Multimedia Tools and Applications (2021) 80:13821–13840
https://doi.org/10.1007/s11042-020-10429-z

Image segmentation encryption algorithm with chaotic sequence generation participated by cipher and multi-feedback loops

Jie Deng[1] · Minjun Zhou[1] · Chunhua Wang[1] · Sicheng Wang[1] · Cong Xu[1]

Received: 5 November 2019 / Revised: 14 December 2020 / Accepted: 22 December 2020 /
Published online: 18 January 2021

Abstract
The existing chaotic image encryption algorithms have common defects: (i) ciphertext does not participate in the generation processes of chaotic pseudo-random sequences and key sequences; (ii) the entire encryption process does not have a closed-loop structure. In order to solve above problems, in this paper, an image segmentation encryption algorithm based on hyperchaotic system is proposed. We decompose the scrambled sequence into three sequences of different lengths: S_1, S_2 and S_3. Then, the initial values of the chaotic system are updated by the sequences S_2 and S_3 and using the updated initial value iterates the chaotic system and generates the key sequence K_3, and the sequence S_1 is encrypted by the sequence K_3 to obtain the cipher sequence C_1, using the sequences C_1 and S_3 updates the initial value of the chaotic system, and using the updated initial value iterates the chaotic system and generates the key sequence K_4, and using the sequence K_4 encrypts the sequence S_2 to obtain the cipher sequence C_2. Thus, ciphertext participates in the generation processes of chaotic pseudo-random sequences and key sequences, and the entire encryption process has a closed-loop structure. The experimental results show that the encryption algorithm has high security and sensitivity.

Keywords Chaotic image encryption · Segmentation encryption · Hyperchaotic system · Ciphertext feedback · Closed-loop structure

1 Introduction

Chaotic systems have some significant features, such as deterministic, pseudo-randomness, ergodicity, and they are sensitive to initial points and parameters. Many new chaotic systems and circuit were proposed in recent years [5, 10, 11, 19, 29]. At present, chaotic

✉ Chunhua Wang
wch1227164@hnu.edu.cn

[1] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, People's Republic of China

image encryption technology is becoming more and more mature, and many image encryption algorithms based on chaotic systems were proposed [2, 4, 6, 8, 12–14, 16, 18, 20–23, 25–27]. Alvarez et al. reviewed the chaotic image encryption algorithm and proposed a common framework for the basic criteria of chaotic image encryption [1]. Zhen et al. proposed a secure image encryption algorithm based on logic and spatiotemporal chaos [26]. In the encryption algorithm, 8 DNA coding rules are used to improve the efficiency of image scrambling and diffusion. Ghebleh et al. proposed an image encryption algorithm for irregularly extracted chaotic maps [6]. The algorithm scrambles the image in a scrambling process in conjunction with a three-dimensional Cat map and a zigzag scan. The diffusion process spreads the image with a combined output of three oblique tent maps. Ye made full use of the excellent characteristics of chaotic systems, and proposed a pixel-bit image scrambling encryption algorithm based on chaotic system [23]. However, Li et al. founded that it could not effectively resist known plaintext or choose plaintext attacks [9]. According to the characteristics of RGB images, Liu et al. combined DNA and chaotic systems to propose an RGB image encryption algorithm based on DNA coding and chaotic mapping [13]. However, it was cracked by Liu et al. using a known plaintext image [15]. Patidar et al. proposed a new lossless symmetric image encryption algorithm based on scrambling-diffusion structure using standard mapping and logical mapping [16]. However, it was cracked by Rhouma et al. using only a pair of plain text or ciphertext [17]. Liu et al. combined a Choquet fuzzy integral (CFI) and hyperchaotic system to design a new color image encryption algorithm [14]. The encryption core of the algorithm is a pseudo-random number generator based on CFI. Kwok et al. proposed a fast image encryption algorithm based on chaotic system and stream cipher structure [8]. Based on the permutation-diffusion architecture, Yin et al. introduced a more sensitive chaotic image encryption scheme using breadth-first search and dynamic diffusion [25]. Recently, Xu et al. proposed an image encryption algorithm based on random walk and hyperchaotic systems [21]. In the algorithm, the way to scramble is novel and efficient. However, the chaotic image encryption algorithms mentioned above all have common defects: (1) In the entire encryption process, cipher does not participate in the generation processes of chaotic pseudo-random sequences and key sequences; (2) the entire encryption process does not have a closed-loop structure, which means there is no valid ciphertext feedback. Due to these two drawbacks, these algorithms have poor ability to resist differential attacks, and are susceptible to chosen-plaintext attack.

In order to overcome the above shortcomings, this paper proposes an image segmentation encryption algorithm with chaotic sequence generation participated by cipher and multi-feedback loops based on hyperchaotic system. In this algorithm, we first decompose the scrambled sequence into three sequences of different lengths: S_1, S_2 and S_3. Then, the initial values of the chaotic system are updated by the sequences S_2 and S_3 and using the updated initial value iterates the chaotic system and generates the key sequence K_3, and the sequence S_1 is encrypted by the sequence K_3 to obtain the cipher sequence C_1; using the sequences C_1 and S_3 updates the initial value of the chaotic system, and using the updated initial value iterates the chaotic system and generates the key sequence K_4, and using the sequence K_4 encrypts the sequence S_2 to obtain the cipher sequence C_2. Similarly, we use the sequences C_1 and C_2 to update the initial value of the chaotic system, and using the updated initial value iterates the chaotic system and generates the key sequence K_5. The sequence K_5 is used to encrypt the sequence S_3 to obtain the cipher sequence C_3. Finally, we splice the cipher sequences C_1, C_2,

and C_3 to obtain the ciphertext image C. As can be seen from the above, in our encryption process, plaintext and cipher participate in the chaotic pseudo-random sequence generation process, and the algorithm makes the key sequence used to generate the encrypted plaintext not only related to the plaintext but also to the cipher, and obviously, there is cipher feedback, so the encryption process has a closed-loop structure. Furthermore, the proposed encryption algorithm makes full use of the pseudo-randomness of the hyperchaotic system and the sensitivity to the initial conditions, so that the security of the encryption algorithm is greatly improved. Various experimental results show that the proposed encryption algorithm has high security and sensitivity.

2 Preliminaries

2.1 Hyperchaotic system

The equation of state corresponding to the four-dimensional hyperchaotic system selected by this scheme is:

$$\begin{cases} \dot{x_1} = a(x_2 - x_1) + x_2 x_3 \\ \dot{x_2} = b x_1 - x_2 - x_1 x_3 + x_4 \\ \dot{z} = x_1 x_2 - c x_3 \\ \dot{x_4} = d x_4 - x_1 x_3 \end{cases} \tag{1}$$

In the formula, when each key parameter is selected as a = 3, b = 60, c = 20, d = 1.3, the system is in a hyperchaotic state.

3 Proposed encryption system

3.1 Encryption process

The flowchart of the encryption algorithm proposed in this paper is shown in Fig. 1.

From Fig. 1, it can be seen that the scrambled sequence is split into three sequences of different lengths S_1, S_2 and S_3. Then, the initial values of the chaotic system are updated by the sequences S_2 and S_3 and using the updated initial value iterates the chaotic system and generates the key sequence K_3, and the sequence S_1 is encrypted by the sequence K_3 to obtain the cipher sequence $C_{1;}$ using the sequences C_1 and S_3 updates the initial value of the chaotic system, and using the updated initial value iterates the chaotic system and generates the key sequence K_4, and using the sequence K_4 encrypts the sequence S_2 to obtain the cipher sequence C_2. We use the sequences C_1 and C_2 to update the initial value of the chaotic system, and using the updated initial value iterates the chaotic system and generates the key sequence K_5. The sequence K_5 is used to encrypt the sequence S_3 to obtain the cipher sequence C_3. Finally, we splice the cipher sequences C_1, C_2, and C_3 to obtain the ciphertext image C. As can be seen from the Fig. 1, in our encryption process, plaintext and cipher participate in the generation

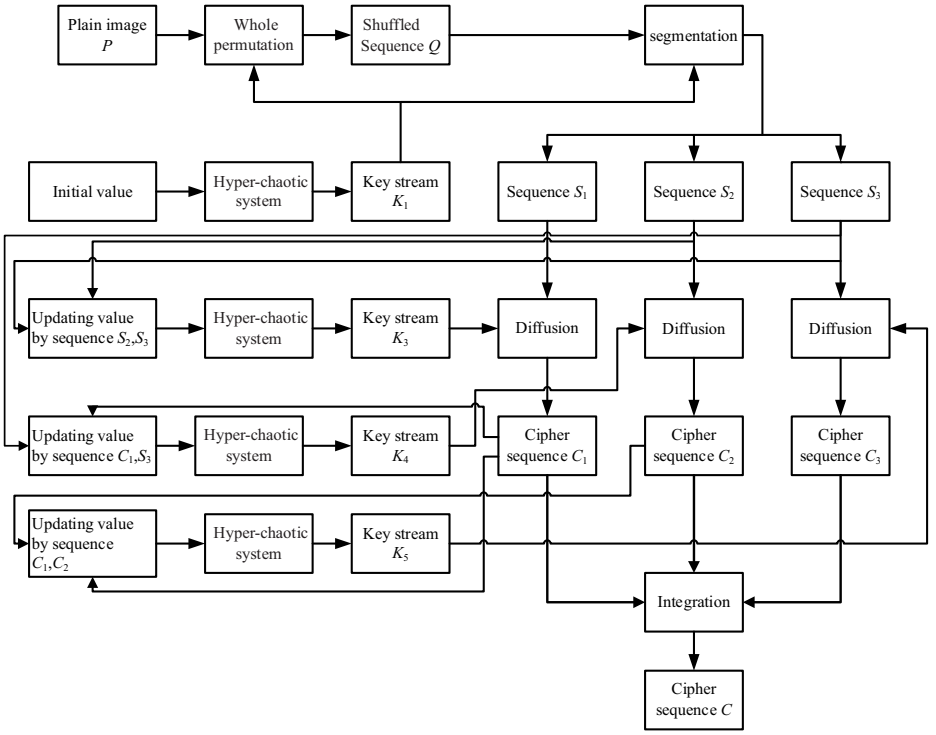

Fig. 1 Block diagram of the encryption algorithm

processes of chaotic pseudo-random sequences and key sequences, and the key sequence used to generate the encrypted plaintext not only related to the plaintext but also to the cipher, and obviously, there is cipher feedback, so the encryption process has a closed-loop structure.

Assume that the plaintext image is represented as P and its size is $M \times N$, we can form the following encryption steps:

Step 1: Use the initial conditions $x_1 = 1$, $x_2 = 0.949$, $x_3 = 1$, $x_4 = 1$, to iterate the chaotic system (1) $N_0 + L$ times, where $N_0 = 500$, $L = M \times N$, thus obtain four pseudo-random chaotic sequences: x_{11}, x_{21}, x_{31}, x_{41}. Furthermore, to get rid of transient effect, we discard the first N_0 sequence values of each sequence. Then, we get two sequences $K_1 = \{k_1(i)\}_{i=1}^{L}$ and $K_2 = \{k_2(i)\}_{i=1}^{L}$ by

$$k_1(i) = x_{11}(i) + x_{21}(i), \quad i = 1, 2, 3, \ldots, L \tag{2}$$

$$k_2(i) = mod\left(\left\lfloor (x_{31}(i) + x_{41}(i)) \times 10^{15} \right\rfloor, 3\right), \quad i = 1, 2, 3, \ldots, L \tag{3}$$

where mod(a, b) returns the remainder of a divided by b.

Step 2: Convert the plaintext image P into a one-dimensional vector $P = \{p(i)\}_{i=1}^{L}$ in a row-scanning priority manner.

Step 3: Sort the sequence K_1 from small to large to obtain K_1', and generate a new sequence $T = \{t(i)\}_{i=1}^{L}$ for recording the position of each element of K_1' in the original sequence K_1. Then, scramble the sequence P by

$$q(i) = b(t(i)), \quad i = 1, 2, 3, \ldots, L \tag{4}$$

to obtain the scrambling sequence $Q = \{q(i)\}_{i=1}^{L}$.

Step 4: By using the position of the element equal to 0, 1, 2 in k_2, respectively, scramble sequence Q is divided into three sequences S_1, S_2, S_3.

Step 5: Let $v = \text{length}(S_1)$, $o = \text{length}(S_2)$, $l = \text{length}(S_3)$.

Step 6: Update the initial value of the chaotic system by

$$x_i = x_i + \frac{\sum_{j=1}^{o} S_2(j) + \sum_{k=1}^{l} S_3(k)}{255 \times (o + l)}, \quad i = 1, 2, 3, 4 \tag{5}$$

and repeat (1) with the updated initial value $N_0 + v$ times and discard the first N_0 elements to obtain four chaotic sequences. Finally, the sequence K_3 used to encrypt the S_1 sequence is obtained by

$$k_3(i) = mod\left(\left\lfloor (x_{21}(i) + x_{22}(i) + x_{23}(i) + x_{24}(i)) \times 10^{15} \right\rfloor, 256\right), i = 1, 2, 3, \ldots, v \tag{6}$$

Step 7: Let $sum_1 = \sum_{i=2}^{v} S_1(i)$, and encrypt the sequence S_1 to obtain the ciphertext sequence C_1 by

$$C_1(1) = S_1(v) \oplus S_1(1) \oplus k_3(1) \tag{7}$$

$$\begin{cases} sum_1 = sum_1 - S_1(i) \\ j = mod(sum_1, i-1) + 1 \\ C_1(i) = C_1(i-1) \oplus C_1(j) \oplus S_1(i) \oplus k_3(i) \\ i = 2, 3, \ldots, v \end{cases} \tag{8}$$

where $\oplus$ is the bit-level XOR operator.

Step 8: Update the initial value of the chaotic system by

$$x_i = x_i + \frac{\sum_{j=1}^{v} C_1(j) + \sum_{k=1}^{l} S_3(k)}{255 \times (v + l)}, \quad i = 1, 2, 3, 4 \tag{9}$$

and repeat (1) $N_0 + o$ times with the updated initial value and discard the first N_0 elements to obtain four chaotic sequences. Finally, the sequence K_4 used to encrypt the S_2 sequence is obtained by

$$k_4(i) = mod\left(\left\lfloor (x_{31}(i) + x_{32}(i) + x_{33}(i) + x_{34}(i)) \times 10^{15} \right\rfloor, 256\right), i = 1, 2, 3, \ldots, o. \quad (10)$$

Step 9: Let $sum_2 = \sum_{i=2}^{o} S_2(i)$, and encrypt the sequence S_2 by

$$C_2(1) = S_2(o) \oplus S_2(1) \oplus k_4(1) \quad (11)$$

$$\begin{cases} sum_2 = sum_2 - S_1(i) \\ j = mod(sum_2, i-1) + 1 \\ C_2(i) = C_2(i-1) \oplus C_2(j) \oplus S_2(i) \oplus k_4(i) \\ i = 2, 3, \ldots, o \end{cases} \quad (12)$$

to obtain the ciphertext sequence C_2.

Step 10: Update the initial value of the chaotic system by

$$x_i = x_i + \frac{\sum_{j=1}^{v} C_1(j) + \sum_{k=1}^{o} C_2(k)}{255 \times (v + o)} \quad i = 1, 2, 3, 4 \quad (13)$$

and re-iterate the chaotic system (1) $N_0 + l$ times with the updated initial value and discard the first N_0 elements to obtain four chaotic sequences. Finally, the K_5 sequence used to encrypt the S_3 sequence is obtained by

$$k_5(i) = mod\left(\left\lfloor (x_{41}(i) + x_{42}(i) + x_{43}(i) + x_{44}(i)) \times 10^{15} \right\rfloor, 256\right) \; i = 1, 2, 3, \ldots, o \; . \quad (14)$$

Step 11: Let $sum_3 = \sum_{i=2}^{l} S_3(i)$, and encrypt the sequence S_3 to obtain the ciphertext sequence C_3 by

$$C_3(1) = S_3(l) \oplus S_3(1) \oplus k_5(1) \quad (15)$$

$$\begin{cases} sum_3 = sum_3 - S_3(i) \\ j = mod(sum_3, i-1) + 1 \\ C_3(i) = C_3(i-1) \oplus C_3(j) \oplus S_3(i) \oplus k_5(i) \\ i = 2, 3, \ldots, l \end{cases} \quad (16)$$

Step 12: The ciphertext sequence C is obtained by splicing the ciphertext sequences C_1, C_2 and C_3 into an image of size $M \times N$.

3.2 Decryption process

The decryption algorithm is the reverse process of encryption algorithm. The general flow of the decryption algorithm is to first divide the ciphertext sequence into three sequences, then perform the inverse diffusion process on the three sequences separately, and finally integrate the three sequences obtained in the inverse diffusion process into one sequence and inverse the sequence. The original clear text image can be obtained by scrambling the process. The specific decryption process is as follows:

Step 1: Use the correct decryption key, which is the same as the initial value of the chaotic system used in the encryption process. Then iterative chaotic system (1) and get two sequences: $K_1 = \{k_1(i)\}_{i=1}^{L}$ and $K_2 = \{k_2(i)\}_{i=1}^{L}$.
Step 2: Let v = length(find(K==0)), o = length(find(K==1)), l = length(find(K==2), and divide the sequence C into $C_1 = \{c_1(i)\}_{i=1}^{v}$, $C_2 = \{c_2(i)\}_{i=1}^{o}$ and $C_3 = \{c_3(i)\}_{i=1}^{l}$ by

$$\begin{cases} C_1 = C(1:v) \\ C_2 = C(v+1:v+o) \\ C_3 = \mathrm{C}(v+o+1:L) \end{cases}. \tag{17}$$

Step 3: Reverse diffusion decryption and get sequence C_3. Update the initial values according to the formula (13) with the sequences C_1 and C_2; then, iterate the chaotic system (1) and get the sequence $K_5 = \{k_5(i)\}_{i=1}^{l}$; and finally, let $sum_3 = 0$, and the sequence C_3 is subjected to a reverse diffusion decryption process to obtain a sequence S_3 by.

$$\begin{cases} j = mod(sum_3, i\text{–}1) + 1 \\ s_3(i) = c_3(i\text{–}1) \oplus c_3(j) \oplus c_3(i) \oplus k_5(i) \\ sum_3 = sum_3 + c_3(i) \\ i = l, l\text{–}1, l\text{–}2, \ldots 2 \end{cases} \tag{18}$$

$$s_3(1) = s_3(l) \oplus c_3(1) \oplus k_5(1) \tag{19}$$

Step 4: Reverse diffusion decryption and get sequence C_2. Update the initial values according to eq. (1.9) with sequences C_1 and S_3; then, iterate the chaotic system (1) and get the sequence $K_4 = \{k_4(i)\}_{i=1}^{o}$; finally, let $sum_2 = 0$, and perform a reverse diffusion decryption process on sequence C_2 to obtain sequence S_2, by

$$\begin{cases} j = mod(sum_2, i\text{–}1) + 1 \\ s_2(i) = c_2(i\text{–}1) \oplus c_2(j) \oplus c_2(i) \oplus k_4(i) \\ sum_2 = sum_2 + c_2(i) \\ i = \mathrm{o}, o\text{–}1, o\text{–}2, \ldots 2 \end{cases}. \tag{20}$$

$$s_2(1) = s_2(o) \oplus c_2(1) \oplus k_4(1). \tag{21}$$

Step 5: Reverse diffusion decryption and get sequence C_1. Update the initial values according to eq. (5) with sequences S_2 and S_3; then, iterate the chaotic system (1) and get the sequence $K_3 = \{k_3(i)\}_{i=1}^{v}$; finally, let $sum_1 = 0$, and according to

$$\begin{cases} j = mod(sum_1, i-1) + 1 \\ s_1(i) = c_1(i-1) \oplus c_1(j) \oplus c_1(i) \oplus k_3(i) \\ sum_1 = sum_1 + c_1(i) \\ i = v, v-1, v-2, \ldots 2 \end{cases} \tag{22}$$

$$s_1(1) = s_1(v) \oplus c_1(1) \oplus k_3(1), \tag{23}$$

the sequence S_1 is obtained by performing a reverse diffusion decryption process on the sequence C_1.

Step 6: Integrate the sequences S_1, S_2 and S_3 into a sequence $Q = \{q(i)\}_{i=1}^{L}$, and the integration rules are as follows: Insert sequence S_1 corresponding to the position where element is equal to 0 in k_2. Insert sequence S_2 corresponding to the position where element is equal to 1 in k_2. Insert sequence S_3 corresponding to the position where element is equal to 2 in k_2.

Step 7: Reverse scrambling sequence Q. The sequence K_1 is first sorted from small to large, and a sequence $T = \{t(i)\}_{i=1}^{L}$ for recording the position of each element of the new sequence in the original sequence is generated. Then, by reversing the chaotic sequence Q by.

$$p(t(i)) = q(i), \tag{24}$$

the original plaintext sequence $= \{p(i)\}_{i=1}^{L}$ is obtained and converted into an $M \times N$ image.

4 Encryption performance analysis

In this section, we analyze the security and performance of the proposed algorithm from key space analysis, gray histogram analysis, correlation analysis, Resisting differential attack analysis, key sensitivity analysis, information entropy analysis and time complexity analysis.

4.1 Experimental result

The 256×256 Lena grayscale image is used as the original image, and it is encrypted and decrypted experimentally by the algorithm proposed in this paper. The simulation results are shown in Fig. 2. Figure 2(a) shows the Lena plaintext image, and Fig. 2(b) shows the encrypted ciphertext image. At this time, there is no plaintext information in the image, and the encrypted content cannot be distinguished. Figure 2(c) shows the decrypted image, and it can be seen that there is no difference from the plaintext image.

4.2 Key space analysis

The key space is the sum of all the key sizes used in the encryption process. In the presented algorithm, the key space contains the initial conditions of the chaotic system. Where x_1, x_2, x_3, and x_4 are double-precision decimals, so the key space of the algorithm is $(10^{15})^4 = 10^{60} \approx 2^{199}$. A secure encryption algorithm should have a large enough key space to resist the purpose of exhaustive attacks. The key space should generally be no less than 2^{100}. Obviously, the proposed image encryption system has a large enough key space to resist brute-force attacks.

4.3 Histogram analysis

The gray histogram can clearly show the relationship between the gray value of the gray image and the gray value frequency. Therefore, important statistical features of some grayscale images can be obtained using grayscale histograms. The X axis of the gray histogram represents the gray value of the gray image, and the Y axis represents the number of each gray value. By analyzing the gray histogram, the related information of the gray image can be obtained. A highly secure image encryption system should make the histogram of the encrypted ciphertext image as flat as possible. The gray histogram of the Barbara image and the gray histogram of the corresponding ciphertext image are shown in Fig. 3. It can be clearly seen that the gray histogram of the plaintext image is jagged and the gray histogram of the ciphertext image is close to horizontal. Therefore, the encryption algorithm can effectively resist statistical attacks.

4.4 Information entropy

Information entropy can be used to evaluate the distribution of gray value of images [7]. The larger the information entropy, the more uniform the image pixel distribution, that is, the higher the uncertainty of image information. Information entropy is estimated as

$$H(m) = \sum_{i=0}^{2^N-1} P(m_i) log \frac{1}{p(m_i)} \tag{25}$$

where N is the number of bits corresponding to the symbol m_i, and $p(m_i)$ is the probability of symbol m. For a 256×256 grayscale image, its ideal information entropy is 8. Table 1 gives the information entropy values of some 256×256 grayscale images before and after encryption using the algorithm and the information entropy values of the ciphertext images after

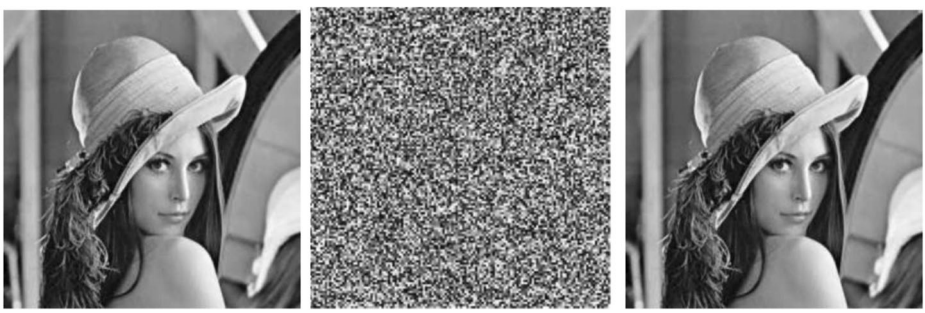

Fig. 2 Results: **a** original image of Lena, **b** encrypted image of Lena, **c** decrypted image of Lena

encrypting these images using the algorithm in Ref. [24], Ref. [3] and Ref. [23]. It shows that the information entropy value of the plaintext image differs greatly from the ideal information entropy value, and the information entropy values of each ciphertext image are close to the ideal entropy value. In addition, the information entropy value of the ciphertext image after encrypting the original image using the algorithm of this chapter is larger than the ones by other algorithms. Therefore, in the improvement of image information, the algorithm of this paper has more advantages than other algorithms.

4.5 Correlation analysis

For any plaintext image, it has a strong correlation between each pixel in the horizontal, vertical, and diagonal directions and its neighboring pixels. Therefore, a highly secure image encryption algorithm should completely break the correlation between adjacent pixels of the image [22]. The corresponding correlation coefficient can be calculated by

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{26}$$

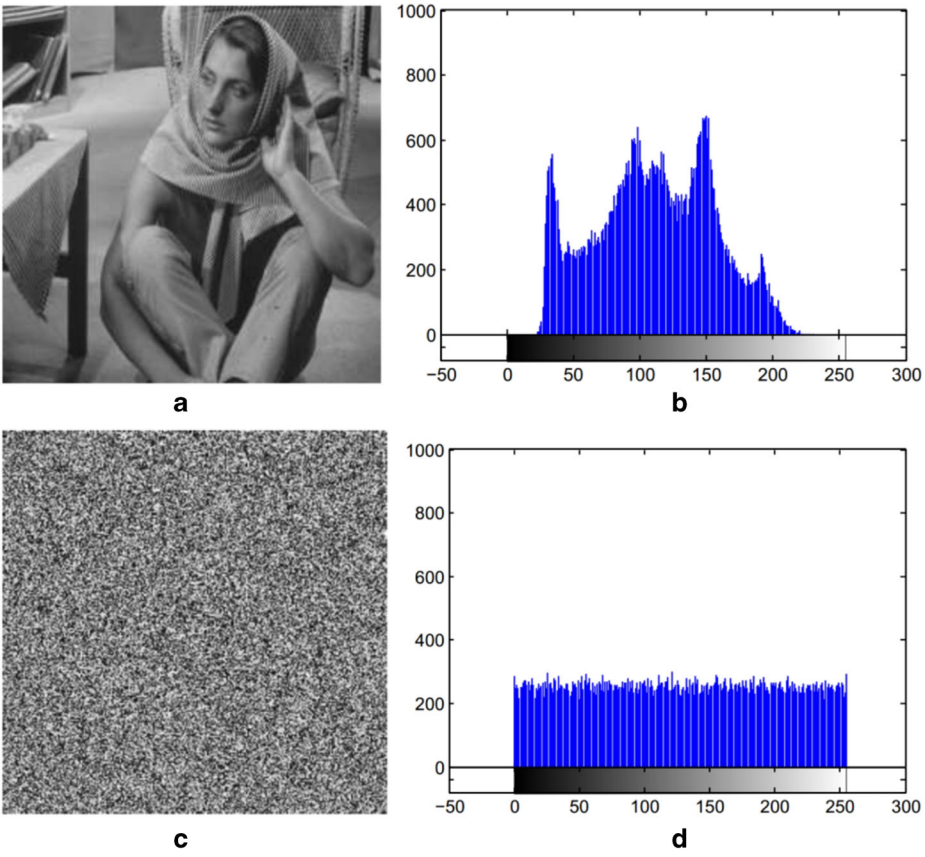

Fig. 3 Histogram of Barbara: **a** original image of Barbara (**b**) histogram of Barbara (**c**) encrypted image of Barbara (**d**) histogram of Barbara

Table 1 Information entropy

Image	Algorithms	Lena	Peppers	Brain
Information entropies	This paper	7.2072	7.5256	5.0421
	Ref. [24]	7.9981	7.9978	7.9977
	Ref. [3]	7.8974	7.8692	7.9868
	Ref. [23]	7.9973	7.9975	7.9975
		7.9974	7.9897	7.9969

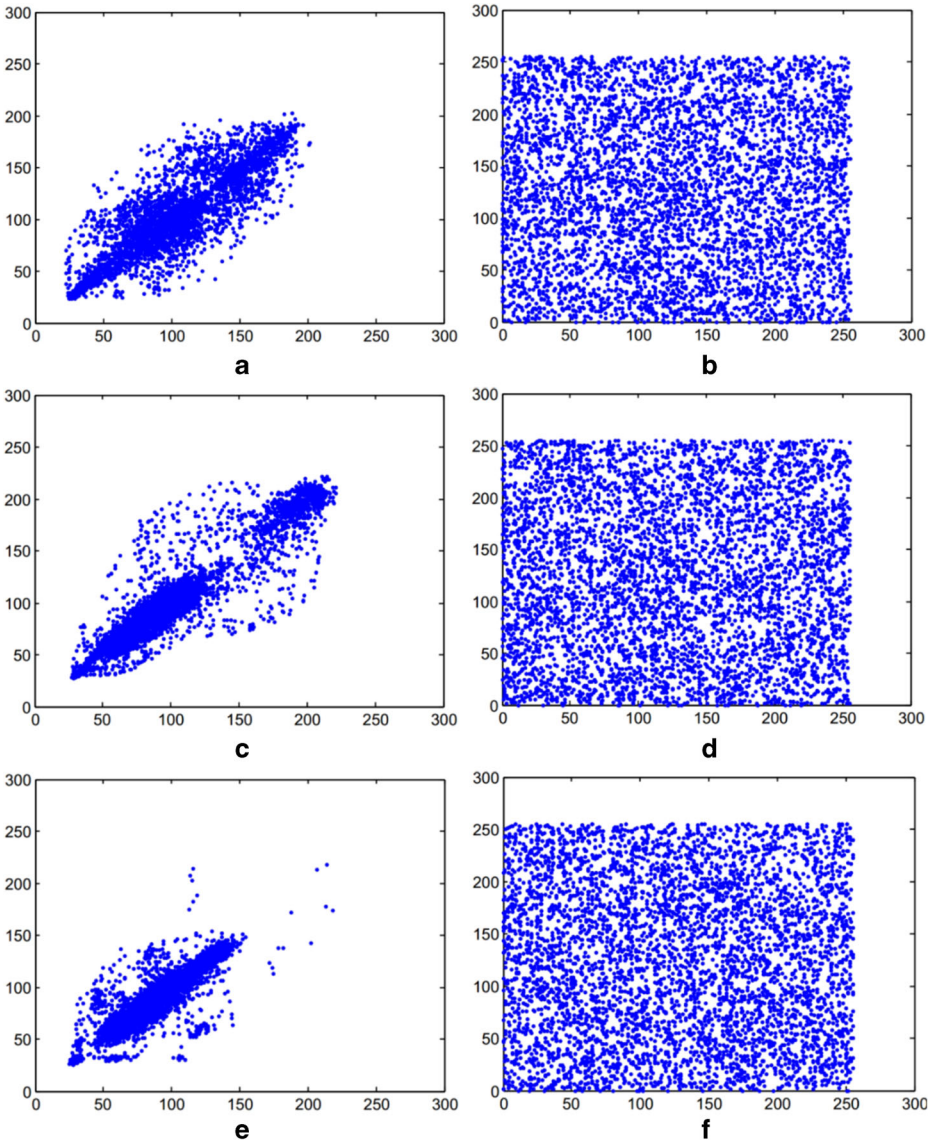

Fig. 4 The correlation plots of Barbara image and corresponding ciphered image of Barbara: (**a**) horizontal correlation of Barbara, (**b**) horizontal correlation of ciphered image, (**c**) vertical correlation of Barbara, (**d**) vertical correlation of ciphered image, (**e**) diagonal correlation of Barbara and (**f**) diagonal correlation of ciphered image

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2 \tag{27}$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{28}$$

$$\gamma_{xy} = \frac{cov(x,y)}{\sqrt{D(x)\cdot D(y)}} \tag{29}$$

In order to prove that the proposed algorithm can completely break the correlation of adjacent pixels in the plaintext image, we select the Barbara image as the experimental object, and then randomly from the Barbara image and its encrypted ciphertext image in the horizontal, vertical and diagonal directions select 5000 pairs of adjacent pixels to draw the corresponding correlation distribution as shown in Fig. 4. It can be seen the correlation of adjacent pixels in the three directions of the plaintext image is strong, while the others in the three directions of the ciphertext image is very low. To further prove that the algorithm can destroy the correlation of adjacent pixels of the image, we randomly select 5000 pairs of adjacent pixel pairs in three directions of Lena, Peppers and Barbara images, and calculate the corresponding correlation coefficients as shown in Table 2. The correlation coefficient of adjacent pixels of the plaintext image in three directions is close to 1, and the correlation coefficient of adjacent pixels of the corresponding ciphertext image in three directions is close to 0. Besides, the correlation coefficients of cipher image Lena compared with other schemes [3, 27, 28] are given in Table 3, we can see that the correlation coefficients among adjacent pixels in the cipher image in our proposed scheme are lower than that in the plain image. Hence, the proposed scheme has the ability of resisting the statistical attack.

4.6 Resisting differential attack analysis

Differential attack is a very common attack method in cryptography. To resist it, a secure encryption system should ensure that any minor changes in the plain image would cause significant effects on the difference between the cipher images.

NPCR and UACI are two indicators used to measure the difference between two images [4]. Their ideal values are calculated as follows:

Table 2 Correlation coefficients of the original and encrypted images

Direction	Horizontal	Vertical	Diagonal
Plain image of Lena	0.9563	0.9957	0.8838
Cipher image of Lena	−0.0162	−0.0038	−0.0023
Plain image of Peppers	0.9445	0.9622	0.9231
Cipher image of Peppers	−0.0064	−0.0058	−0.0027
Plain image of Barbara	0.8882	0.9324	0.9104
Cipher image of Barbara	−0.0029	0.0018	0.0027

$$\mathrm{NPCR}_{\mathrm{expected}} = \left(1 - \frac{1}{2^{\log_2 \mathrm{L}}}\right) \times 100\% \tag{30}$$

$$\mathrm{UACI}_{\mathrm{expected}} = \frac{1}{L^2}\left(\frac{\sum_{i=1}^{L-1} i(i+1)}{L-1}\right) \times 100\% \tag{31}$$

Where L is the gray level of image. Thus, for an image whose L is 256, the ideal NPCR is 0.996049, and the ideal UACI is 0.334635.

Lena is used for testing. We calculate values of NPCR and UACI to test the effects of a 1-bit change in the original image on the corresponding cipher images. Repeat 1000 times and get the graphs in Fig. 5. It can be clearly seen from the Fig. 5 that the NPCR and UACI values fluctuate above and below their respective ideal values, indicating that the proposed algorithm can effectively resist differential attacks. In addition, the average UACI and NPCR values of the proposed encryption scheme have been obtained through simulation. As shown in Table 4, The experiment results indicate that the value of NPCR and UACI in our proposed scheme are 0.996090 and 0.334907 respectively, and they are both closer to the theoretical value than most of the other schemes [4, 20–22]. Therefore, our proposed scheme can effectively resist differential attack.

4.7 Chosen-plaintext attack analysis

In the proposed algorithm, the initial conditions of hyper-chaotic system are related to plain image, so different key streams are obtained when encrypting different images. Therefore, the key streams obtained from one chosen-plain image cannot be used to decrypt other cipher images.

In a chosen-plaintext attack, the attacker try to obtain some information from the histogram of cipher images, such as cipher images of all white or black image (Fig. 6(c) and (f)). We can see from the Fig. 6(c) and (f) that the grayscale values of cipher images are distributed uniformly, so the attacker cannot get useful information. Then the attacker chose a black image (all zero image) as a plain image as shown in Fig. 6(d), and obtained the cipher image (Fig. 6(e)), he extracted the key stream as follows:

$$\mathrm{KS}_0(i,j) = C_0(i,j) \oplus P_0(i,j) \tag{32}$$

Table 3 The comparison of correlation coefficient

	Encryption schemes	Horizontal	Vertical	Diagonal
Cipher image of Lena	ours	−0.0162	−0.0038	−0.0023
Cipher image of Lena	Ref. [3]	0.0044	0.0067	0,0081
Cipher image of Lena	Ref. [28]	0.0102	−0.0053	−0.0161
Cipher image of Lena	Ref. [27]	−0.038118	−0.029142	−0.002736

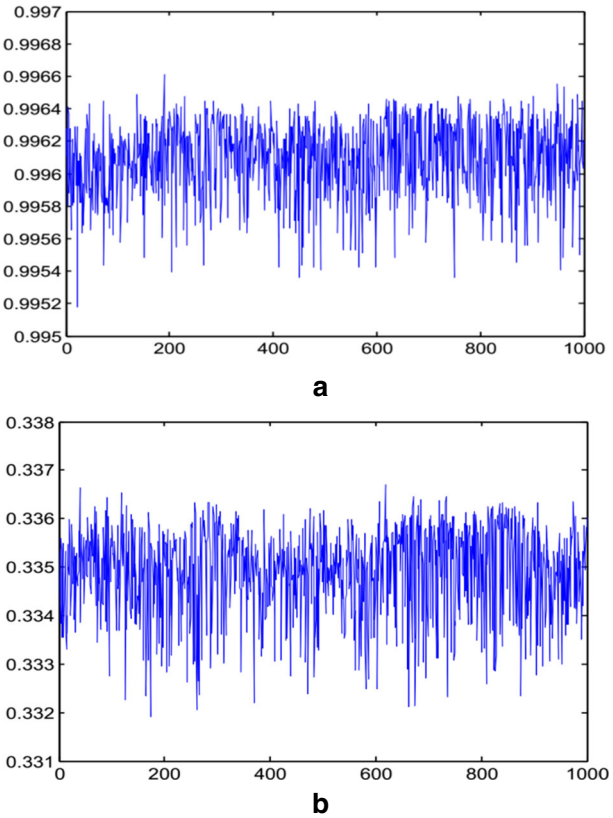

Fig. 5 1000 sets of NPCR and UACI with 1-bit change of plaintext. **a** NPCR, **b** UACI

where P_0 is the pixel value of black image (all zero), and C_0 is cipher image of black image. After that, The attacker attempted to perform *xor* operation on the obtained key streams (KS_0) and the cipher image of Lena image to obtain the original Lena image, but the result he obtained was shown in Fig. 6(i), from which no information of the plaintext image could be seen. Therefore, the proposed encryption algorithm can effectively resist chosen-plaintext attack.

Table 4 NPCR and UACI performance

	Image (256*256)	UACI	The absolute value of the difference from the standard UACI value	NPCR	The absolute value of the difference from the standard NPCR value
Ours	Lena	0.334907	0.000272	0.996090	0.000041
Ref. [21]	Lena	0.3347	0.000065	0.9961	0.000051
Ref. [20]	Lena	0.3350	0.000365	0.9959	0.000149
Ref. [4]	Lena	0.3364	0.001765	0.9965	0.000451
Ref. [22]	Lena	0.3354	0.000765	0.9962	0.000151

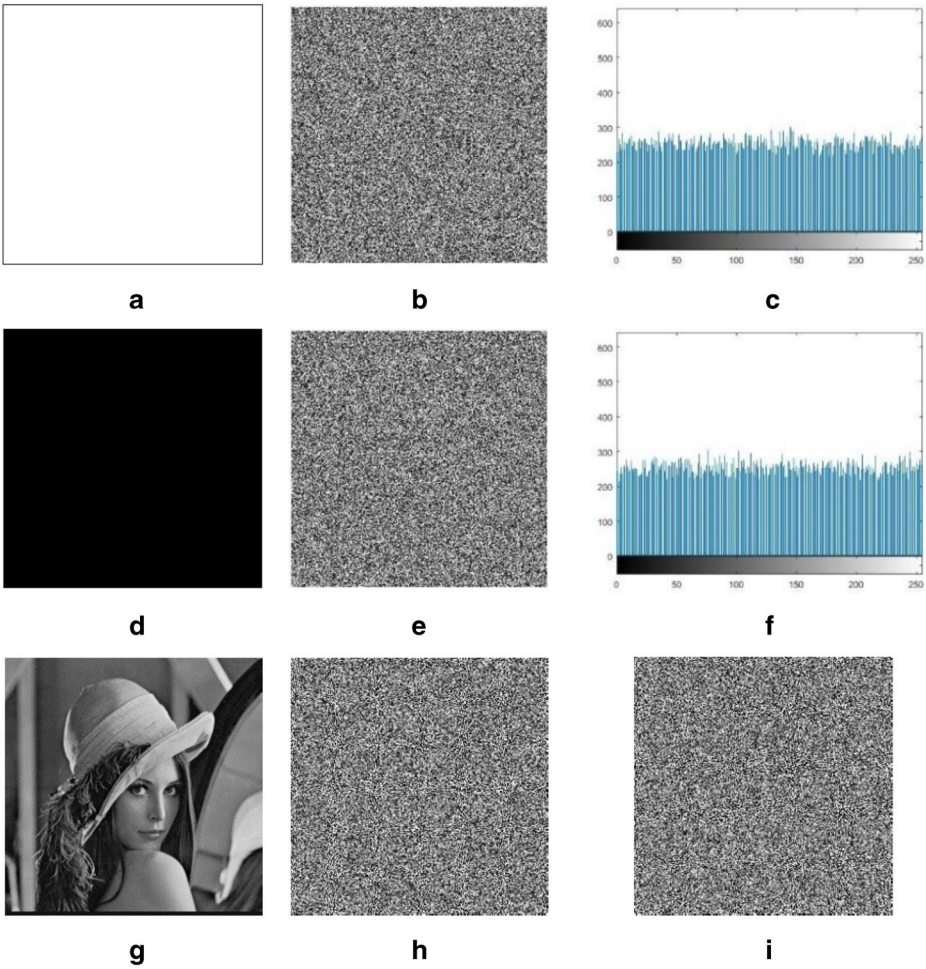

Fig. 6 Chosen-plaintext attack: **a** white image; **b** cipher image of white image; **c** histogram of cipher image; **d** black image; **e** cipher image of black image; **f** histogram of cipher image; **g** Lena image; **h** cipher image of Lena; **i** decoded Lena image by attacker

4.8 Key sensitivity test

A high security cryptosystem must be sensitive to the key. In the encryption phase, key sensitivity means that a small change in the key causes the encrypted ciphertext to be completely different. In the decryption phase, key sensitivity means that the ciphertext image will not be decrypted correctly after the key has been slightly changed.

There are four keys in this paper and they are set as($x_1 = 1$, $x_2 = 0.949$, $x_3 = 1$, $x_4 = 1$). Use Barbara as original image, and only make minor changes to the key x_1 ($x_1 = 1 + 10^{-15}$), while other keys are unchanged. Finally experiment with the encryption process and the decryption process respectively. The experimental results of the encryption process are shown in Fig. 7. Figure 7(a) shows the original Barbara image, and Fig. 7(b) and (c) show the two dense images after encrypting the Barbara image with the correct key and the wrong key, respectively. The text image, Fig. 7 (d) shows the difference between the two ciphertext images, it is obvious

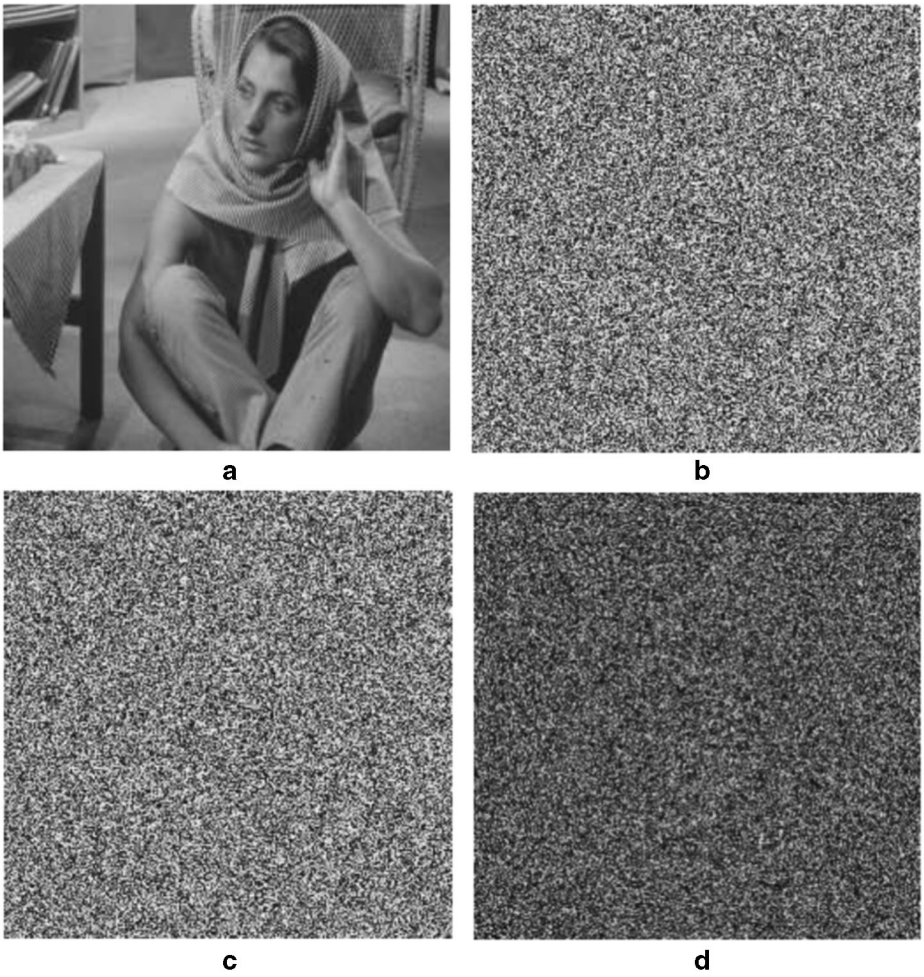

Fig. 7 Key sensitivity for image encryption: **a** Barbara image, **b** encrypted image using the original key, **c** encrypted image using the modified key and **d** the difference between (**b**) and (**c**)

that the two ciphertext images are completely different. The experimental results of the decryption process are shown in Fig. 8. Figure 8(a) shows the original image, Fig. 8(b) shows the ciphertext image encrypted with the correct key, and Fig. 8(c) shows the decryption with the correct key. The image can be seen exactly the same as the original image. Figure 8(d) shows the image decrypted with the wrong key, which is completely different from the original image. From these two experiments, it can be concluded that the algorithm is extremely sensitive to the key.

4.9 Peak signal-to-noise ratio analysis

PSNR (peak signal-to-noise ratio) is an indicator used to evaluate the quality of image encryption [27]. The smaller the value, the better the encryption effect. The Eq. (32) and Eq. (33) can be used to calculate the value of the PSNR:

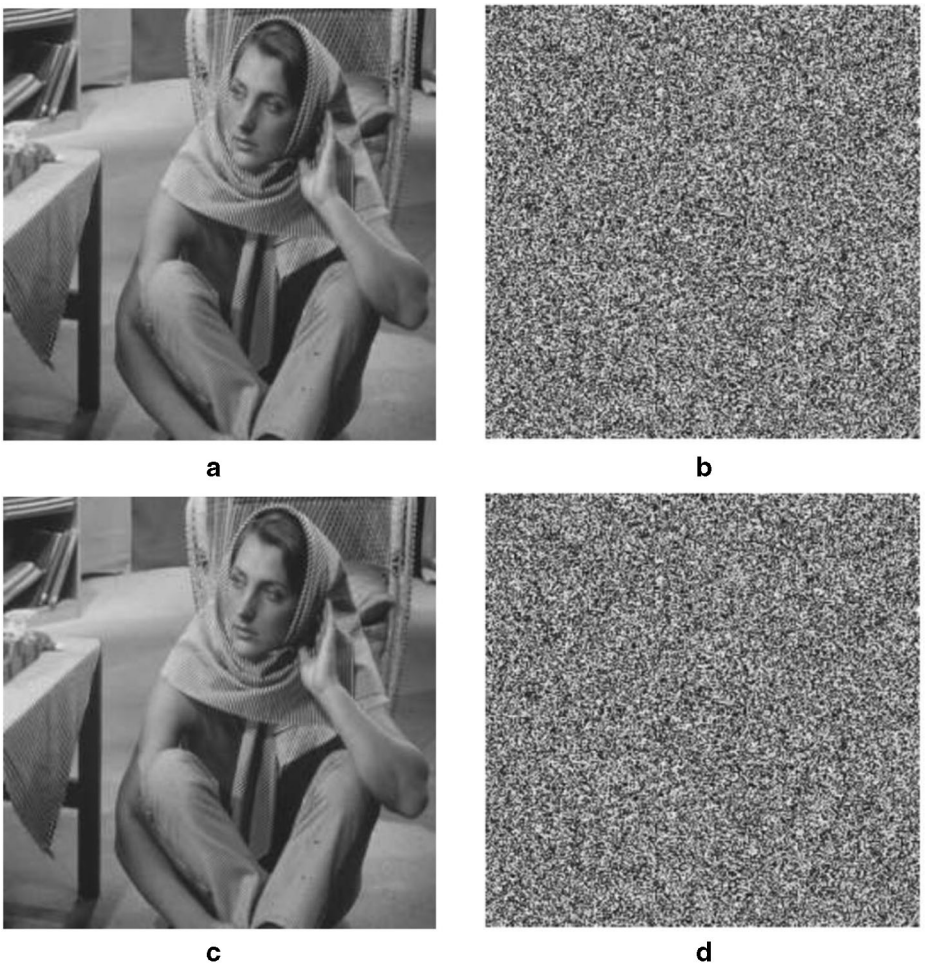

Fig. 8 Key sensitivity for image decryption: **a** Barbara image, **b** encrypted image using the original key, **c** decrypted image using the incorrect decryption key and **d** decrypted image using the correct decryption key

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{\mathrm{M}} \sum_{j=1}^{\mathrm{N}} [P(i,j) - C(i,j)]^2 \tag{32}$$

$$\mathrm{PSNR} = 10 \times \lg\left(\frac{I_{\max}^2}{\mathrm{MSE}}\right) \tag{33}$$

Table 5 PSNR for the encryption

Images	Lena	Peppers	Brain
Ours	8.5494	8.8889	5.7699
Ref. [24]	9.2783	9.4372	9.0104
Ref. [3]	9.2532	8.8971	9.3215
Ref. [28]	9.3228	8.9327	9.0443

Table 6 Speed analysis

Image	Ciphered Time(s)		
	Lena	Peppers	Brain
Ours	0.161	0.155	0.167
Ref. [24]	1.835	1.862	1.823
Ref. [3]	1.224	1.263	1.285
Ref. [28]	0.624	0.673	0.651
Ref. [23]	0.527	0.496	0.554
Ref. [25]	0.417	0.448	0.429

Table 5 compares the PSNR values calculated by the algorithms in this paper and the literature [3, 24, 28] for different 256 × 256 grayscale images, according to equations above. As can be seen from the data in the table, the algorithm of this paper has a smaller PSNR value than other algorithms, that is, the encryption effect is better than other algorithms.

4.10 Encryption speed performance analysis

Image encryption speed is a very important indicator in practical applications. In this paper. The algorithm uses scrambling and diffusion encryption structure. The whole algorithm process only performs one round of scrambling and one round of diffusion process, and makes full use of the four pseudo-random sequences generated by the hyperchaotic system. In addition, the encryption process rarely uses complex operators, and most of them use simple operators such as modulo, addition, XOR. Therefore, the algorithm in this paper has a faster encryption speed. In practical applications, the algorithm encryption speed depends on many external factors, such as CPU, operating system, memory structure and so on. In order to prove that the algorithm is faster than other algorithms, we use the algorithm in [3, 23–25, 28] and the algorithm in this paper to perform experimental simulation on the same computer through MATLAB (R2014a). Table 6 shows the time consumed to encrypt different 256 × 256 grayscale images for each encryption algorithm. It can be seen that the algorithm encrypts faster than the rest of the algorithms.

5 Conclusion

In this paper, an image encryption algorithm based on hyperchaotic system is proposed. At first, the plaintext image is randomly divided into three sequences of different lengths under the control of a pseudo-random sequence generated by the hyperchaotic system. Then, the three sequences are separately encrypted. The key sequence used to encrypt the first sequence is affected by the other two sequences, and the key sequence used to encrypt the second sequence is subjected to the first ciphertext sequence and the third sequence. The effect of encrypting the third sequence is that the key sequence is affected by the first two ciphertext sequences. Finally, the final ciphertext sequence is obtained by splicing the three ciphertext sequences together. Therefore, the key sequence used to encrypt the plaintext image is not only affected by the plaintext but also by the ciphertext, which can effectively enhance the ability of the algorithm to resist differential attacks. At the same time, in order to prove the security of the algorithm, we conducted many common security analysis. All experimental results show that the algorithm has high ability to resist differential attacks and statistical attacks.

Acknowledgments This work is supported by the National Natural Science Foundation of China (No.61971185) and Natural Science Foundation of Hunan Province(2020JJ4218).

References

1. Alvarez G, Li S (2006) Some basic cryptographic requirements for chaos-based cryptosystems. Int J Bifurcat Chaos 16(08):2129–2151
2. Arroyo D, Diaz J, Rodriguez FB (2013) Cryptanalysis of a one round chaos-based substitution permutation network. Signal Processing 93(5):1358–1364
3. Chai XL, Gan ZH, Lu Y et al (2016) A novel color image encryption algorithm based on genetic recombination and the four-dimensional memristive hyperchaotic system[J]. Chinese Physics B 25(10): 100503
4. Cheng G, Wang C, Xu C (2020) A novel hyper-chaotic image encryption scheme based on quantum genetic algorithm and compressive sensing[J]. Multimed Tools Appl 79(39):29243–29263
5. Deng Q, Wang C, Yang L (2020) Four-wing hidden attractors with one stable equilibrium point[J]. Int J Bifurcat Chaos 30(06):2050086
6. Ghebleh M, Kanso A, Noura H (2014) An image encryption scheme based on irregularly decimated chaotic maps. Signal Processing Image Communication 29(5):618–627
7. Kaur M, Kumar V (2018) Efficient image encryption method based on improved Lorenz chaotic system[J]. Electronics Letters 54(9):562–564
8. Kwok HS, Tang WKS (2007) A fast image encryption system based on chaotic maps with finite precision representation. Chaos, Solitons and Fractals 32(4):1518–1529
9. Li C, Lin D (2016) Lü, J H. Cryptanalyzing an Image-Scrambling Encryption Algorithm of Pixel Bits. IEEE Multimedia 24(3):64–71
10. Lin H, Wang C, Tan Y (2020) Hidden extreme multistability with hyperchaos and transient chaos in a Hopfield neural network affected by electromagnetic radiation[J]. Nonlinear Dynamics 99(3):2369–2386
11. Lin H, Wang C, Yao W et al (2020) Chaotic dynamics in a neural network with different types of external stimuli[J]. Commun Nonlinear Sci Numer Simul 105390
12. Liu M, Feng J, Tse CK (2010) A new hyperchaotic system and its circuit implementation. Int J Bifurcat Chaos 20(04):1201–1208
13. Liu L, Zhang Q, Wei X (2012) A RGB image encryption algorithm based on DNA encoding and chaos map[J]. Computers & Electrical Engineering 38(5):1240–1248
14. Liu H, Wang X, Kadir A (2013) Color image encryption using choquet fuzzy integral and hyper chaotic system. Optik - International Journal for Light and Electron Optics 124(18):3527–3533
15. Liu Y, Tang J, Xie T (2014) Cryptanalyzing a RGB image encryption algorithm based on DNA encoding and chaos map. Optics & Laser Technology 60:111–115
16. Patidar V, Pareek NK, Sud KK (2009) A new substitution–diffusion based image cipher using chaotic standard and logistic maps. Communications in Nonlinear Science and Numerical Simulation 14(7):3056–3075
17. Rhouma R, Solak E, Belghith S (2010) Cryptanalysis of a new substitution-diffusion based image cipher. Commun Nonlinear Sci Num Simul 15(7):1887–1892
18. Wang XY, Yang L, Liu R et al (2010) A chaotic image encryption algorithm based on perceptron model. Nonlinear Dynamics 62(3):615–621
19. Wang C, Xia H, Zhou L (2017) A memristive hyperchaotic multiscroll jerk system with controllable scroll numbers[J]. Int J Bifurcat Chaos 27(06):1750091
20. Wang SC, Wang CH, Xu C (2020) An image encryption algorithm based on a hidden attractor chaos system and the Knuth–Durstenfeld algorithm[J]. Optics and Lasers in Engineering 128:105995
21. Xu C, Sun J, Wang C (2020) An image encryption algorithm based on random walk and hyperchaotic systems[J]. Int J Bifurcat Chaos 30(04):2050060
22. Xu C, Sun J, Wang C (2020) A novel image encryption algorithm based on bit-plane matrix rotation and hyper chaotic systems. Multimed Tools Appl, 79(9–10):5573–5593.
23. Ye G (2010) Image scrambling encryption algorithm of pixel bit based on chaos map. Pattern Recognition Lett 31(5):347–354
24. Ye G, Huang X (2017) An efficient symmetric image encryption algorithm based on an intertwining logistic map[J]. Neurocomputing 251:45–53
25. Yin Q, Wang C (2018) A new chaotic image encryption scheme using breadth-first search and dynamic diffusion[J]. Int J Bifurcat Chaos 28(4):1850047

26. Zhen P, Zhao G, Min LQ, Jin X (2016) Chaos-based image encryption scheme combining DNA coding and entropy. Multimed Tools Appl 75(11):6303–6319
27. Zhou M, Wang C (2020) A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks[J]. Signal Processing 171:107484
28. Zhou Y, Cao W, Chen CLP (2014) Image encryption using binary bitplane[J]. Signal Processing 100:197–207
29. Zhu M, Wang C, Deng Q et al (2020) Locally active memristor with three coexisting pinched hysteresis loops and its emulator circuit[J]. Int J Bifurcat Chaos 30(13):2050184