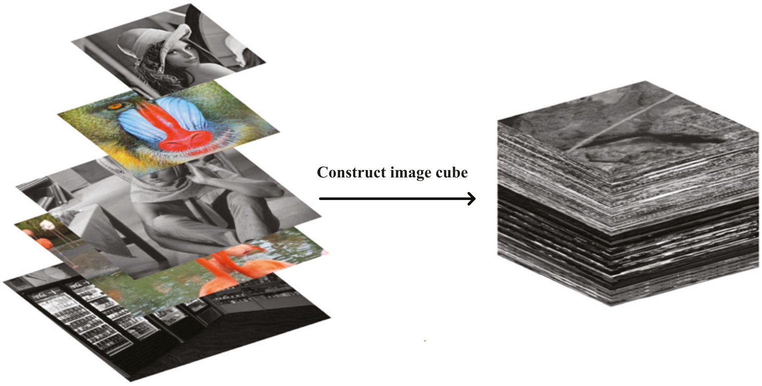


Fig. 2 Image cube

3 The design of dynamic pseudo-random sequence generator based on roulette algorithm

This section explains the design of the dynamic pseudo-random sequence generator based on the roulette algorithm and a dynamic chaotic library with three kinds of one-dimension chaotic maps that are highly efficient and have good complexity. And the pseudo-random sequence is highly associated with the plaintext image.

3.1 The design of dynamic chaos library

Chaotic sequences that are used for image encryption are required to have sufficient complexity and ergodicity, and at the same time, computation efficiency must be taken into consideration. Based on these considerations, select three new types of one-dimensional chaos to construct the dynamic chaos library, improved Henon map [26], Hybrid map [6] and L-F cascade map [18].

The optimised Henon mapping has higher complexity and larger parameter space than the traditional Henon mapping, as shown in formula (1).

$$\begin{cases} x_{n+1} = 1{-}a\cos x_n^6 + y_n \\ y_{n+1} = bx_n \end{cases} \tag{1}$$

When $a \in [0,1.4]$, $b \in [0,0.3]$, the system is in chaos.

Hybrid mapping is a chaotic mapping composed of Logistic mapping, Tent mapping, and Sin mapping. It has good complexity and uniformity. The formula is shown in Eq. (2).

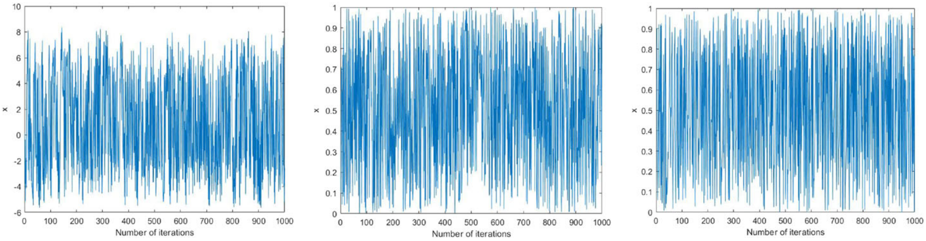
$$\begin{cases} x_{n+1} = \left(r^{11}/4{*}\sin\left(\pi r^2/2\right)(1{-}x_n)\right) mod 1, rx_n(1{-}x_n) \le 1/2 \\ x_{n+1} = \Big(r^{11}/4{*}\sin\Big(\pi r^2/2{*}(1{-}rx_n(1{-}x_n))mod 1, rx_n(1{-}x_n) > 1/2 \end{cases} \tag{2}$$

Where, when $r \in [1.4,4]$, the system is in chaos.

L-F cascade chaos overcomes the blank area of traditional Logistic mapping and it makes the distribution of parameters more uniform. The formula is as Eq. (3) follows.

$$\begin{cases} x_{n+1} = \mu x_n(1{-}x_n) \\ F_n = (x_{n-1}F_{n-1} + x_{n-2}F_{n-2} + x_{n-3}F_{n-3}) mod 1 \end{cases} \tag{3}$$

When $\mu \in [1.5,4]$, the system is in chaos.



(a) Optimize Henon chaotic sequence (b) Hybrid mapping sequence (c) L-F cascade chaotic sequence

Fig. 3 Chaotic sequence. **a** Optimize Henon chaotic sequence, **b** Hybrid mapping sequence, **c** L-F cascade chaotic sequence

It can be observed from Fig. 3 that these three chaotic systems have good ergodicity and randomness.

Chaos library is composed of the above three types of chaos. Each chaotic mapping theory generates three chaos systems with different parameters. The initial values of all chaos systems are determined by the hash value of plaintext. There are twelve chaotic systems in total, as shown in formula (4).

$$\begin{cases} C_1 : x_{n+1} = 1{-}4.7\cos x_n^6 + y_n, y_{n+1} = 0.3x_n \\ C_2 : x_{n+1} = 1{-}4.8\cos x_n^6 + y_n, y_{n+1} = 0.28x_n \\ C_3 : x_{n+1} = 1{-}4.9\cos x_n^6 + y_n, y_{n+1} = 0.27x_n \\ C_4 : x_{n+1} = 1{-}5.0\cos x_n^6 + y_n, y_{n+1} = 0.26x_n \\ C_5 : \begin{cases} x_{n+1} = \big(3.4^{11}/4^{*} \sin\big(\pi\, 3.4^2/2\big)(1-x_n)\big) \bmod 1, 3.4_{x_n}(1-x_n) \le 1/2 \\ x_{n+1} = \big(3.4^{11}/4{*}\sin\big(\pi\, 3.4^2/2{*}(1-3.4_{x_n}(1-x_n))\big)\big) \bmod 1, 3.4_{x_n}(1-x_n) > \frac{1}{2} \end{cases} \\ C_6 : \begin{cases} x_{n+1} = \big(3.6^{11}/4{*}\sin\big(\pi\, 3.6^2/2\big)(1-x_n)\big) \bmod 1, 3.6_{x_n}(1-x_n) \le 1/2 \\ x_{n+1} = \big(3.6^{11}/4{*}\sin\big(\pi\, 3.6^2/2{*}(1-3.6_{x_n}(1-x_n))\big) \bmod 1, 3.6_{x_n}(1-x_n) > \frac{1}{2} \end{cases} \\ C_7 : \begin{cases} x_{n+1} = \big(3.8^{11}/4{*}\sin\big(\pi\, 3.8^2/2\big)(1-x_n)\big) \bmod 1, 3.8_{x_n}(1-x_n) \le 1/2 \\ x_{n+1} = \big(3.8^{11}/4{*}\sin\big(\pi\, 3.8^2/2{*}(1-3.8_{x_n}(1-x_n))\big) \bmod 1, 3.8_{x_n}(1-x_n) > \frac{1}{2} \end{cases} \\ C_8 : \begin{cases} x_{n+1} = \big(3.9^{11}/4{*}\sin\big(\pi\, 3.9^2/2\big)(1-x_n)\big) \bmod 1, 3.9_{x_n}(1-x_n) \le \frac{1}{2} \\ x_{n+1} = \big(3.9^{11}/4{*}\sin\big(\pi\, 3.9^2/2{*}(1-rx_n(1-x_n))\big) \bmod 1, 3.9x_n = (1-x_n\ ) > 1/2 \end{cases} \\ C_9 : F_n = (x_{n-1}F_{n-1} + x_{n-2}F_{n-2} + x_{n-3}F_{n-3}) \bmod 1, x_{n+1} = 3.7x_n(1-x_n) \\ C_{10} : F_n = (x_{n-1}F_{n-1} + x_{n-2}F_{n-2} + x_{n-3}F_{n-3}) \bmod 1, x_{n+1} = 3.8x_n(1-x_n\ ) \\ C_{11} : F_n = (x_{n-1}F_{n-1} + x_{n-2}F_{n-2} + x_{n-3}F_{n-3}) \bmod 1, x_{n+1} = 3.9x_n(1-x_n\ ) \\ C_{12} : F_n = (x_{n-1}F_{n-1} + x_{n-2}F_{n-2} + x_{n-3}F_{n-3}) \bmod 1, x_{n+1} = 3.99x_n(1-x_n\ ) \end{cases} \tag{4}$$

3.2 Roulette algorithm

Roulette algorithm is an algorithm based on the probability of individual choice. This paper uses the roulette algorithm to select the chaotic system from the chaos library that generates the pseudo-random sequence for encryption. Suppose k chaos systems are selected from n chaos systems (C_1, C_2, $\cdots$, C_n). Set their individual fitness as (s_1, s_2, $\cdots$, s_n). Specific steps are as follows:

Step 1 Calculate the probabilities of each chaos system: $p_i = s_i / \sum_{k=1}^{n} s_k, i \in [1, n]$.

Step 2 Calculate the cumulative probabilities of each chaos system: $q_i = \sum_{k=1}^{i} p_k, i \in [1, n]$.

Step 3 Generate a random sequence in the range (r_1, r_2, $\cdots$, r_k).

Step 4 Judge the interval of the cumulative probability of the random number $\boldsymbol{r}$, and output the selected individual C_i. The rules are as formula (5) shows.

$$\begin{cases} i = 1, r_i < q_1 \\ i = j, q_j < r_i < q_{j+1} \end{cases} \tag{5}$$

3.3 The design of the dynamic pseudo-random sequence generator based on roulette algorithm

The specific steps are as follows. Suppose the plaintext image is ***P***.

Step 1 Calculate the hash values of the plaintext image by the SHA-256 algorithm, and get a hexadecimal string of length 64. The two adjacent hexadecimal characters are converted into a decimal number, and a sequence of 32 is obtained, denoted as ***H***.

Step 2 Calculate the sequence x_0 by the first 20 digits of ***H***, $H(1\ :\ 20)$, the specific calculation formula is as formula (6).

$$x_0 = \mathrm{mod}\left(H(1:20)*10^{-3}, 1\right). \tag{6}$$

Step 3 Calculate the fitness of the individual $(s_1,\ s_2,\ \cdots,\ s_{12})$ in the roulette algorithm through the middle 12 digits of the sequence ***H***, $H(12\ :\ 23)$.

Step 4 Suppose n chaotic sequences need to be generated. Calculate $n_1 = n/32$,$R = 1/n$. The random number $(r_1,\ r_2,\ \cdots,\ r_n)$ used for selection in the roulette algorithm is calculated as formula (7).

$$r_i = \mathrm{mod}(sum(H((i{-}1)\cdot n_1 : i\cdot n_1))\cdot(i{-}1)R, 1), i{\in}[1, 2, \cdots n]. \tag{7}$$

Where, *sum*() is the sum function.

Step 5 Input individual fitness $(s_1,\ s_2,\ \cdots,\ s_{12})$ and selection random number $(r_1,\ r_2,\ \cdots,\ r_n)$ to the roulette algorithm and get the chaos systems $(C_1,\ C_2,\ \cdots,\ C_n)$. Generate n chaotic sequences by n chaotic systems respectively.

4 The design of 3D scrambling algorithm based on shuffling algorithm

4.1 Knuth–Durstenfeld shuffling algorithm

The Knuth–Durstenfeld shuffle algorithm is improved from the Fisher-Yates shuffle algorithm, which uses a random sequence to sort an array. Suppose the array to be sorted is ***X*** and the length is n. The specific steps are as follows.

Step 1 The length of the array ***X*** is n. Set k = n;

Step 2 Generate a random number p from [0,k-1] and exchange the value of subscript p with the value of subscript k in the array ***X***.

Step 3 Set k = k-1. Repeat steps 2 and 3;

Step 4 Until all the elements in ***X*** are processed, a new array ***X*** is obtained.

4.2 The design of 3D shuffling scrambling algorithm based on the Knuth-Durstenfeld shuffling algorithm

The 3D shuffling scrambling algorithm uses the Knuth–Durstenfeld shuffling algorithm to scramble the pixels in the image cube. The specific steps are as follows:

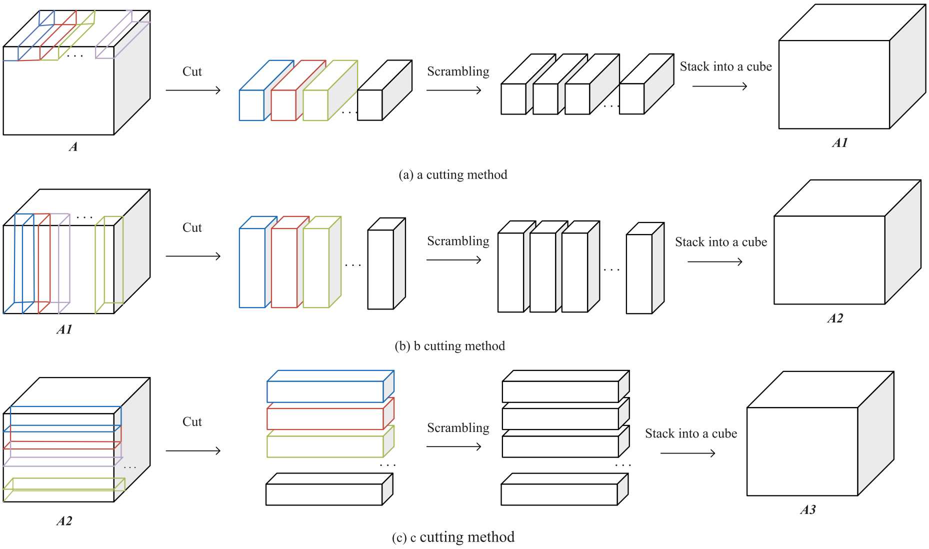
(1) The image cube ***A*** was cut into a series of one-dimensional vectors according to the cutting method in Fig. 4a. The one-dimensional vectors were arranged into matrices in order. Then chaotic sequence ***C1*** and Knuth-Durstenfeld shuffling algorithm were used to scramble the matrix by column, and the scrambled matrices were then stacked into cube ***A1*** in order.

(2) The image cube ***A1*** was cut into a series of one-dimensional vectors according to the cutting method in Fig. 4b. The one-dimensional vectors were arranged into matrices in order. Then chaotic sequence ***C2*** and Knuth-Durstenfeld shuffling algorithm were used to scramble the matrix by column, and the scrambled matrices were then stacked into cube ***A2*** in order.

(3) The image cube ***A2*** was cut into a series of one-dimensional vectors according to the cutting method in Fig. 4c. The one-dimensional vectors were arranged into matrices in order. Then chaotic sequence ***C3*** and Knuth-Durstenfeld shuffling algorithm were used to scramble the matrix by row, and the scrambled matrices were then stacked into cube ***A3*** in order.

5 The proposed algorithm

Assume that the image that needs to be encrypted is P_1, P_2, …, P_n. The encryption step flowchart is shown in Fig. 5. The specific encryption steps are as follows:



(a) a cutting method

(b) b cutting method

(c) c cutting method

Fig. 4 Image cube cutting method. **a** cutting method, **b** cutting method, **c** cutting method

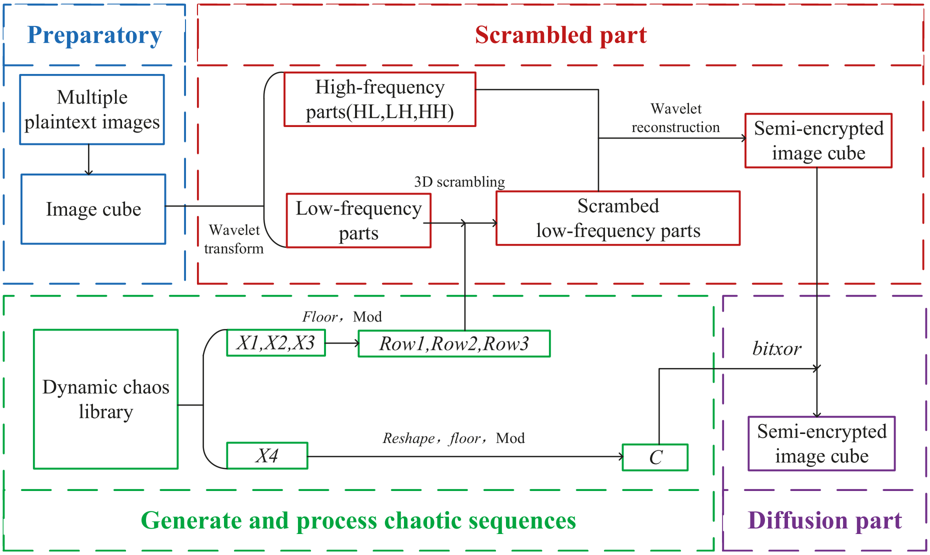


Fig. 5 Encryption step flow

Step 1 Reconstitute the images P_1, P_2, …, P_n into an image cube ***Im*** as subsection 2.2, assuming the size is ***a*** × ***a*** × ***b***. Record the size of each image as the key.

Step 2 Calculate the hash value of the image cube ***Im***, and input it into the dynamic pseudo-random sequence generator in subsection 3.3 to produce pseudo-random sequences ***×1, ×2, ×3, ×4***, whose lengths are ***(a/4)*b-1***, ***(a/4)* (a/4)-1***, ***(a/4)*b-1*** and ***a*** × ***a*** × ***b*** respectively. Record the hash value as the key.

Step 3 Perform Haar wavelet transform on each layer of the image cube ***Im*** to obtain the low-frequency part of the cube ***LL*** and other high-frequency detail parts of the cube ***LH***, ***HL***, ***HH***.

Step 4 Calculate ***row1***, ***row2***, ***row3*** according to algorithm 1.

```
Algorithm 1
Input: x
Ouput: row
s=size(x)
for i=1:s
    row(i)=mod(floor(x(i)*10^14),s-i+1)+1;
end
```

Let ***row1***, ***row2***, and ***row3*** be the chaotic sequence of the shuffling algorithm in subsection 4.3. After cutting the cube, row/column vectors are arranged in order to form a matrix, and the row/column scrambled calculation is shown in Algorithm 2, supposed the matrix is ***LL1*** and scrambled its column. Perform three rounds of cutting, row or column scrambling, and reconstruction on the low-frequency part of the cube ***LL*** to obtain the scrambling cube ***LL'***.

```
Algorithm 2
Input: row1, LL1
Output: LL1'
LL1'=LL;
s=size(LL1');
for i=1:s
t=LL1'(:,s-i+1);
LL'(:,s-i+1)=LL'(:,row(i));
LL'(:,row(i))=t;
end
```

Step 5 Perform Haar wavelet two-level reconstruction on each layer of the scrambled cube ***LL'*** and other high-frequency detail cubes ***LH***, ***HL***, ***HH*** to obtain a semi-encrypted cube ***C-Im***.

Step 6 Calculate ***c*** by ***×4*** as shown in formula (8).

$$c = \mathrm{mod}\left(\mathit{floor}\left(\mathit{reshape}(x4, a, a)*10^{1}4\right), 256\right) \tag{8}$$

Where, mod is the modulus function. *Floor* is the round-down function. And *reshape* is the reconstructed array size function. Convert ***c*** into a pseudo-random matrix with a range of [0,255] and a size of ***a*** × ***a***.

Step 7 Perform bitwise XOR operation on each layer of the semi-encrypted cube ***C-Im*** and ***c*** to obtain the encrypted image ***En-IM*** as shown in formula (9).

$$En{-}im(:,:,i) = bitxor(C{-}im(:,:,i), c), i{\in}[1, 2, \cdots, b]. \tag{9}$$

6 Experimental simulation and analysis

6.1 Experimental simulation

To verify the effectiveness of the proposed algorithm, six pictures were selected for simulation experiments, including four grayscale images and two colour images as shown in Table 1. The simulation experiment is carried out on the MATLAB R2020 platform. The encryption and decryption results are shown in Figs. 6 and 7. After encrypting six pictures with different sizes, colors and textures, six meaningless snowflake pictures are obtained. The decrypted pictures

Table 1 Test images

Number	Name	Colour	Size
Img.1	Lena	grayscale	256 × 256
Img.2	Barbara	grayscale	512 × 512
Img.3	market	grayscale	1024 × 1024
Img.4	west	grayscale	512 × 256
Img.5	flamingos	colour	256 × 256
Img.6	Baboon	colour	512 × 512

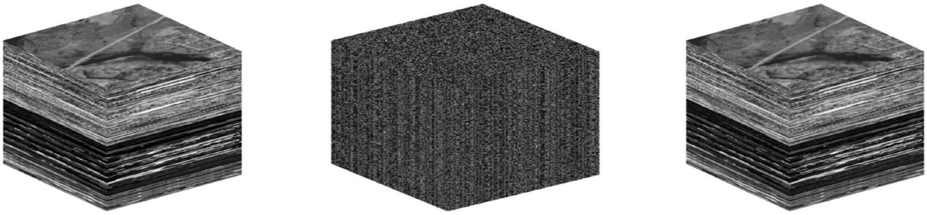
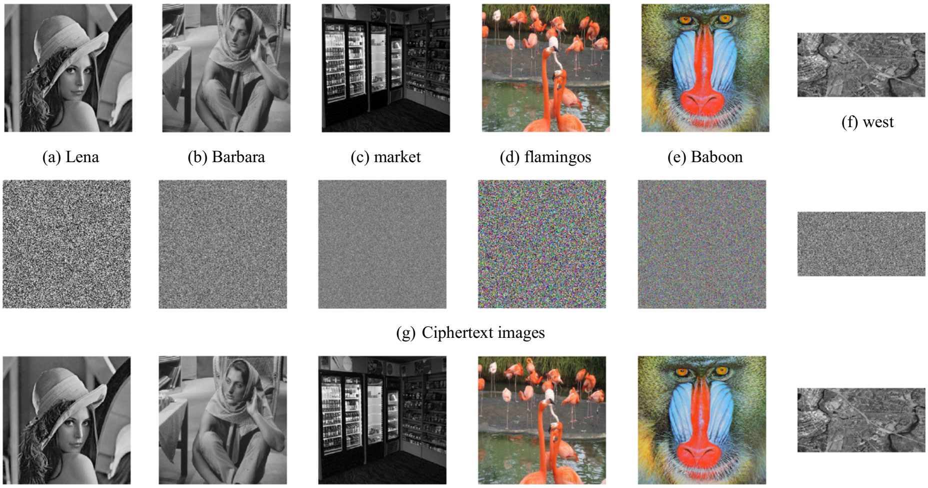

(a) Plaintext image cube (b) Ciphertext image cube (c) Decrypt image cube

Fig. 6 Encryption and decryption image cubes. **a** Plaintext image cube, **b** Ciphertext image cube, **c** Decrypt image cube

are indistinguishable from the original image in visual, indicating that the proposed algorithm is effective.

6.2 Decrypted image quality analysis

In this paper, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are used as indicators to measure the quality of decrypted images. PSNR is a measure of picture distortion.


(a) Lena (b) Barbara (c) market (d) flamingos (e) Baboon (f) west

(g) Ciphertext images

(h) Decrypt images

Fig. 7 Plaintext images, ciphertext images and decrypt images. **a** Lena, **b** Barbara, **c** market, **d** flamingos, **e** Baboon, **f** west, **g** Ciphertext images, **h** Decrypt images

When PSNR is greater than 30, the picture quality is considered acceptable. SSIM measures the similarity of two images. The SSIM closer to 1, the smaller the image distortion. The PSNR and SSIM values of the decrypted image in this paper are shown in Fig. 8. The PSNR values comparison is shown in Table 2. The PSNR values are greater than 31, and the SSIM values are greater than 0.96. Therefore, it can be considered that the decrypted image has less distortion and better quality.

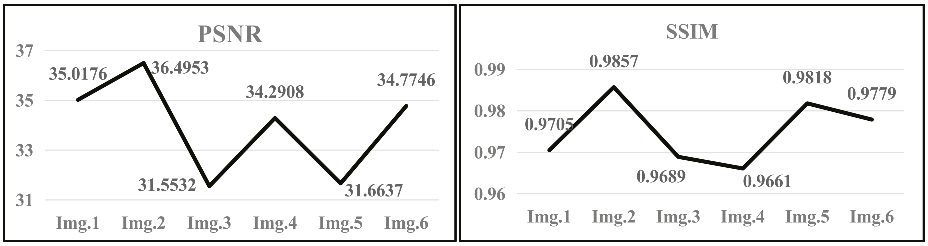
6.3 Statistical analysis

6.3.1 Histogram analysis

The 3D histogram distribution represents the grey level of each position in the image and it is a graphical indicator of statistical properties that reflects the spatial distribution and grey level value of the image. The plaintext histogram is shown in Fig. 9. The ciphertext histogram is shown in Fig. 10. It can be observed that the plaintext histogram has one or several peaks, showing the characteristics of the plaintext. In contrast, the ciphertext image is almost evenly distributed, which well conceals the statistical characteristics of the plaintext.

6.3.2 Correlation coefficient analysis of adjacent pixels

The adjacent pixels in the plaintext image have a high correlation, which is one of the important characteristics of the image. An effective encrypted image should generate ciphertext with a low correlation between adjacent pixels. Baboon's plaintext and ciphertext neighboring pixel relationship diagram is shown in Fig. 11. It can be observed that since the relationship between adjacent pixels of the plaintext is distributed on the diagonal, the similarity of adjacent pixels is very high while the ciphertext image is evenly distributed, and the characteristics of the plaintext cannot be obtained. Table 3 lists the adjacent pixel correlation coefficients of this method and other methods. The correlation coefficients of adjacent pixels of the ciphertext of the proposed algorithm are all close to 0, and the correlation of the ciphertext is very low.

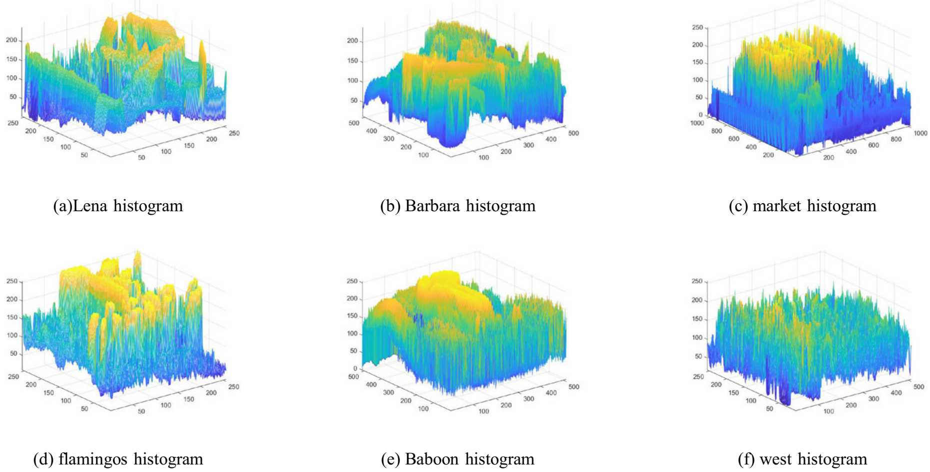


(a) Decrypt image PSNR value (b) Decrypt image SSIM value

Fig. 8 Decrypted image quality indicators. **a** Decrypted image PSNR value, **b** Decrypt image SSIM value

Table 2 PSNR comparison

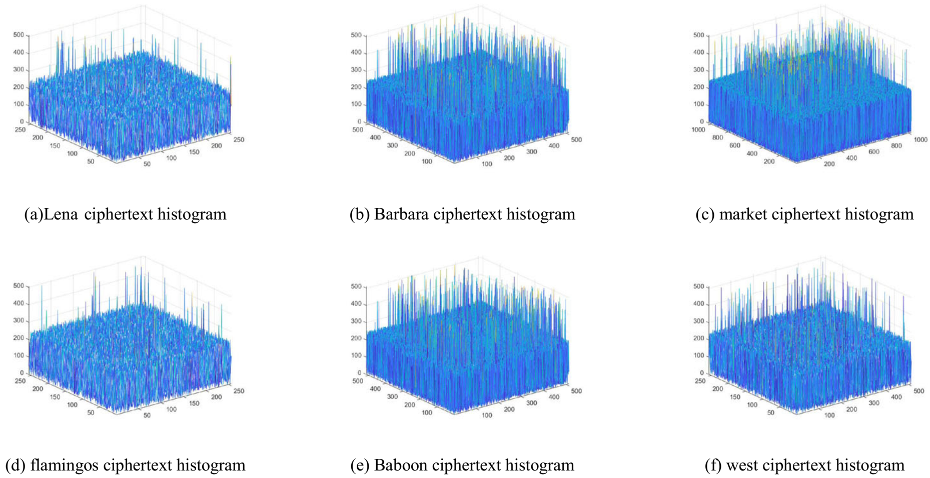
Method	Proposed	Ref [3]	Ref [12]
PSNR	33.9658	21.0667	30.342



(a)Lena histogram (b) Barbara histogram (c) market histogram

(d) flamingos histogram (e) Baboon histogram (f) west histogram

Fig. 9 Plaintext histogram. **a** Lena histogram, **b** Barbara histogram, **c** market histogram, **d** flamingos histogram, **e** Baboon histogram, **f** west histogram



(a)Lena ciphertext histogram (b) Barbara ciphertext histogram (c) market ciphertext histogram

(d) flamingos ciphertext histogram (e) Baboon ciphertext histogram (f) west ciphertext histogram

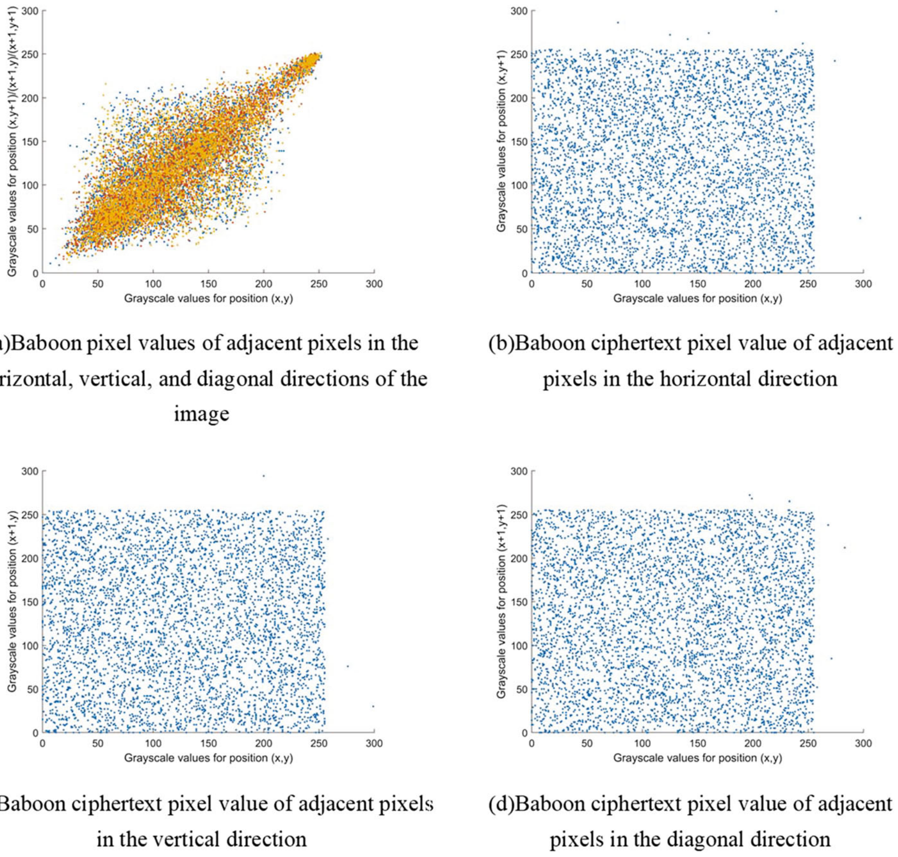
Fig. 10 Ciphertext histogram. **a** Lena ciphertext histogram, **b** Barbara ciphertext histogram, **c** market ciphertext histogram, **d** flamingos ciphertext histogram, **e** Baboon ciphertext histogram, **f** west ciphertext histogram

6.3.3 Complexity test

Time sequence complexity refers to the degree to which the sequence is close to a random sequence. The greater the complexity, the greater the randomness, and the more difficult it is for the sequence to be recovered. The complexity algorithm C_0 and spectral entropy (SE) can correctly reflect the complexity characteristics of the system [23]. The larger the C_0 and SE values the value, the greater the complexity. The C_0 and SE values of the ciphertext in this paper, logistic chaotic sequence and Lorenz chaotic sequence are shown in Table 4. It can be observed that the complexity of the

Table 3 Correlation coefficient comparison

Method		Correlation coefficient		
		Horizontal	Vertical	Diagonal
Proposed	Img.1	−0.0016	0.0052	0.0049
	Img.2	0.0023	−0.0001	−0.0095
	Img.3	−0.0043	−0.0019	0.0019
	Img.4	0.0000	−0.0045	−0.008
	Img.5	−0.0077	0.0036	−0.0012
	Img.6	0.0042	−0.0024	−0.0036
Ref [11]		0.0862	0.0683	0.0646
Ref [4]		0.0046	−0.0040	0.0008
Ref [16]		0.002	0.0047	0.0028
Ref [14]		−0.0074	0.0032	0.0121
Ref [3]		0.0779	0.028	0.0462



(a)Baboon pixel values of adjacent pixels in the horizontal, vertical, and diagonal directions of the image

(b)Baboon ciphertext pixel value of adjacent pixels in the horizontal direction

(c)Baboon ciphertext pixel value of adjacent pixels in the vertical direction

(d)Baboon ciphertext pixel value of adjacent pixels in the diagonal direction

Fig. 11 Baboon's plaintext and ciphertext neighboring pixel relationship. **a** Baboon pixel values of adjacent pixels in the horizontal, vertical, and diagonal directions of the image, **b** Baboon ciphertext pixel value of adjacent pixels in the horizontal direction, **c** Baboon ciphertext pixel value of adjacent pixels in the vertical direction, **d** Baboon ciphertext pixel value of adjacent pixels in the diagonal direction

Table 4 Complexity comparison

Complexity index	Ciphertext	Logistic	Lorenz		
			x	y	z
C_0	0.2362	0.3358	0.0702	0.0882	0.0623
SE	0.9456	0.9523	0.4999	0.4665	0.5258

ciphertext is slightly smaller than Logistic sequence, but much larger than the Lorenz system, which has strong complexity.

6.4 Anti-differential attack analysis

The differential attack is to use the same encryption algorithm to encrypt the plaintext image before and after the change by making small changes to the plaintext image, and to compare the encrypted results for deciphering the algorithm. The number of pixels changes rate (NPCR) is used to evaluate the dependence of the ciphertext image of the algorithm on the plaintext image. The ideal value of NPCR is 99.6045%. Table 5 lists the NPCR values of this method and other methods. It can be seen that this algorithm is closer to the ideal value than other algorithms.

6.5 Running speed

Comparing the running time with the proposed algorithm and the recent multi-image encryption algorithm, the experimental data volume, decryption time and encryption time are shown in Table 6. It can be observed that although the algorithm in this paper handles a larger amount of data, it still has a relatively fast speed. Therefore, the proposed algorithm has superiority in running speed.

Table 5 NPCR value comparison

Method	This paper	Ref [20]	Ref [25]	Ref [12]
NPCR	99.62%	99.58%	99.26%	82.04%

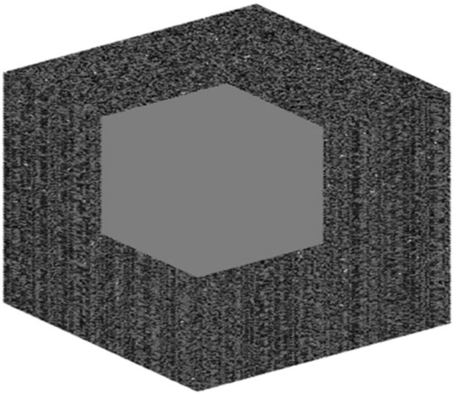
Table 6 Running speed comparison

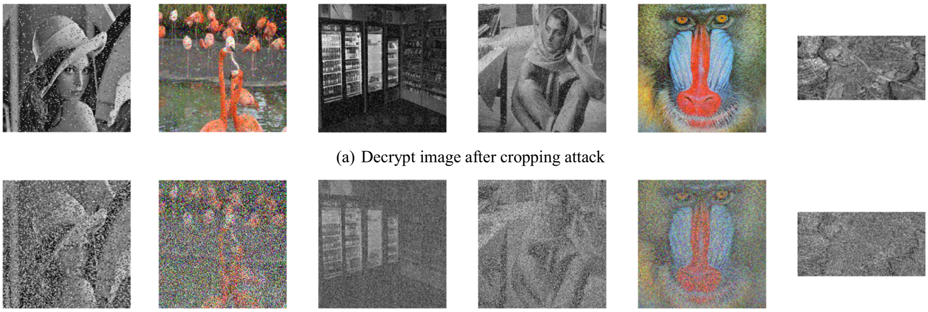
Method	Data volume	Encryption time /s	Decryption time /s
This paper	128*128*152	0.923	0.718
Ref [21]	128*128*128	1.78	–
Ref [25]	128*128*48	1.283	–
Ref [20]	128*128*68	3.2618	2.8451
Ref [8]	128*128*12	1.853	–

6.6 Noise attack and cutting attack

The 3D ciphertext cube image is subjected to a clipping attack and a 10% salt and pepper noise attack before decryption. The 12.5% ciphertext cube is cut as shown in Fig. 12. The decrypted image is shown in Fig. 13. The PSNR values and SSIM values of the decrypted image are shown in Table 7. It can be observed that after cutting 12.5% of the ciphertext cube, there is a certain amount of noise in the decrypted image, but the overall image content is relatively clear. After 10% of the salt and pepper noise attack, the decrypted image is noisy. However, the image content can still be distinguished, so the proposed algorithm has a certain degree of anti-attack.

Fig. 12 Cut cube





(a) Decrypt image after cropping attack

(b) Decrypt image after noise attack

Fig. 13 Decrypted image after the attack. **a** Decrypt image after cropping attack, **b** Decrypt image after noise attack

Table 7 Decrypted image PSNR value and SSIM value

Attack	Cut attack 12.5%		Salt and pepper noise 15%	
Evaluation index	PSNR	SSIM	PSNR	SSIM
Average value	17.5996	0.5379	11.6850	0.1840

7 Conclusion

This paper proposes a multi-image encryption algorithm based on wavelet transform and 3D shuffling scrambling. This paper designs the dynamic pseudo-random sequence generator based on the roulette algorithm and a dynamic chaotic library with three kinds of one-dimension chaotic maps that are highly efficient and have a good complexity, which is highly associated with the plaintext image. To better scramble the image cube, this paper proposes a 3D shuffling scrambling algorithm, which divides the cube into one-dimensional vectors and scrambles the order of the one-dimensional vectors, and then reorganizes the cube. To encrypt multiple images, first, the images to be encrypted are divided into blocks, cut, and then are reconstructed into a 3D cube image. Then, calculate the hash values of the image cube and combine hash value and dynamic chaos library to generate chaos sequences. Perform Haar wavelet transform on each layer of the image cube, and perform 3D shuffling of low-frequency coefficients in three different directions. Reconstruct the cube with the scrambled low-frequency coefficients and high-frequency three parts. Last, the chaotic matrix is XOR with each layer of the image cube to obtain the final encrypted image. The experimental results show that the proposed algorithm is better than other reference methods on the whole. The PSNR values of the decrypted images are all higher than 31 and the SSIM values are all higher than 0.96, so the decrypted image quality is higher. In the histogram analysis, correlation coefficient analysis, complexity analysis and anti-differential attack analysis, the information distribution of the ciphertext is more random, and the ciphertext has good cryptographic statistics. In the running speed test, the algorithm in this paper shows superiority. Under the attack of ciphertext and noise, the decrypted image can still have clear and identifiable content, so the proposed algorithm has a strong anti-attack. The algorithm has no restrictions on the shape of encrypted pictures, which is flexible. In summary, the proposed algorithm is safe, efficient and anti-attack, and has practical application value.

Acknowledgments This work was funded by Guangxi Natural Science Foundation (No.2018GXNSFAA138177), Guilin University of Electronic Technology Fund (No. C21YJM00QX99) the Innovation Project of GUET Graduate Education (No.2021YCXS119, No.2022YCX139) and supported by Data Analysis and Computing Laboratory of Guangxi University Key Laboratory.

Availability of data and material The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Code availability Not applicable.

Authors' contributions
Conceptualization: [Zhong Huiyan] and [Li Guodong];
Formal analysis: [Zhong Huiyan];
Investigation: [Zhong Huiyan];
Project administration: [Zhong Huiyan] and [Li Guodong];
Software: [Zhong Huiyan];
Validation: [Zhong Huiyan], [Li Guodong];
Writing – original draft: [Zhong Huiyan];
Writing – review \& editing: [Li Guodong], [Zhong Huiyan].

Declarations

Conflicts of interest All authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest or non-financial interest in the subject matter or materials discussed in this manuscript.

Ethics approval Informed consent.

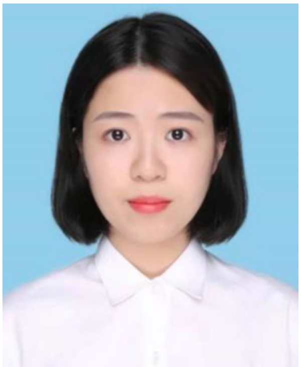
Consent to participate Informed consent

Consent for publication Informed consent

References

1. Abdulla AA, Sellahewa H, Jassim S (2019) Improving embedding efficiency for digital steganography by exploiting similarities between secret and cover images. Multimed Tools Appl 78:1–25. https://doi.org/10.1007/s11042-019-7166-7
2. Alawida M, Samsudin A, Teh JS, Alkhawaldeh RS (2019) A new hybrid digital chaotic system with applications in image encryption. Signal Process 160:45–58. https://doi.org/10.1016/J.SIGPRO.2019.02.016
3. An F, Liu J (2019) Image encryption algorithm based on adaptive wavelet Chaos. J Sensors 2019:Article ID 2768121. https://doi.org/10.1155/2019/2768121
4. Enayatifar R, Guimarães F, Siarry P (2019) Index-based permutation-diffusion in multiple-image encryption using DNA sequence. Opt Lasers Eng 115:131–140. https://doi.org/10.1016/J.OPTLASENG.2018.11.017
5. Farah MAB, Guesmi R, Kachouri A, Samet M (2020) A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation. Opt Laser Technol 121:105777. https://doi.org/10.1016/J.OPTLASTEC.2019.105777
6. Farah MA, Farah A, Farah T (2020) An image encryption scheme based on a new hybrid chaotic map and optimized substitution box. Nonlinear Dyn 99:3041–3064. https://doi.org/10.1007/s11071-019-05413-8
7. Guo Y, Zhou Y, Jing S (2020) Multiple-image encryption based on image recombination and bit scrambling. Acta Photonica Sin 49:410002–410002. https://doi.org/10.3788/gzxb20204904.0410002
8. Han S-m, Zhang W, Zhang X, Wei X-x, Wan X-j (2020) Multiple image encryption method based on light field imaging theory and chaotic system. Acta Photonica Sin 49:310002–310002. https://doi.org/10.3788/gzxb20204903.0310002
9. Hassaballah M, Hameed MA, Alkinani MH (2020) Introduction to digital image steganography. https://doi.org/10.1016/b978-0-12-819438-6.00009-8
10. Hua Z, Zhou Y, Huang H (2019) Cosine-transform-based chaotic system for image encryption. Inf Sci 480:403–419. https://doi.org/10.1016/j.ins.2018.12.048
11. Huang Z, Cheng S, Gong L, Zhou N (2020) Nonlinear optical multi-image encryption scheme with two-dimensional linear canonical transform. Opt Lasers Eng 124:105821. https://doi.org/10.1016/J.OPTLASENG.2019.105821
12. Huo D, Zhu Z, Wei L, Han C, Zhou X (2021) A visually secure image encryption scheme based on 2D compressive sensing and integer wavelet transform embedding. Opt Commun 492:126976. https://doi.org/10.1016/J.OPTCOM.2021.126976
13. Kadhim IJ, Premaratne P, Vial P, Halloran B (2019) Comprehensive survey of image steganography: techniques, evaluations, and trends in future research. Neurocomputing 335:299–326. https://doi.org/10.1016/j.neucom.2018.06.075
14. Kumar D, Joshi AB, Mishra VN (2020) Optical and digital double color-image encryption algorithm using 3D chaotic map and 2D-multiple parameter fractional discrete cosine transform. Results in Optics 1:100031. https://doi.org/10.1016/j.rio.2020.100031
15. Li G, Wang L (2018) Double chaotic image encryption algorithm based on optimal sequence solution and fractional transform. Vis Comput:1–11. https://doi.org/10.1007/s00371-018-1574-y
16. Li G-D et al (2014) Research on application of image encryption technology based on chaotic of cellular neural network. J Digit Inf Manag 12:151–158 Corpus ID: 13997561
17. Li C, Li H, Li F, Wei D, Yang X, Zhang J (2018) Multiple-image encryption by using robust chaotic map in wavelet transform domain. Optik 171:277–286. https://doi.org/10.1016/J.IJLEO.2018.06.029
18. Liu Y, Jiang Z, Xu X, Zhang F, Xu J (2020) Optical image encryption algorithm based on hyper-chaos and public-key cryptography. Opt Laser Technol 127:106171. https://doi.org/10.1016/j.optlastec.2020.106171

19. Lu Q, Zhu C, Deng X (2020) An efficient image encryption scheme based on the LSS chaotic map and single S-box. IEEE Access 8:25664–25678. https://doi.org/10.1109/ACCESS.2020.2970806
20. Patro KK, Soni A, Netam PK, Acharya B (2020) Multiple grayscale image encryption using cross-coupled chaotic maps. J Inf Secur Appl 52:102470. https://doi.org/10.1016/j.jisa.2020.102470
21. Sahasrabuddhe A, Laiphrakpam DS (2021) Multiple images encryption based on 3D scrambling and hyper-chaotic system. Inf Sci 550:252–267. https://doi.org/10.1016/j.ins.2020.10.031
22. Shakir HR (2019) An image encryption method based on selective AES coding of wavelet transform and chaotic pixel shuffling. Multimed Tools Appl 78:26073–26087. https://doi.org/10.1007/s11042-019-07766-z
23. Sun K-h (2013) Analysis of chaotic complexity characteristics based on C_0 algorithm. Acta Electron Sin 41(09):1765–1771 Corpus ID: 124977292
24. Xiao J, Xiao Y, Xie Y, Liu B, Ye Y, Song T, Junxiong C, Liu Y (2021) Exploiting optical chaos for double images encryption with compressive sensing and double random phase encoding. Opt Commun 484: 126683. https://doi.org/10.1016/j.optcom.2020.126683
25. Zhang L, Zhang X (2020) Multiple-image encryption algorithm based on bit planes and chaos. Multimed Tools Appl 79:20753–20771. https://doi.org/10.1007/s11042-020-08835-4
26. Zhao H, Xie S, Zhang J, Wu T (2020) Efficient image encryption using two-dimensional enhanced hyperchaotic Henon map. J Electron Imaging 29:023007–023007. https://doi.org/10.1117/1.JEI.29.2.023007
27. Zhu S, Wang G, Zhu C (2019) A secure and fast image encryption scheme based on double chaotic S-Boxes. Entropy 21:790. https://doi.org/10.3390/e21080790
28. Xu X-L, Li G-D, Dai W-Y, Song X-M (2021) Multi-direction chain and grid chaotic system based on Julia fractal. Fractals 29(08). https://doi.org/10.1142/S0218348X21502455
29. Dai W, Xu X, Song X, Li G (2022) Audio encryption algorithm based on Chen Memristor Chaotic System. Symmetry 14(1):17. https://doi.org/10.3390/sym14010017
30. Song X, Xu D, Li G, Xu W (2021) Multi-image reorganization encryption based on S-L-F cascade chaos and bit scrambling. Journal of Web Engineering. https://doi.org/10.13052/jwe1540-9589.20410



Huiyan Zhong received the B.E degree in applied statistics from Huizhou University, China. She is currently pursuing the M.S. degree in mathematics with Guilin University of Electronic Technology Guilin, Guangxi, China. Her main research interests include image encryption and nonlinear system.

Guodong Li received Ph.D degree in control theory and control engineering from University of Science and Technology Beijing, Beijing, China. He is currently a Professor with the School of Mathematics and Computational Science, Guilin University of Electronic Technology Guilin, Guangxi, China. His main research interests include information security, nonlinear system, machine learning, data mining and statistical prediction.