Multimedia Tools and Applications
https://doi.org/10.1007/s11042-022-12239-x

A novel hyperchaotic image encryption algorithm with simultaneous shuffling and diffusion

Xiangquan Gui[1] · Jun Huang[1] · Li Li[1] · Shouliang Li[2] · Jie Cao[1]

Received: 4 December 2020 / Revised: 23 July 2021 / Accepted: 14 January 2022

Abstract
The performance of an image encryption algorithm based on chaos is largely determined by the nonlinear characteristics of the underlying chaotic system. This paper proposes a mixed one- and two-dimensional chaotic map (MOTDCM) that has a wider hyperchaotic interval, a larger maximum Lyapunov exponent, and more complex nonlinear dynamics than most existing chaotic systems. Using the hyperchaotic sequences generated by the MOTDCM, a novel image encryption algorithm with different structures is proposed, in which shuffling and diffusion are carried out simultaneously from the perspective of the whole input image. Simulation results and a comparative analysis show that the proposed encryption algorithm has a large key space, high sensitivity to the secret key, and good statistical ciphertext properties. It has a better diffusion effect than existing algorithms and meets the imposed security requirements within only one round of operation, with a reduction in algorithm complexity and an improvement in encryption efficiency. Experimental results demonstrate that this encryption algorithm has good performance and can resist chosen-plaintext attacks and known-plaintext attacks effectively.

Keywords Hyperchaotic map · Image encryption · Simultaneous shuffling and diffusion

✉ Jun Huang
brioal@foxmail.com

Xiangquan Gui
xqgui@lut.cn

Li Li
lili0226@139.com

Shouliang Li
lishoul@lzu.edu.cn

Jie Cao
caoj@lut.edu.cn

[1] School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, China

[2] School of Information Science and Engineering, Lanzhou University, Lanzhou, 730000, China

Published online: 17 March 2022

1 Introduction

With the rapid development of computer network technology, much private image information, such as medical images and military satellite images, is transmitted over public networks. How to ensure transmission security is receiving increasing attention. Different from text data, image information is characterized by large data volumes, high redundancy, and strong correlations between pixels. This means that traditional algorithms such as the Data Encryption Standard (DES) and Rivest-Shamir-Adleman (RSA) are not suitable for efficient image encryption. Due to features such as ergodicity, unpredictability, and sensitivity to the initial parameters [11], chaotic systems have been widely used in digital image encryption [18] in recent decades.

In 1998, J. Fridrich [9] applied a chaotic system to digital image encryption and proposed a classic shuffling-diffusion structure. Shuffling involves changing the positions of pixels in the original image and destroying the spatial distribution and local correlations between pixels, making it impossible to recognize and confirm the original image. Diffusion confuses the current pixel value and makes it related to the other pixels so that any change in any pixel in the image can have a great impact on the ciphertext. Through the use of shuffling and diffusion, various types of typical attacks can be effectively resisted. Reference [28] proposed a color image encryption algorithm based on chaos. This algorithm makes the red, green, and blue components of the image interact with each other and encrypts them at the same time, which improves the overall security level. Pak C et al. [19] introduced a nonlinear combination method for one-dimensional chaotic maps with a wide chaotic interval and applied it to color image encryption to increase the security of the algorithm. In reference [26], a color image encryption method that combines rectangular transformation and a chaotic tent map was proposed. It simultaneously encrypts the three components of a color image with a keystream related to the plaintext to meet the imposed security requirements. In recent years, to improve the efficiency and security of encryption algorithms, technologies such as compressed sensing [3], DNA coding [4, 6, 23], and S-boxes [13, 14, 27, 36] have also been applied to chaotic image encryption.

In the algorithms mentioned above, the chaotic sequences used in the encryption process are related to the key instead of the plaintext. Attackers can decrypt the keystream by chosen-plaintext attacks. For example, the encryption scheme in reference [28] has been broken by reference [22] with chosen-plaintext attacks. The hackers can obtain the diffusion rule and the displacement matrix from the relationship between the selected pixels of the plaintext image and the corresponding pixels of the arranged image. To resist chosen-plaintext attacks, reference [15] presented an image encryption algorithm whose key stream is related to the plaintext. During the shuffling process, the initial parameters are calculated with the key and the total pixels. This schema makes two different plaintext images have different position transformation matrices even if they have the same key. During the diffusion process, the initial value is related to the key and the nine particular pixels of the shuffled image. The chaotic sequences used for diffusion are different if the nine pixels of the two images are different such that they can resist chosen-plaintext attacks. Although the algorithm has good security performance, there are still some problems. For example, because a single shuffling or diffusion-based encryption operation is too simple to ensure the satisfaction of encryption strength requirements, traditional algorithms usually need to repeat these operations multiple times. In reference [37], k rounds of encryption operations were performed, and each of them included m rounds of shuffling and n rounds of diffusion. More rounds of cycles bring greater time consumption and a greater loss of efficiency.

The above methods have improved the ability of algorithms to resist chosen-plaintext attacks by strengthening the connection between the encryption process and the input plaintext image. However, there are still some problems to be solved. (1) The continuous or discrete chaotic systems used in encryption algorithms are not random enough, and the dynamic characteristics of the generated sequence need to be further improved. (2) The chaotic sequences generated for the encryption process are not correlated with the plaintext, and they are vulnerable to chosen-plaintext attacks. (3) Due to security requirements, shuffling and diffusion operations usually need to be executed multiple times. Too many cycles make the resulting algorithm complicated and inefficient.

To solve the defects discussed above, we introduce a novel hyperchaotic image encryption algorithm with simultaneous shuffling and diffusion. Three innovations are provided in this work. (1) We construct a new chaotic map composed of a mixed one- and two-dimensional chaotic map. This map has a wide hyperchaotic interval without periodic windows and can produce nonlinear chaotic sequences. (2) To ensure the correlation between the secret key and the input plaintext, an initial value generator base on the SHA-256 hash function is presented, and the generator is sensitive to any changes in the pixels in the plaintext, so it can provide reliable initial values and parameters. (3) Based on the hyperchaotic map and the initial value generator, we introduce an encryption algorithm that performs shuffling and diffusion operations in one encryption iteration. The relationship between plaintext and ciphertext is fully strengthened, and only one round of encryption is required to meet the imposed security requirements. Encryption analyses confirm that the algorithm has a good encryption effect and high security.

The remainder of this paper is organized as follows. The proposed hybrid hyperchaotic model is presented in detail in Section 2. Its chaotic characteristics are analyzed and compared with those of the existing chaotic models on trajectory maps, Lyapunov exponents, and permutation entropies. The proposed image encryption scheme is described in Section 3. The experimental results and security analyses are stated in Section 4. Finally, Section 5 concludes this paper and provides the research results.

2 Mixed one- and two-dimensional chaotic map

2.1 MOTDCM model

We proposed a mixed one- and two-dimensional chaotic map (MOTDCM) model, which is a combination of a two-dimensional logistic map and an improved one-dimensional Feigenbaum transcendental map [8]. The model structure is shown in Fig. 1.

The MOTDCM model has the same structure as that of a two-dimensional logistic map, and it uses an improved one-dimensional Feigenbaum transcendental map, which is marked

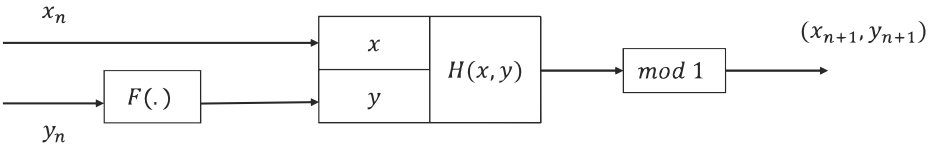

Fig. 1 MOTDCM model structure

as $F(.)$, to pre-process y values. mod represents the modulus operation, and the MOTDCM model can be expressed by (1).

$$\begin{cases} x_{n+1} = mod\left(4 * \alpha * x_n * (1 - x_n) + \gamma_1 * y_n^2, 1\right) \\ y_{n+1} = mod\left(4 * \beta * F(y_n) * (1 - F(y_n)) + \gamma_2 * x_n^2, 1\right) \end{cases} \tag{1}$$

where α, β, γ_1, and γ_2 are control parameters and $mod(x, 1)$ is the remainder of x divided by one. The defination for $F(.)$ is shown in (2).

$$F(x) = 3 * \sin(\pi * x) \tag{2}$$

Therefore, the complete formula expression of the MOTDCM model can be obtained as (3).

$$\begin{cases} x_{n+1} = mod\left(4 * \alpha * x_n * (1 - x_n) + \gamma_1 * y_n^2, 1\right) \\ y_{n+1} = mod\left(12 * \beta * \sin(\pi * y_n) * (1 - 3 * \sin(\pi * y_n)) + \gamma_2 * x_n^2, 1\right) \end{cases} \tag{3}$$

where $\alpha \in [0, 6]$, $\beta \in [0, 6]$, $\gamma_1 \in [0, 2]$, and $\gamma_2 \in [0, 2]$, respectively. When $\beta > 0.1$, the system is hyperchaotic, and the chaotic performance improves as β increases. In this paper, we set $\alpha = 5$, $\beta = 4$, $\gamma_1 = 2$, and $\gamma_2 = 2$ for simplicity.

2.2 Performance evaluation of MOTDCM

We use trajectory, Lyapunov exponents [21] and Permutation entropy [1] to evaluate the chaotic performance of MOTDCM and obtain conclusions in comparison with 2D-SIMM [17], 2D-SLMM [10] and 2D-Logistic [25].

2.2.1 Trajectory

A trajectory diagram shows the distribution interval of a chaotic sequence generated by a chaotic model. A wide and uniform distribution indicates that the chaotic sequence has great nonlinearity. It can be seen from Fig. 2 that the trajectory map of the MOTDCM model is uniform and dispersed throughout the whole interval, which indicates that the MOTDCM model has excellent chaos performance and can generate scattered, random chaotic sequences.

2.2.2 Lyapunov exponent

As a significant index for describing chaotic performance, the Lyapunov exponent represents the exponential growth rate of a random map over continuous iterations. If the maximum Lyapunov exponent of a chaotic map is greater than 0, the existence of chaos can be confirmed. The larger the maximum Lyapunov exponent the map obtains, the better its nonlinear performance.

It can be seen from Fig. 3(c) that $\gamma_1 > 0$ when $\alpha \in [0.88, 1]$, which means that the 2D-SLMM has entered the chaotic state. When $\alpha \in [0.91, 1]$, γ_1 and γ_2 are both greater than 0, the system is in the hyperchaotic state [20]. This map has hyperchaotic characteristics, but

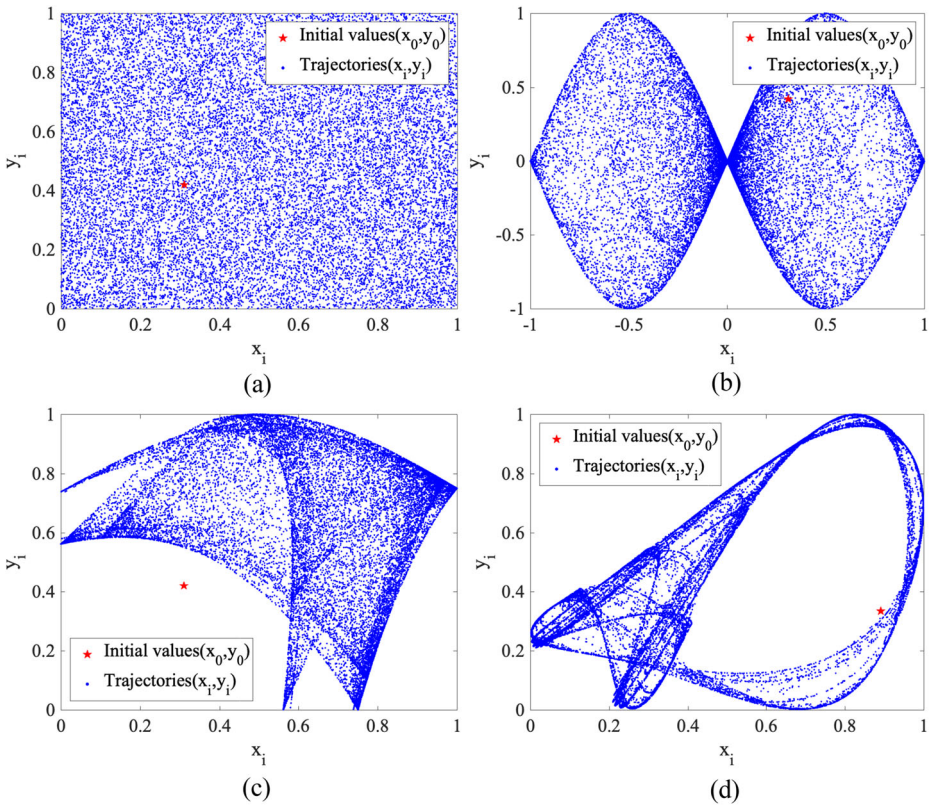

Fig. 2 Trajectories of (a) the MOTDCM, (b) 2D-SIMM, (c) 2D-SLMM, (d) 2D-Logistic

the interval is short, and the maximum value is small, so the nonlinear characteristics of this map are not rich enough.

In the same way, from Fig. 3(b), we can see that when $\alpha \in [0.7, 2.7]$, 2D-SIMM enters the hyperchaotic state with fluctuations during the interval, which indicates that the map has an excellent but unstable nonlinear characteristic. From Fig. 3(d), we can observe that the hyperchaotic characteristics exist in 2D-Logistic map when $r \in [1.3, 4]$, but the maximum value of its largest Lyapunov exponent is small than 2.

As shown in Fig. 3(a), the MOTDCM has entered a hyperchaotic state since $\beta > 0.1$. The fact that the MOTDCM obtains the largest Lyapunov exponents indicates that the map has excellent chaotic characteristics, which further illustrates that it is suitable for image encryption algorithms.

2.2.3 Permutation entropy

Permutation entropy indicates the complexity of a random sequence. Figure 4 gives the permutation entropies of the MOTDCM in comparison with the result of 2D-SLMM, 2D-SIMM, and 2D-Logistic map. From the figure, we can see that the permutation entropies of 2D-SLMM and the other maps are small and fluctuate greatly in the whole interval.

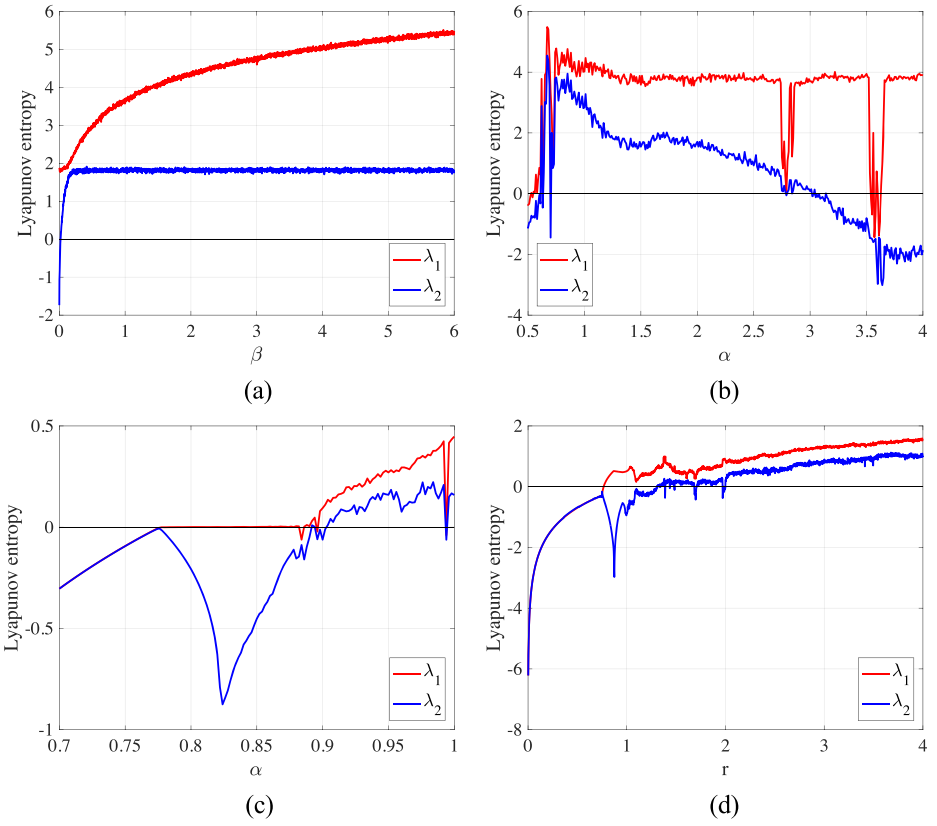

Fig. 3 Lyapunov exponents of (a) the MOTDCM, (b) 2D-SIMM, (c) 2D-SLMM, (d) 2D-Logistic

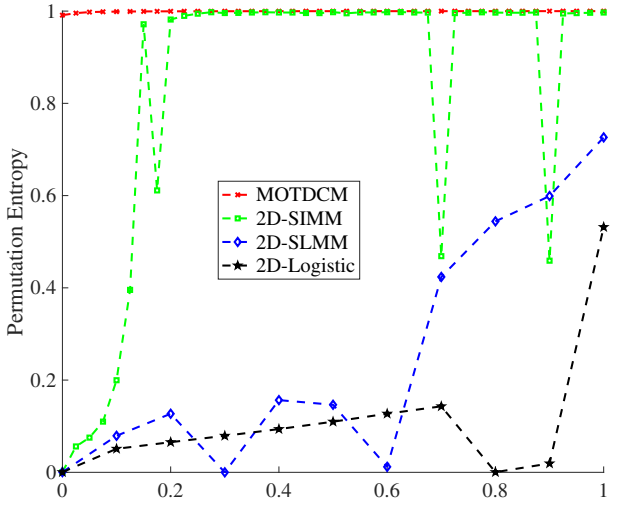

Fig. 4 Permutation entropies of the MOTDCM, 2D-SIMM, 2D-SLMM, and 2D-Logistic map

However, the permutation entropy of the MOTDCM is always close to 1, and it is stable throughout the whole interval, without periodic windows. This means that the sequences generated by the MOTDCM have high and stable complexity.

3 Image encryption algorithm dynamically related to pixels with simultaneous shuffling and diffusion

Commonly used image encryption algorithms usually include shuffling and diffusion processes. Motivated by security needs, the entire encryption process generally requires multiple rounds. Simple shuffling and diffusion yield an increase in speed but lack security. Complex or multiple rounds of shuffling and diffusion ensure encryption security but require more time.

This paper proposes a single-round image encryption scheme, and Fig. 5 shows the structure of this approach. The algorithm has only one round of encryption, which contains forward shuffling and diffusion operations and reverse diffusion operations. The two types of operations are based on the x output and y output of the MOTDCM. The pixels used for diffusion in the first stages are randomly obtained from the input plaintext according to the chaotic sequence. This random selection process composes the shuffle operation. The reverse diffusion operation is a simplified version of the forward shuffling and diffusion operation. The difference is that the direction of operation is reversed and that it contains only the diffusion step. The flow chart of the encryption algorithm is shown in Fig. 6.

3.1 Secret key structure

To ensure the correlation between the secret key and the plaintext, the hash value generated by the SHA-256 scheme is used as the secret key of this encryption system. The secret key H is 256 bits in length and highly related to the plaintext. Any changes in the pixels in the plaintext will cause enormous changes in the secret key.

Image encryption depends on chaotic sequences, and the generation of chaotic sequences depends on the chosen parameters and initial values. This paper proposes a new password generation strategy in which any small change in the secret key will completely change the parameters and initial values. First, we split the 256-bit key H into K_1, K_2, $K_3 ... K_{32}$, as shown in Fig. 7.

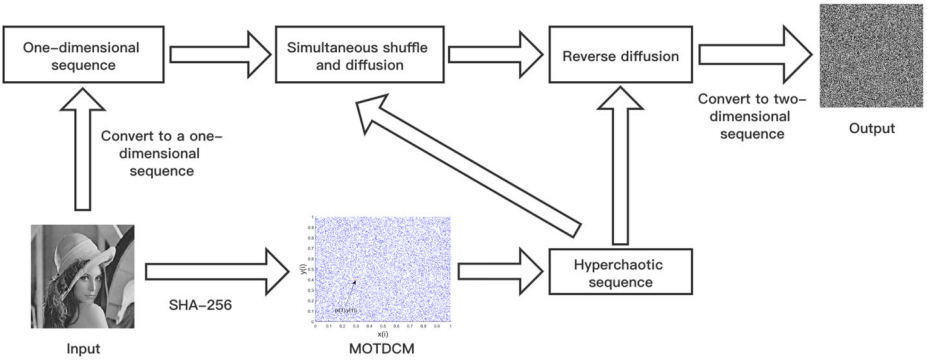

Fig. 5 Encryption algorithm structure

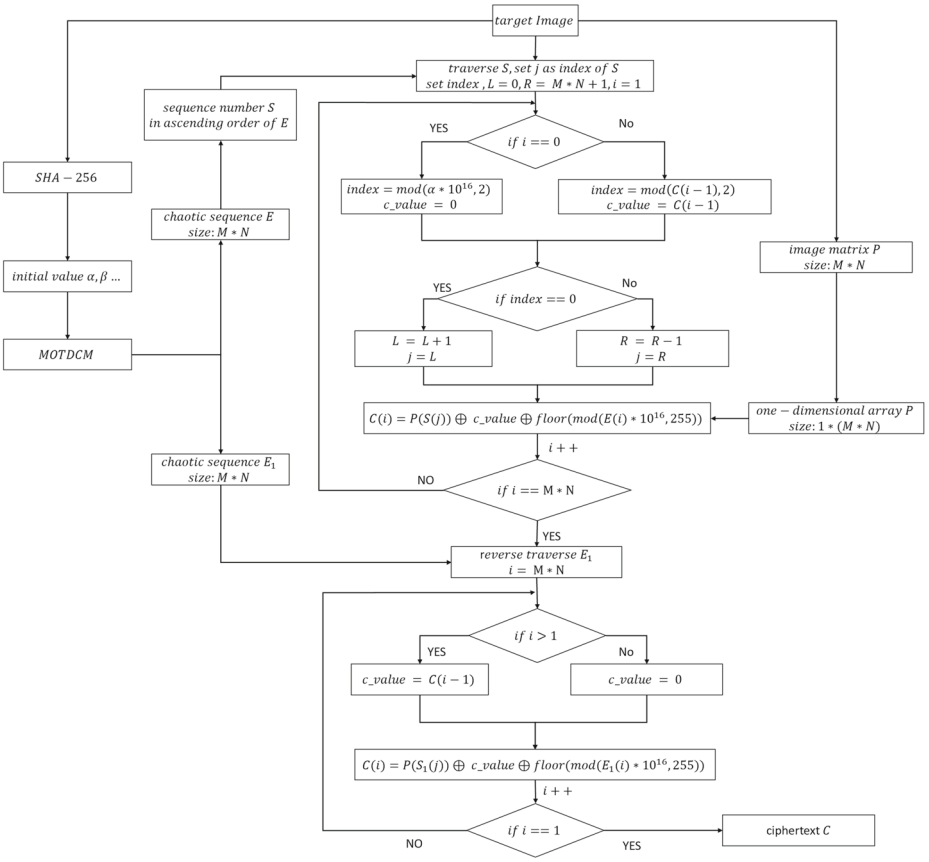

Fig. 6 Flow chart of the encryption algorithm

Second, the h_1, h_2, h_3, h_4 values are calculated with K_1, K_2, $K_3 ... K_{32}$, and the calculation rule is shown below.

$$\begin{cases} h_1 = 8 * \frac{k_1 \oplus k_2 \oplus ... \oplus k_8}{256} \\ h_2 = h_1 + 4 * \frac{k_9 \oplus k_{10} \oplus ... \oplus k_{16}}{256} \\ h_3 = h_2 + 2 * \frac{k_{17} \oplus k_{18} \oplus ... \oplus k_{24}}{256} \\ h_4 = h_3 + \frac{k_{25} \oplus k_{26} \oplus ... \oplus k_{32}}{256} \end{cases} \tag{4}$$

where $\oplus$ is the operation where two numbers are bit-XORed by their binary values.

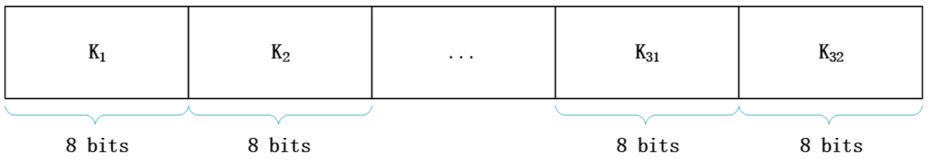

Fig. 7 Splitting the 256-bit key into K_1, K_2, $K_3 ... K_{32}$

Then, $\alpha, \beta, \gamma_1, \gamma_2, x_0, y_0$ are calculated based on h_1, h_2, h_3, h_4. Because the initial value of the MOTDCM has a certain range, to ensure the chaos of the map, we set $\overline{\alpha} = 5, \overline{\beta} = 4, \overline{\gamma_1} = 2, \overline{\gamma_2} = 2, \overline{x_0} = 0.2, \overline{y_0} = 0.3$. The initial value generation rules are shown in (5).

$$\begin{cases} \alpha = \overline{\alpha} + \frac{mod((h_1+h_2)*10^{14},255)}{256} \\ \beta = \overline{\beta} + \frac{mod((h_1+h_3)*10^{14},255)}{256} \\ \gamma_1 = \overline{\gamma_1} + \frac{mod((h_1+h_4)*10^{14},255)}{256} \\ \gamma_2 = \overline{\gamma_2} + \frac{mod((h_2+h_3)*10^{14},255)}{256} \\ x_0 = \overline{x_0} + \frac{mod((h_2+h_4)*10^{14},255)}{256} \\ y_0 = \overline{y_0} + \frac{mod((h_3+h_4)*10^{14},255)}{256} \end{cases} \tag{5}$$

3.2 Forward shuffling and diffusion process

During the process of forward shuffling and diffusion, a two-dimensional input plaintext with a size of $M * N$ needs to be processed into a one-dimensional sequence. The specific description of this technique is as follows:

Input: A plaintext image matrix P with a size of $M * N$, the initial values $K = \{\alpha, \beta, \gamma_1, \gamma_2, x_0, y_0\}$.

Step 1: Convert the $M * N$ two-dimensional matrix P into a one-dimensional sequence. Initialize the ciphertext storage sequence C with the same size as that of P.

Step 2: Input the initial values $K = \{\alpha, \beta, \gamma_1, \gamma_2, x_0, y_0\}$ into the MOTDCM for $M * N + N_0$ iterations, where N_0 is a constant between 100 and 500. To overcome the transient effect, discard the first N_0 sequences and obtain chaotic sequences E and E_1, which are derived from the x output and y output of the MOTDCM, respectively. Additionally, obtain a number sequence S from the sequence E in ascending order. The process of obtaining sequence S is shown in Fig. 8.

Step 3: Traverse the sequence S, and record the current traversal index as j. The traversal direction is given by the initial value and the ciphertext pixel value. Set the flag $index$, the traversal direction is from left to right when $index = 0$, and turn around when $index = 1$. After each traversal, perform a recalculation to determine the position of the next traversal, and the calculation of the $index$ is shown in (6), where $i = 1, 2, ..., M * N$.

$$index = \begin{cases} mod(\alpha * 10^{16}, 2), i = 1 \\ mod(C(i-1), 2), i > 1 \end{cases} \tag{6}$$

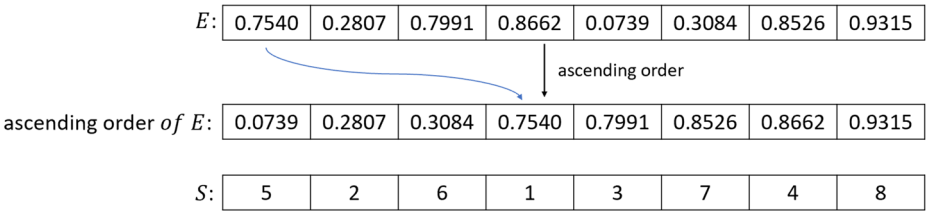

Fig. 8 Obtaining the number sequence S from E in ascending order

Shuffling and diffuse the plaintext sequence P to obtain the ciphertext C during the traversal of S, where $i = 1, 2, ..., M * N$. The operation rules are shown in (7).

$$C(i) = \begin{cases} P(S(j)) \oplus floor(mod(E(i) * 10^{16}, 255)), i = 1 \\ P(S(j)) \oplus C(i-1) \oplus floor(mod(E(i) * 10^{16}, 255)), i > 1 \end{cases} \tag{7}$$

where $floor(x)$ is the largest integer not greater than x. Figure 9 shows the process of forward shuffling and diffusion.

3.3 Reverse diffusion process

The ciphertext C obtained in Step 3 is traversed in reverse using the chaotic sequence E_1 to perform the diffusion operation during the traversal process. The diffusion operation is shown in (8), where $i = M * N, M * N - 1, M * N - -2, ..., 1$.

$$C(i) = \begin{cases} C(i) \oplus floor(mod(E_1(i) * 10^{16}, 255)), i = 1 \\ C(i) \oplus C(i-1) \oplus floor(mod(E_1(i) * 10^{16}, 255)), i > 1 \end{cases} \tag{8}$$

After the forward shuffling-diffusion and reverse diffusion processes, we can obtain the encrypted image C.

3.4 Decryption process

The decryption process is the reverse of the encryption process. The specific steps are as follows:

Input: The encrypted image C and the initial values $K = \{\alpha, \beta, \gamma_1, \gamma_2, x_0, y_0\}$.

Step 1: Obtain chaotic sequences E , E_1 and S in the same way as in the encryption process.

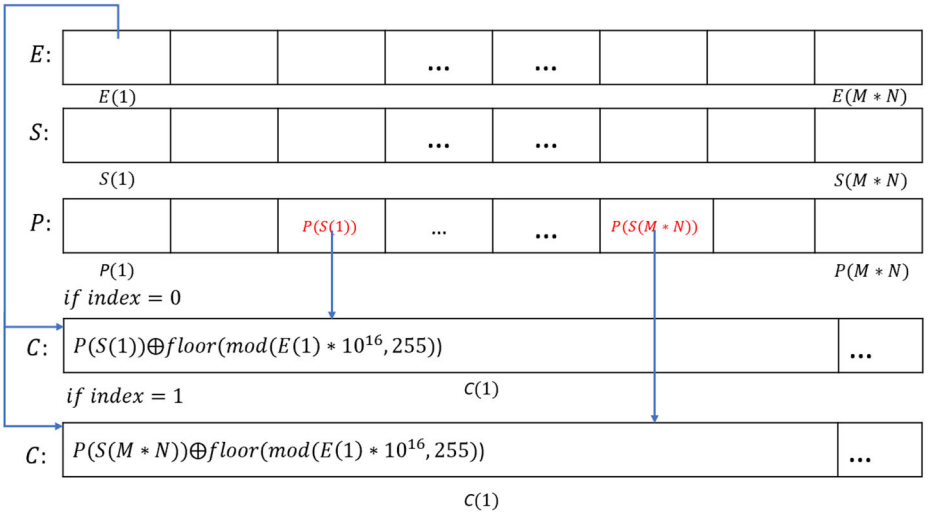

Fig. 9 Process of forward shuffling and diffusion

Step 2: Reverse the diffusion process from the encryption process. Traverse the ciphertext C, and use E_1 to achieve diffusion. The calculation method is shown in (9).

$$C(i) = \begin{cases} C(i) \oplus floor(mod(E_1(i) * 10^{16}, 255)), i = 1 \\ C(i) \oplus C(i-1) \oplus floor(mod(E_1(i) * 10^{16}, 255)), i > 1 \end{cases} \tag{9}$$

Step 3: Reverse the diffusion process of the forward shuffling and diffusion operation from the encryption procedure. Traverse the ciphertext C in reverse and use E to decrypt it. The decryption method is shown in (10).

$$C(i) = \begin{cases} C(i) \oplus floor(mod(E(i) * 10^{16}, 255)), i = 1 \\ C(i) \oplus C(i-1) \oplus floor(mod(E(i) * 10^{16}, 255)), i > 1 \end{cases} \tag{10}$$

Step 4: According to Step 3 in the encryption process, calculate j for traversal i of the incremental process, and the decryption process is shown in (11).

$$P(S(j)) = C(i) \tag{11}$$

The P obtained after this step is the plaintext image after decryption.

4 Simulation results and attack tests

4.1 Simulation results

We use Lena, Baboon, Cameraman, and Peppers as experimental images. The experimental results are shown in Fig. 10. From the figure, we can see that the pixels of the encrypted pictures are randomly distributed, and the encrypted pictures can be entirely and clearly decrypted into the original images.

4.2 Key space and key sensitivity analyses

The size of the utilized key space determines the ability of a given encryption algorithm to resist brute force attacks. The key we use is a 256-bit binary number. Based on the current computer operating speed, this key has sufficient strength to resist violent attacks.

To test the key sensitivity of the algorithm, we change the value of a bit within the 256-bit key randomly and then use it to decrypt the encrypted image. Figure 11 shows the result obtained for the Lena image after decryption. It can be seen from the figure that even if one bit of the 256-bit key is changed, the decryption result changes drastically, which indicates that the algorithm is extremely sensitive to the key.

4.3 Histogram analyses

Figure 12 shows the histogram distribution of the pixel values of the original images and the encrypted images. The pixel distributions of the original images are regular and concentrated, and the pixel distributions after encryption are evenly distributed, which indicates that the encryption process disrupts the regularity of the original images and can block attacks that use image regularity.

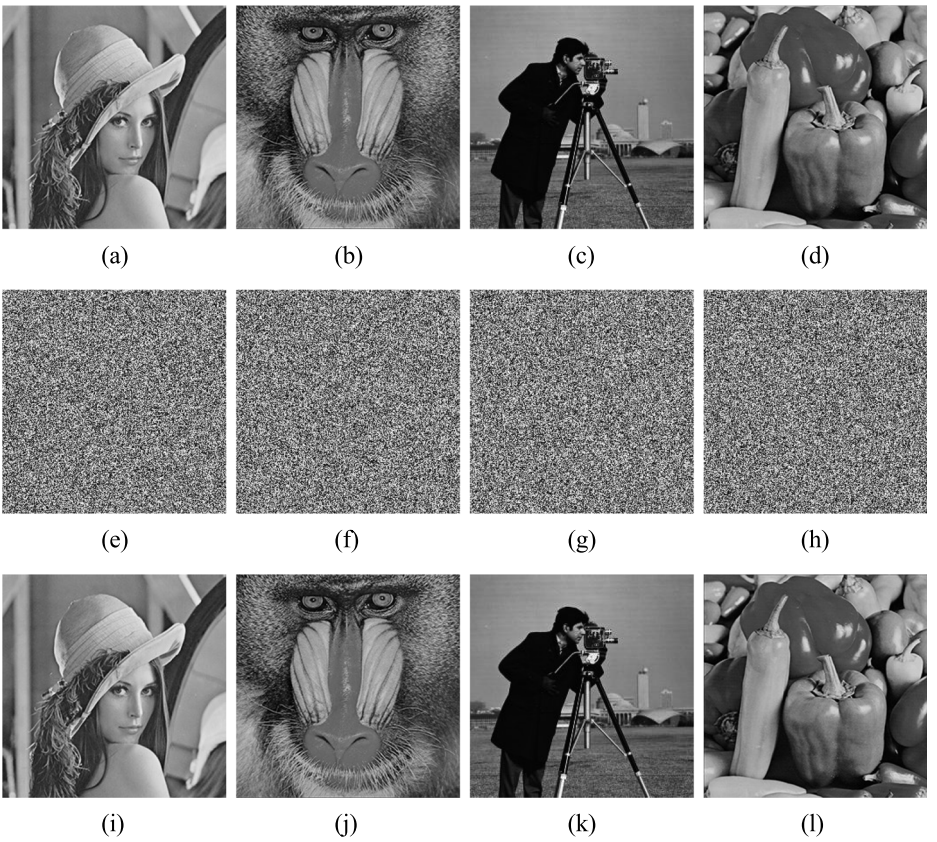

Fig. 10 Simulation results. (a-d) are the original Lena, Baboon, Cameraman, and Peppers images, respectively. (e-h) are the encrypted Lena, Baboon, Cameraman, and Peppers images, respectively. (i-l) are the decrypted Lena, Baboon, Cameraman, and Peppers images, respectively

4.4 Correlation coefficient analyses

There are high degrees of correlation between the pixels of the original images, so they contains information that can be recognized by human eyes. The encryption algorithm should minimize these correlations as much as possible. To verify the correlation of the encrypted images, we use (12) to calculate the correlations between the image pixels.

$$\rho_{xy} = \frac{E\{[x - E(x)][y - E(y)]\}}{\sqrt{D(x)}\sqrt{D(y)}} \tag{12}$$

where $E(x) = \frac{1}{l}\sum_{i=1}^{l} x_i$ and $D(x) = \frac{1}{l}\sum_{i=1}^{l}[x_i - E(x)]^2$ represent the mean and the variance of l pixels, respectively. To evaluate the correlation coefficients of each image in the horizontal, vertical, and diagonal directions, we select 10,000 pixels randomly from each direction of the image and then calculate the correlations.

The calculation results are shown in Table 1. The table indicates that after encryption, the correlations between pixels are greatly reduced. This means that encryption makes the resulting image exhibit almost no regularity. Furthermore, we compare the correlation

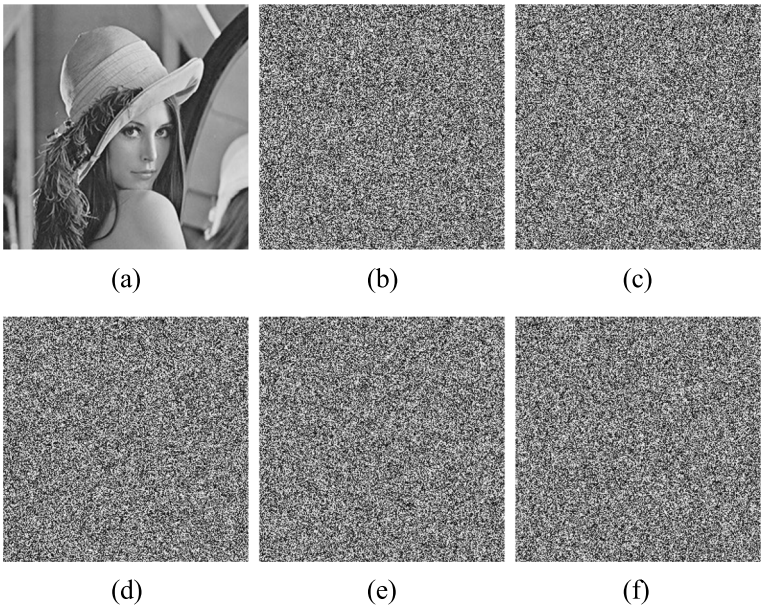

Fig. 11 Sensitivity test results for the Lena image. (a) Decrypted image with the correct key. (b-f) Decrypted result after the key is randomly changed by one bit

coefficients of the encrypted Lena image using different algorithms. By matching, we find that the image encryption algorithm proposed in this paper functions more effectively than the other methods.

Figure 13 shows the distributions of adjacent pixels in the original Lena image and the encrypted image. It can be seen from the figure that the pixels in all directions of the original image are concentrated around $y = x$, but the pixels of the encrypted image are evenly distributed throughout the interval.

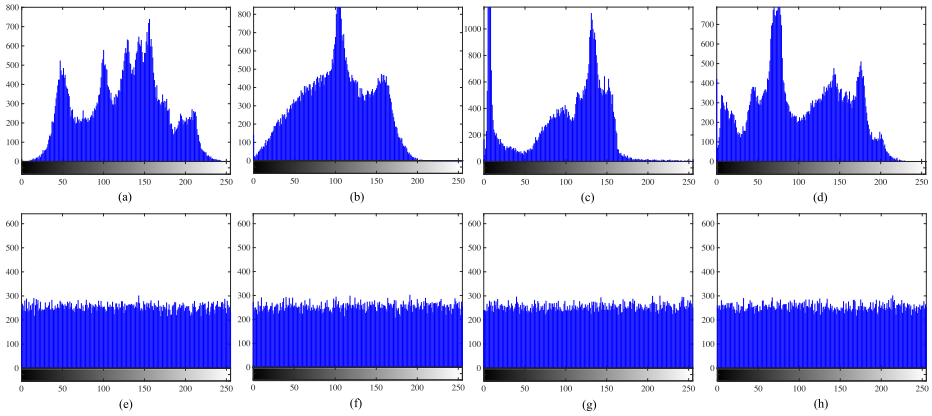

Fig. 12 Histograms. (a-d) Histograms of Lena, Baboon, Cameraman, and Peppers, respectively. (e-h) Histograms of the encrypted Lena, encrypted Baboon, encrypted Cameraman, and encrypted Peppers images, respectively

Table 1 Correlation coefficient analysis

Images	Horizontal		Vertical		Diagonal	
	Plain	Cipher	Plain	Cipher	Plain	Cipher
Baboon	0.7399	−0.0032	0.7962	0.0003	0.6928	0.0154
Cameraman	0.9482	0.0022	0.9261	0.0007	0.8974	0.0102
Peppers	0.9561	0.0014	0.9516	−0.0068	0.9209	−0.0002
Lena	0.9567	−0.0022	0.9171	0.0018	0.8921	−0.0019
Lena in Ref. [7]	–	0.003	–	0.005	–	−0.002
Lena in Ref. [10]	–	0.0024	–	−0.0086	–	0.0402
Lena in Ref. [33]	–	−0.0226	–	0.0041	–	0.0368
Lena in Ref. [35]	–	0.0039	–	−0.0035	–	0.0009

4.5 Information entropy analyses

The information entropy of an image represents the degree of confusion between pixels. The higher the information entropy after encryption, the better the encryption effect is. The information entropy calculation method is shown in (13).

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i) \log_2 \frac{1}{p(m_i)} \tag{13}$$

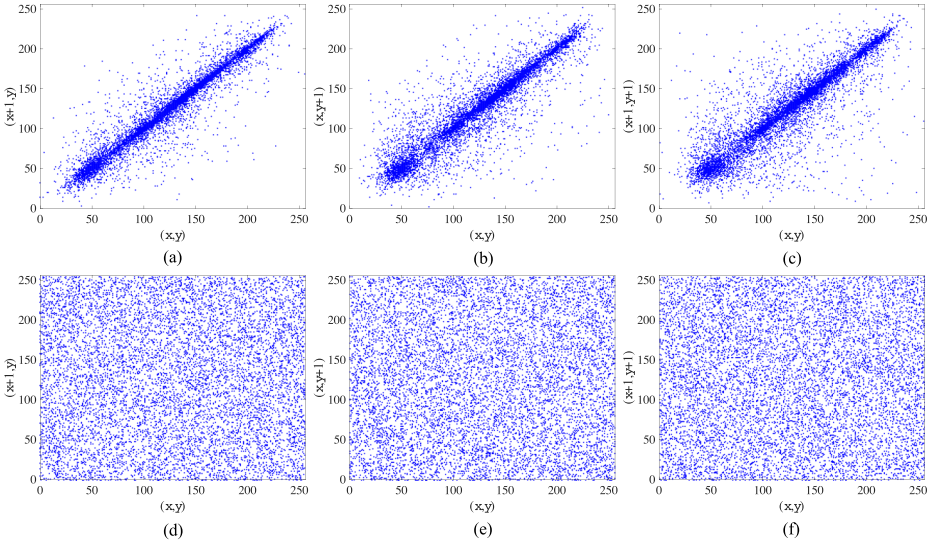

Fig. 13 Distributions of adjacent pixels in the original image and encrypted image of Lena. (**a**-**c**) Distributions of the original image in the horizontal, vertical, and diagonal directions, respectively. (**d**-**f**) Distributions of the encrypted image in the horizontal, vertical, and diagonal directions, respectively

Table 2 Entropy analyses

Test Image	Information entropy	
	Plain	Cipher
Baboon	7.3739	7.9973
Cameraman	6.9488	7.9974
Peppers	7.5676	7.9974
Lena	7.5151	7.9974
Lena in Ref. [34]	–	7.9971
Lena in Ref. [29]	–	7.9970
Lena in Ref. [33]	–	7.9973

Where m represents the information source. For 256 gray-level images, the theoretical maximum information entropy is 8. Table 2 shows the calculation results for the encrypted Lena based on different algorithms. From the table, we can see that the information entropy of the image after encryption is very close to the theoretical value, indicating that the pixels after encryption are highly confused.

4.6 Differential attack analyses

Differential attacks strike algorithms by comparing and analyzing the spreads of encrypted plaintexts with specific differences. We use the following three indicators to evaluate the similarity between plaintext and ciphertext, including the change rate of the number of

Table 3 The NPCR, UACI, BACI values of different images

Test Images	Type	Min(%)	Max(%)	Mean(%)
Lena	NPCR	99.5498(−0.0596)	99.6612(+0.0518)	99.6107(+0.0013)
	UACI	33.2519(−0.2116)	33.6671(+0.2036)	33.4576(−0.0059)
	BACI	26.7639(−0.0073)	26.9203(+0.1491)	26.7744(+0.0032)
Baboon	NPCR	99.5574(−0.052)	99.6688(+0.0594)	99.6089(−0.0005)
	UACI	33.2551(−0.2084)	33.6605(+0.1970)	33.4479(−0.0156)
	BACI	26.5708(−0.2004)	26.8977(+0.1265)	26.7483(−0.0229)
Cameraman	NPCR	99.5559(−0.0535)	99.6673(+0.0579)	99.6101(+0.0007)
	UACI	33.2863(−0.1772)	33.7165(+0.253)	33.5030(+0.0395)
	BACI	26.6467(−0.1245)	26.9700(+0.1988)	26.8035(+0.0323)
Peppers	NPCR	99.5468(−0.0626)	99.6795(+0.0701)	99.6088(−0.0006)
	UACI	33.2059(−0.2576)	33.6053(+0.1418)	33.4295(−0.0340)
	BACI	26.5596(−0.2116)	26.8940(+0.1228)	26.7348(−0.0364)

The numbers in parentheses indicate the gaps from the corresponding standard values

Table 4 The NPCR, UACI, BACI values of different algorithms

Algorithm	NPCR	UACI	BACI
Proposed	99.6107(+0.0013)	33.4576(−0.0059)	26.7744(+0.0032)
Ref. [31]	99.5926(−0.0168)	33.5075(+0.0440)	–
Ref. [5]	99.6150(+0.0056)	33.4320(−0.0315)	–
Ref. [30]	99.5650(−0.0444)	33.4500(−0.0135)	–
Ref. [33]	99.6192(+0.0102)	33.5316(+0.0711)	–

The numbers in parentheses indicate the gaps from the corresponding standard values

pixels (NPCR), uniform average changing intensity (UACI) [24], and block average changing intensity (BACI). The utilized methods are given by the following equations.

$$NPCR(P,C) = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{D(i,j)}{M\times N}\times 100 \tag{14}$$

$$UACI(P,C) = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{|P(i,j)-C(i,j)|}{M\times N\times 255}\times 100 \tag{15}$$

$$BACI(P,C) = \frac{1}{(m-1)(n-1)}\sum_{i=1}^{m-1}\sum_{j=1}^{n-1}\frac{m_{ij}(P,C)}{255} \tag{16}$$

where

$$D(i,j) = \begin{cases} 0, P(i,j) = C(i,j) \\ 1, P(i,j) \neq C(i,j) \end{cases} \tag{17}$$

$$m_{i,j}(P,C) = \frac{1}{6}\sum\nolimits_{l=1}^{3}\sum\nolimits_{k=l+1}^{4}\left|d_{(i,j),l}-d_{(i,j),k}\right| \tag{18}$$

$$d_{(i,j),1} = P_{i,j}-C_{i,j}, d_{(i,j),2} = P_{i,j+1}-C_{i,j+1} \tag{19}$$

$$d_{(i,j),3} = P_{i+1,j}-C_{i+1,j}, d_{(i,j),4} = P_{i+1,j+1}-C_{i+1,j+1} \tag{20}$$

For $256 * 256$ grayscale images, the theoretical values of the above three indicators are 99.6094%, 33.4635%, and 26.7712%, respectively. This paper chooses a test picture, randomly selects a pixel value and XORs it with 1 to change its lowest bit, and then calculates the indicators after encryptions. Different test images are choosed, this process is repeated 150 times, and the simulation results are shown in Table 3. It can be seen from the table that the NPCR, UACI, and BACI indicators make small fluctuations around the theoretical

Table 5 Comparison of the encryption times (second) required by different encryption algorithms

Algorithm	Proposed	Ref. [16]	Ref. [32]	Ref. [2]	Ref. [12]
256*256	0.2571	0.2695	3.1342	0.4389	0.039
512*512	0.9126	1.1869	12.6917	1.8112	0.156

Table 6 Comparison of the ETs and numbers of cycles required by different encryption algorithms

Algorithm	ET (MBps)	number of cycles
Proposed	0.2431	11,794.32
Ref. [16]	0.2319	10,692.34
Ref. [32]	0.0473	50,405.62
Ref. [2]	0.1424	23,440.03
Ref. [12]	1.6025	1368.76

value. Table 4 shows the comparison between the indicators of different algorithms. The results demonstrate that this algorithm has a better performance than the other algorithms and has a strong resistance to differential attacks.

4.7 Encryption efficiency

We use the encryption time, encryption throughput (ET), and number of cycles per byte to measure encryption efficiency. The definitions of these metrics are shown in (21) and (22).

$$ET = \frac{image_{size}(byte)}{encryption_{time}(second)} \tag{21}$$

$$Number\ of\ cycles\ perbyte = \frac{CPU_{speed}(Hertz)}{ET(byte)} \tag{22}$$

An efficient image encryption algorithm, should have a short encryption time, large encryption throughput, and as few cycles as possible. The experimental environment of this article is MATLAB R2016a, with a PC containing an Inter(R) Core(TM) i7-7700HQ CPU @ 2.80 GHz and 16 GB of RAM on Windows 10. The method of calculating encryption time in this article is to average the time taken by the algorithm to encrypt the Lena image after proceeding 100 times. Table 5 shows the comparison between the encryption times of different algorithms. Table 6 shows the comparison between the ETs and cycle times of different algorithms. It can be seen from the table that our algorithm is in the upper-middle level regarding the speed of encryption, encryption throughput, and number of cycles under the premise of maintaining security and complexity.

Table 7 lists the time-consuming situations of the various stages in the encryption process. It can be seen from the table that the longest procedure is the forward shuffling and diffusion process, followed by the reverse diffusion process, and the least time-consuming process is the process of obtaining chaotic sequences.

Table 7 Time-consuming situations

Stages	Times (s)
Obtaining chaotic sequences	0.0238
Forward shuffling and diffusion process	0.1381
Reverse diffusion process	0.0952

5 Conclusions

In this paper, we design a novel hyperchaotic image encryption algorithm with simultaneous shuffling and diffusion. First, a hyperchaotic map combined with a two-dimensional Logistic map and the improved one-dimensional Feigenbaum transcendental map is proposed, and the simulation results show that it has a uniformly dispersed trajectory map, a wide hyperchaos window, a large Lyapunov exponent value, and a stable Permutation entropy. The system has excellent nonlinear characteristics and can provide the sequences required for the digital encryption process. Second, this paper presents a 256-bit hash-based secret key generation scheme, which is capable of generating initial values and parameters for the chaotic system that are highly sensitive to the input image content. Based on the above work, a new image encryption algorithm is given. The algorithm shuffles and diffuses the image simultaneously in one encryption operation, and each step is based on the full-text pixels of the image and the previous encryption result, which improves the encryption efficiency and increases the security level of the algorithm. It is proven that this algorithm needs only one round of encryption to meet the imposed encryption strength requirement. Simulation results and performance analyses show that the algorithm greatly improves upon the encryption effects, encryption speed and encryption security, and can resist various typical attacks.

Acknowledgements This research is supported by The National Key Research and Development Program of China (no.2018YFB1702902) and The National Natural Science Foundations of China under grants nos. 61763028 and 61862040. The authors gratefully acknowledge the anonymous reviewers for their helpful comments and suggestions.

Declarations

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

References

1. Bandt C, Pompe B (2002) Permutation entropy: A natural complexity measure for time series. Phys Rev Lett 88:174102. https://doi.org/10.1103/PhysRevLett.88.174102
2. Cai S, Huang L, Chen X, Xiong X (2018) A symmetric plaintext-related color image encryption system based on bit permutation. Entropy 20(4). https://doi.org/10.3390/e20040282
3. Chai X, Zheng X, Gan Z, Han D, Chen Y (2018) An image encryption algorithm based on chaotic system and compressive sensing. Sig Process 148:124–144. https://doi.org/10.1016/j.sigpro.2018.02.007
4. Chai X, Fu X, Gan Z, Lu Y, Chen Y (2019) A color image cryptosystem based on dynamic dna encryption and chaos. Sig Process 155:44–62. https://doi.org/10.1016/j.sigpro.2018.09.029
5. Chen J, Han F, Qian W, Yao YD, Zhu ZL (2018) Cryptanalysis and improvement in an image encryption scheme using combination of the 1d chaotic map. Nonlinear Dyn 93(4):2399–2413. https://doi.org/10.1007/s11071-018-4332-9
6. Enayatifar R, Abdullah AH, Isnin IF (2014) Chaos-based image encryption using a hybrid genetic algorithm and a dna sequence. Opt Lasers Eng 56:83–93. https://doi.org/10.1016/j.optlaseng.2013.12.003
7. Farwa S, Bibi N, Muhammad N (2020) An efficient image encryption scheme using fresnelet transform and elliptic curve based scrambling. Multimed Tools Appl 79(37):28225–28238. https://doi.org/10.1007/s11042-020-09324-4
8. Fei G, Shumao S (1992) Eigenvalues of non-linear equations and the feigenbaum formula. Comput Math Math Phys 32:403–405
9. Fridrich J (1998) Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurcation chaos 8(06):1259–1284. https://doi.org/10.1142/S0218127498000978

10. Hua Z, Zhou Y, Pun CM, Chen CP (2015) 2d sine logistic modulation map for image encryption. Inform Sci 297:80–94. https://doi.org/10.1016/j.ins.2014.11.018
11. Hua Z, Zhou Y (2016) Image encryption using 2d logistic-adjusted-sine map. Inform Sci 339:237–253. https://doi.org/10.1016/j.ins.2016.01.017
12. Huang X, Ye G (2014) An efficient self-adaptive model for chaotic image encryption algorithm. Commun Nonlinear Sci Numer Simul 19(12):4094–4104. https://doi.org/10.1016/j.cnsns.2014.04.012
13. Hussain I, Shah T, Gondal MA (2013) Application of s-box and chaotic map for image encryption. Math Comput Model 57(9):2576–2579. https://doi.org/10.1016/j.mcm.2013.01.009
14. Hussain I, Gondal MA (2014) An extended image encryption using chaotic coupled map and s-box transformation. Nonlinear Dyn 76(2):1355–1363. https://doi.org/10.1007/s11071-013-1214-z
15. Li Z, Peng C, Li L, Zhu X (2018) A novel plaintext-related image encryption scheme using hyper-chaotic system. Nonlinear Dyn 94(2):1319–1333. https://doi.org/10.1007/s11071-018-4426-4
16. Li S, Yin B, Ding W, Zhang T, Ma Y (2018) A nonlinearly modulated logistic map with delay for image encryption. Electronics 7(11). https://doi.org/10.3390/electronics7110326
17. Liu W, Sun K, Zhu C (2016) A fast image encryption algorithm based on chaotic map. Opt Lasers Eng 84:26–36. https://doi.org/10.1016/j.optlaseng.2016.03.019
18. Özkaynak F (2018) Brief review on application of nonlinear dynamics in image encryption. Nonlinear Dyn 92(2):305–313. https://doi.org/10.1007/s11071-018-4056-x
19. Pak C, Huang L (2017) A new color image encryption using combination of the 1d chaotic map. Sig Process 138:129–137. https://doi.org/10.1016/j.sigpro.2017.03.011
20. Rossler O (1979) An equation for hyperchaos. Phys Lett A 71(2):155–157. https://doi.org/10.1016/0375-9601(79)90150-6
21. Shevchenko I (2014) Lyapunov exponents in resonance multiplets. Phys Lett A 378(1):34–42. https://doi.org/10.1016/j.physleta.2013.10.035
22. Tu G, Liao X, Xiang T (2013) Cryptanalysis of a color image encryption algorithm based on chaos. Optik 124(22):5411–5415. https://doi.org/10.1016/j.ijleo.2013.03.113
23. Ur Rehman A, Liao X, Hahsmi MA, Haider R (2018) An efficient mixed inter-intra pixels substitution at 2bits-level for image encryption technique using dna and chaos. Optik - Int J Light Electron Opt 153:117–134. https://doi.org/10.1016/j.ijleo.2017.09.099
24. Wu Y, Noonan JP, Agaian S et al (2011) Npcr and uaci randomness tests for image encryption. Cyber journals: multidisciplinary journals in science and technology. J Sel Areas Telecommun (JSAT) 1(2):31–38
25. Wu Y, Noonan JP, Yang G, Jin H (2012) Image encryption using the two-dimensional logistic chaotic map. J Electron Imaging 21(1):013014. https://doi.org/10.1117/1.JEI.21.1.013014
26. Wu X, Zhu B, Hu Y, Ran Y (2017) A novel color image encryption scheme using rectangular transform-enhanced chaotic tent maps. IEEE Access 5:6429–6436. https://doi.org/10.1109/ACCESS.2017.2692043
27. Wang Y, Wong KW, Liao X, Xiang T (2009) A block cipher with dynamic s-boxes based on tent map. Commun Nonlinear Sci Numer Simul - Commun Nonlinear Sci Numer SI 14:3089–3099. https://doi.org/10.1016/j.cnsns.2008.12.005
28. Wang X, Teng L, Qin X (2012) A novel colour image encryption algorithm based on chaos. Sig Process 92(4):1101–1108. https://doi.org/10.1016/j.sigpro.2011.10.023
29. Wang XY, Gu SX, Zhang YQ (2015) Novel image encryption algorithm based on cycle shift and chaotic system. Opt Lasers Eng 68:126–134. https://doi.org/10.1016/j.optlaseng.2014.12.025
30. Wang X, Zhao H, Feng L, Ye X, Zhang H (2019) High-sensitivity image encryption algorithm with random diffusion based on dynamic-coupled map lattices. Opt Lasers Eng 122:225–238. https://doi.org/10.1016/j.optlaseng.2019.04.005
31. Wang X, Xue W, An J (2021) Image encryption algorithm based on ldcml and dna coding sequence. Multimed Tools Appl 80(1):591–614. https://doi.org/10.1007/s11042-020-09688-7
32. Xu L, Li Z, Li J, Hua W (2016) A novel bit-level image encryption algorithm based on chaotic maps. Opt Lasers Eng 78:17–25. https://doi.org/10.1016/j.optlaseng.2015.09.007
33. Xu L, Gou X, Li Z, Li J (2017) A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion. Opt Lasers Eng 91:41–52. https://doi.org/10.1016/j.optlaseng.2016.10.012
34. Ye G, Huang X (2016) A feedback chaotic image encryption scheme based on both bit-level and pixel-level. J Vib Control 22(5):1171–1180
35. Ye X, Wang X, Gao S, Mou J, Wang Z (2020) A new random diffusion algorithm based on the multi-scroll chua's chaotic circuit system. Opt Lasers Eng 127:105905. https://doi.org/10.1016/j.optlaseng.2019.105905

36. Zhang Y (2018) The unified image encryption algorithm based on chaos and cubic s-box. Inform Sci 450:361–377. https://doi.org/10.1016/j.ins.2018.03.055
37. Zhang Y, Tang Y (2018) A plaintext-related image encryption algorithm based on chaos. Multimed Tools Appl 77(6):6647–6669. https://doi.org/10.1007/s11042-017-4577-1