Multimedia Tools and Applications (2022) 81:40755–40773
https://doi.org/10.1007/s11042-022-12621-9

Image encryption using chaotic map and cellular automata

Lanhang Li[1] · Yuling Luo[1] · Senhui Qiu[1] · Xue Ouyang[1] · Lvchen Cao[1] · Shunbin Tang[1]

Received: 14 January 2021 / Revised: 11 May 2021 / Accepted: 9 February 2022 /
Published online: 13 May 2022

Abstract

Recent encryption schemes are not sensitive enough to plain-images, which leads to low robustness and easy vulnerability to attacks. By employing chaotic maps and Cellular Automata, a novel image encryption algorithm is proposed in this work to increase the sensitivity to plain-images and improve security. Firstly, initial values of the two-dimensional Logistic-Sine-Coupling Map and the Logistic-Sine-Cosine Map are calculated by the SHA-256 of the original image, and the process of diffusion is conducted. Secondly, the key matrices are produced by iterating chaotic maps in the process of permutation. The diffused image is scrambled by the index matrices, which are produced by sorting every row or column of the key matrices. Finally, the scrambled image is transformed into cipher-image by using Cellular Automata. Experimental results and theoretical analysis show that the proposed scheme has good security as it can effectively resist various attacks.

Keywords Image encryption · Cellular automata · SHA-256 · Chaotic maps

1 Introduction

Digital images may be leaked due to the non-secure channels and lead to a significant threat to information security when they are transmitted on the internet. Therefore, image encryption before transmitting the information to the receiver is an efficient way to protect individual privacy and social safety. The digital images can be converted into binary data and encrypted with Advanced Encryption Standard. However, the high correlation between adjacent pixels may still be kept after encrypting a digital image without considering the property of digital images. Chaotic maps own some special features, i.e., high sensitivity to the initial conditions, unpredictable, ergodicity, etc., which have been widely implemented on image encryption [18, 19, 21, 24, 25, 36]. Moreover, coupled chaotic map is a new system that can be acquired by conducting combinations using one-dimensional (1D) chaotic maps. Thus, the coupled chaotic map has more complex chaotic behaviour than its seed maps [40].

✉ Yuling Luo
yuling0616@gxnu.edu.cn

[1] School of Electronic Engineering, Guangxi Normal University, Guilin, China

In this algorithm, coupled chaotic maps are adopted to produce chaotic sequences with an excellent pseudo-randomness.

Cellular Automata (CA) is a typical discrete system [5]. In addition, a method of random sequence generation based on CA is proposed in [35], which is further used for image encryption [5, 10, 26]. There are mainly two CA-based image encryption methods, one is that CA can be used to produce pseudo-random numbers and the other is that CA is utilized to encrypt the image in bit-level. Specifically, in [23], a key matrix is built using pseudo-random numbers, which are generated by 1D CA. Random numbers in the key matrix are selected to encrypt the plain-image. In addition, an encryption algorithm using a second-order life-like CA is designed in [27], the image is transformed into binary matrix and is diffused by the life-like CA. In [28], reversible CA is applied to image encryption, and the pixels are confused by using CA and the histogram distribution of the encrypted image is more uniform because of the excellent pseudo-randomness of CA. Therefore, in this algorithm, CA is used to encrypt the scrambled image to reduce the correlation and further improve security.

Based on the above discussion, a novel image encryption scheme based on chaotic maps and CA is proposed. The following are the main contributions of this algorithm: (1) The new diffusion mechanism is designed and the confusion method of sorting is utilized by another form, which is different from others. The diffusion-confusion-CA transform architecture guarantees the nonlinear feature of the encryption method. (2) The plain image can get the hash value under the mapping of the hash function, and the initial value of the chaotic system can be obtained by using the designed formula. The generated chaotic sequences for encryption are sensitive to plain image, which guarantee the high sensitivity of the encryption algorithm. (3) In the proposed algorithm, 1D CA is used to encrypt the image in bit-level, and a new selection method of transform rules based on CA is presented. The sequence numbers based on CA are randomly generated by judging the interval of the values of the pseudo-random sequence. The selection of CA transform rules is more random.

The rest of this article is composed as follows. The fundamental knowledge about chaotic maps and CA are given in Section 2. Section 3 describes the proposed encryption algorithm. Simulation results and security performance analysis are discussed in Section 4. This paper is concluded in Section 5.

2 Preliminaries

2.1 Chaotic maps

The two-dimensional Logistic-Sine-Coupling Map (2D-LSCM) [12] is given by

$$\begin{cases} x_{i+1} = \sin\left(\pi\left(4\theta x_i\left(1-x_i\right)+\left(1-\theta\right)\sin\left(\pi y_i\right)\right)\right), \\ y_{i+1} = \sin\left(\pi\left(4\theta y_i\left(1-y_i\right)+\left(1-\theta\right)\sin\left(\pi x_{i+1}\right)\right)\right), \end{cases} \tag{1}$$

where $\theta \in [0, 1]$ is the control parameter. It has been demonstrated that the 2D-LSCM has chaotic behaviour when $\theta \in (0, 1)$.

In addition, the Logistic-Sine-Cosine Map (LSCM) [13] is defined by

$$x_{i+1} = \cos\left(\pi\left(4rx_i\left(1-x_i\right)+\left(1-r\right)\sin\left(\pi x_i\right)-0.5\right)\right), \tag{2}$$

where $r \in [0, 1]$. It has been proved that the LSCM has more complex chaotic behaviour than the Logistic map.

2.2 Cellular automata

In 1D CA, the two neighbours of each cell have two values, i.e., zero or one. Therefore, for three adjacent cells, there are $2 \times 2 \times 2 = 2^3$ possible states, which are 000, 001, 010, 011, 100, 101, 110, 111. The state function is given by

$$S_i^{t+1} = f(S_{i-1}^t, S_i^t, S_{i+1}^t), \tag{3}$$

where t denotes the time, S_i presents the current state of the i^{th} cell, and S_i^{t+1} denotes the next state of S_i at time $t + 1$.

The state of S_i^{t+1} is controlled by the i^{th} cell's state and the two neighbouring states of $i - 1^{th}$ cell and $i + 1^{th}$ cell at time t. $f(S_{i-1}^t, S_i^t, S_{i+1}^t)$ denotes Boolean function, which means logical operation. Table 1 shows different logical operation mode. Taking three adjacent cells as an operational unit at time t, and there are eight possible states of an operational unit. The eight states of the i^{th} at time $t + 1$ are obtained by using a kind of Boolean function. Then, eight binary numbers are converted into a decimal number and the decimal number is named as the rule number corresponding to this kind of Boolean function. For example, when rule 90 is selected to conduct the logical operation. The next state of S_i can be shown as $f(000) = 0$, $f(001) = 1$, $f(010) = 0$, $f(011) = 1$, $f(100) = 1$, $f(101) = 0$, $f(110) = 1$, $f(111) = 0$. Number 90 represents the decimal number of the S_i , that is $(01011010)_2$=$(90)_{10}$.

3 The proposed encryption algorithm

3.1 Generating the initial values of chaotic maps

Hash function is used to transform the input with any length into a fixed-length output, which can be applied to authenticated encryption [37]. In this algorithm, the initial secret keys are obtained by employing the SHA-256 hash value of the original image, which is calculated as H. H represents 64 hexadecimal values since each hexadecimal denotes four binary values. Then, H is converted to decimal values and further divided into eight blocks. Each block includes eight values, which can be given by

$$H = d_1, d_2, ..., d_7, d_8. \tag{4}$$

Then, the intermediate values $\alpha = d_1 \oplus d_2 \oplus d_3 \oplus d_4$, and $\beta = d_5 \oplus d_6 \oplus d_7 \oplus d_8$.

Table 1 Different rules of 1D CA

S_i	Rule number	Boolean function
1	30	$S_i^{t+1} = S_{i-1}^t \oplus \left\lvert S_i^t + S_{i+1}^t \right\rvert$
2	90	$S_i^{t+1} = S_{i-1}^t \oplus S_{i+1}^t$
3	150	$S_i^{t+1} = S_{i-1}^t \oplus S_i^t \oplus S_{i+1}^t$
4	153	$S_i^{t+1} = S_i^t \odot S_{i+1}^t$
5	165	$S_i^{t+1} = S_{i-1}^t \odot S_{i+1}^t$
6	86	$S_i^{t+1} = \overline{S_{i-1}^t + S_i^t} \oplus \overline{S_i^t}$
7	105	$S_i^{t+1} = \overline{S_{i-1}^t \oplus S_i^t \oplus S_{i+1}^t}$
8	101	$S_i^{t+1} = S_{i-1}^t \odot S_{i+1}^t + \left(S_i^t \oplus S_{i+1}^t\right) \cdot S_{i-1}^t$

Finally, the initial values are given by

$$\begin{cases} x_1 = \alpha(1) \oplus \alpha(2) \oplus \alpha(3) \oplus \alpha(4), \\ y_1 = \beta(1) \oplus \beta(2) \oplus \beta(3) \oplus \beta(4), \\ z_1 = \frac{x_1+y_1}{2}. \end{cases} \tag{5}$$

3.2 The process of encryption

The flow chat of the proposed method is displayed in Fig. 1, and the detailed encryption steps are presented next.

Step 1. Suppose the original image is I, and the dimension of it is $m \times n$. Initial values of two chaotic maps can be acquired according to H, which is represented in Section 3.1.

Step 2. The 2D-LSCM is iterated for $m \times n + 500$ times by using x_1, y_1, and two sequences X, Y are obtained by discarding the former 500 values. Similarly, the chaotic sequence Z is generated by iterating the LSCM for $m \times n \times 8$ times using z_1. Then, the sequence X is reshaped into matrix X_1 with the same size as I. In addition, $2m$ values are selected from sequence Z, and quantification operations are performed by

$$\begin{cases} Y_1 = \text{mod}\ \left(\text{floor}\left(Y \times 10^{14}\right), 256\right), \\ R = \text{mod}\ \left(\text{floor}\left(Z(1:n) \times 10^{14}\right), 256\right), \\ C = \text{mod}\ \left(\text{floor}\left(Z(m+1:2m) \times 10^{14}\right), 256\right), \end{cases} \tag{6}$$

where $Z(1:n)$ represents the first value to the n^{th} value in sequence Z, $Z(m+1:2m)$ represents the $(m+1)^{th}$ value to the $2m^{th}$ value in sequence Z. The diffusion matrix D is obtained by reshaping vector Y_1 into matrix.

Step 3. The plain-image is diffused by using the new diffusion method. Firstly, the pixels in the first row of plain-image are encrypted by vector R and the first row of matrix D, which is calculated by

$$C_1(1, x) = \text{mod}\ (R(x) + I(1, x), 256) \oplus D(1, x), \tag{7}$$

where $x \in [2, n]$, R is 1D vector produced by the LSCM.

Secondly, the pixels in the first column of plain-image are encrypted by vector C and the first column of matrix D, which is calculated by

$$C_1(y, 1) = \text{mod}\ (C(y) + I(y, 1), 256) \oplus D(y, 1), \tag{8}$$

where $y \in [2, m]$, C is 1D vector produced by the LSCM.

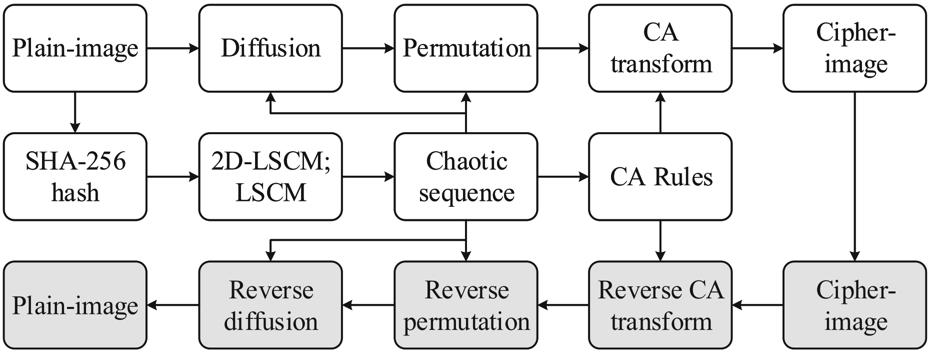


Fig. 1 Flow chat of the proposed encryption algorithm

Finally, when $x \in [2, m]$, $y \in [2, n]$, other pixels are encrypted by the corresponding values in matrix D, which is shown in

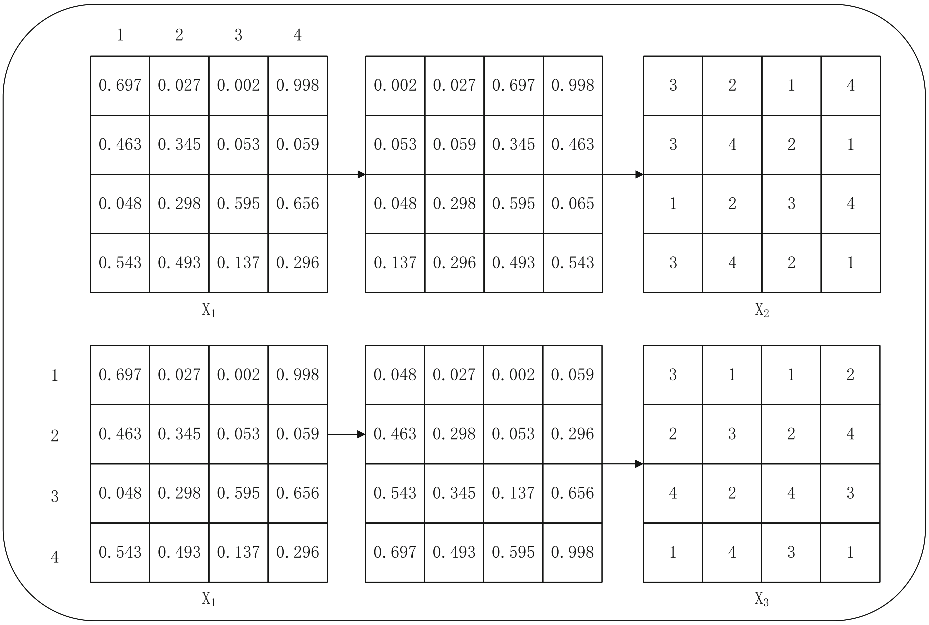
$$C_1(x, y) = \bmod\ (C_1(x-1, y-1) + I(x, y), 256) \oplus D(x, y). \quad (9)$$

Specially, the first value $I(1, 1)$ is encrypted by

$$C_1(1, 1) = \bmod\ (I(1, 1) + R(1), 256) \oplus D(1, 1). \quad (10)$$

Step 4. The process of permutation is presented. The values in every row and column of the key matrix X_1 are sorted in ascending order. Figure 2 shows the process of generating the index matrices for permutation when the size of the image is 4 × 4. In detail, the first row in X_1 is 0.697,0.027,0.002,0.998. After sorting in ascending, this row changed to 0.002,0.027,0.697,0.998. The original sequence 1,2,3,4 is changed to new sequence 3,2,1,4, which is shown in the first row of X_2. As a result, two index matrices X_2 and X_3 are obtained. In this algorithm, the permutated image C_2 can be acquired by using the index matrices to confuse the diffused image.

Step 5. CA transform is adopted to obtain the final cipher-image. Eight CA transform rules in Table I are randomly selected to encrypt the image in this paper. The selection of the rule number is controlled by the sequence numbers R_i. For example, if R_i is five, the Rule 165 CA is selected to transform the pixel in bit-level. R_i can be obtained by



1	2	3	4
0.697	0.027	0.002	0.998
0.463	0.345	0.053	0.059
0.048	0.298	0.595	0.656
0.543	0.493	0.137	0.296

X1

0.002	0.027	0.697	0.998
0.053	0.059	0.345	0.463
0.048	0.298	0.595	0.065
0.137	0.296	0.493	0.543

3	2	1	4
3	4	2	1
1	2	3	4
3	4	2	1

X2

1	0.697	0.027	0.002	0.998
2	0.463	0.345	0.053	0.059
3	0.048	0.298	0.595	0.656
4	0.543	0.493	0.137	0.296

X1

0.048	0.027	0.002	0.059
0.463	0.298	0.053	0.296
0.543	0.345	0.137	0.656
0.697	0.493	0.595	0.998

3	1	1	2
2	3	2	4
4	2	4	3
1	4	3	1

X3

Fig. 2 The process of generating index matrices with the size of 4×4

judging the interval of the values of chaotic sequences Z produced by the LSCM. R_i is given by

$$R_i = \begin{cases} 1, 0 \leq Z(i) \leq 0.125, \\ 2, 0.125 < Z(i) \leq 0.25, \\ 3, 0.25 < Z(i) \leq 0.375, \\ 4, 0.375 < Z(i) \leq 0.5, \\ 5, 0.5 < Z(i) \leq 0.625, \\ 6, 0.625 < Z(i) \leq 0.75, \\ 7, 0.75 < Z(i) \leq 0.875, \\ 8, 0.875 < Z(i) \leq 1, \end{cases} \tag{11}$$

where i denotes the iterating time, and $i \in [1, m \times n \times 8]$. R_i denotes the sequence numbers, and $R_i \in [1, 8]$.

What's more, the Rule numbers in the process of CA transform are randomly selected. The chaotic sequence Z is obtained by iterating the LSCM and the initial value of the LSCM is related with plain-image, which certifies that the process of CA transform is sensitive to plain-image.

Step 6. The permuted matrix C_2 is reshaped into a vector, and the vector is transformed into sequence C_3. The value in C_3 is binary, and the length of C_3 is $m \times n \times 8$. Then, C_3 is regarded as the input of CA, and the corresponding rules of CA transform are values in the sequence Z_i. The output can be obtained after conducting CA transform, and the output is converted into decimal vector C_4. Finally, the vector C_4 is reshaped into the final encrypted image.

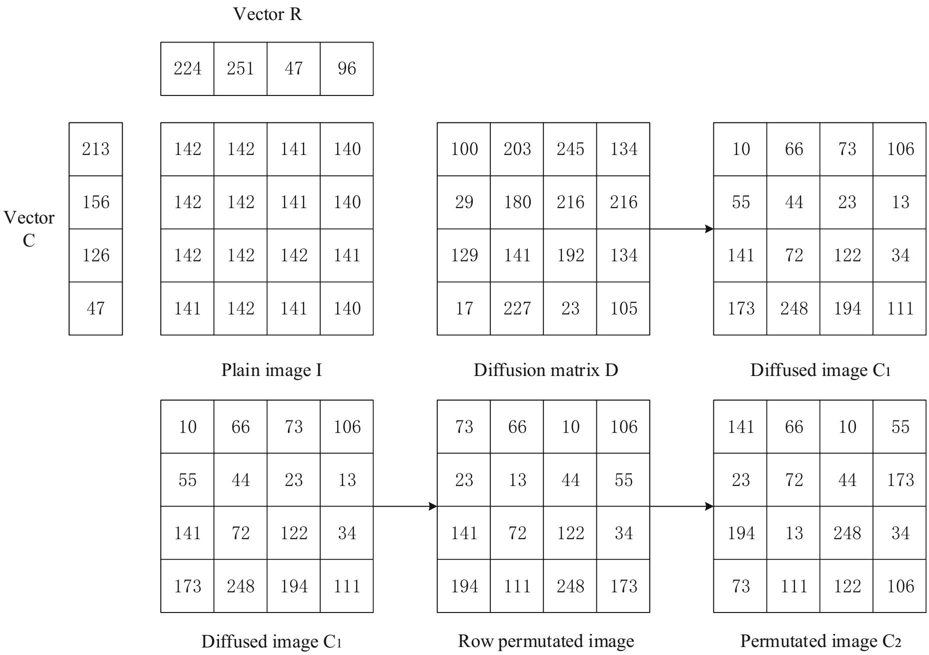
For better understanding the process of encryption. Figures 3 and 4 are regarded as an example to explain the process of encrypting an image with the size of 4 × 4. Firstly, the vector C, R, and diffusion matrix D can be obtained according to the description in Step 2. The first row in Fig. 3 shows the result of diffusion, which is calculated according to Step 3. The rows of diffused image C_1 are reordered by using the index matrix X_2, and then the columns are reordered by using the index matrix X_3. In Fig. 4, the permutated image C_2 is transformed into sequence. The first pixel in the sequence is regarded as an example to conduct CA transform. The decimal number 141 is blue, and the binary form of it is 10001100. From Table 1, it can be seen that the corresponding rule numbers represented by sequence R(1,8,7,8,3,6,4,1) are 30,101,105,101,150,86,153,30. According to the Boolean function, the transform result can be obtained, that is 10100111. So, 141 is converted into 167. Decimal number 167 is remarked in yellow. Similarly, other pixels are conducted the same process. CA transform result is obtained, and it is reshaped into the final cipher image with the size of 4 × 4.

3.3 The process of decryption

Obviously, the decryption process is the inverse of encryption. The detail is presented as follows.

Step1: The chaotic sequences X, Y, and Z are generated by iterating the chaotic systems using secret keys. Then the quantification operations are conducted by using (6). Sequences Y_1, R, and C are obtained. X is reshaped into a matrix X_1, which is named as the key matrix. And Y_1 is reshaped into diffusion matrix D. According to Z and (11), the Rule sequence R_i is obtained.

Step2: The cipher image is transformed into binary sequence C_4. Then, it is regarded as the input of CA. The output can be obtained after conducting CA transform by using



Vector R

224	251	47	96

Vector C

213
156
126
47

Plain image I

142	142	141	140
142	142	141	140
142	142	142	141
141	142	141	140

Diffusion matrix D

100	203	245	134
29	180	216	216
129	141	192	134
17	227	23	105

Diffused image C1

10	66	73	106
55	44	23	13
141	72	122	34
173	248	194	111

Diffused image C1

10	66	73	106
55	44	23	13
141	72	122	34
173	248	194	111

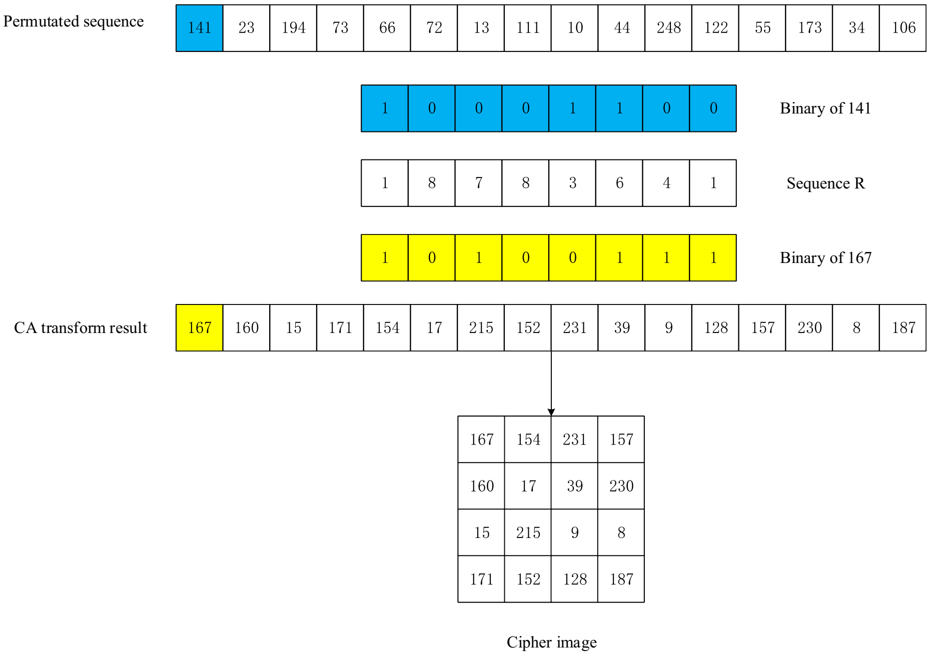
Row permutated image

73	66	10	106
23	13	44	55
141	72	122	34
194	111	248	173

Permutated image C2

141	66	10	55
23	72	44	173
194	13	248	34
73	111	122	106

Fig. 3 The process of diffusion and permutation



Permutated sequence

141	23	194	73	66	72	13	111	10	44	248	122	55	173	34	106

Binary of 141

1	0	0	0	1	1	0	0

Sequence R

1	8	7	8	3	6	4	1

Binary of 167

1	0	1	0	0	1	1	1

CA transform result

167	160	15	171	154	17	215	152	231	39	9	128	157	230	8	187

Cipher image

167	154	231	157
160	17	39	230
15	215	9	8
171	152	128	187

Fig. 4 The process of CA transform

Rule sequence R_i. The output is converted into decimal vector C_3, and further reshaped into matrix C_2.

Step3: Two index matrices X_2, and X_3 are obtained by using X_1 according to the description of Step 4 in Section 3.2. The elements of the column in matrix C_2 are recovered according to the position of index matrix X_2. Further, the reverse permutation is finished by using the index matrix X_3. The result is named as C_1.

Step4: Firstly, the pixel values other than the first row and the first column are decrypted by

$$I(x, y) = \bmod\ (C_1(x, y) \oplus D(x, y) - C_1(x-1, y-1), 256)\,, \tag{12}$$

where $x \in [m, 2]$, and $y \in [n, 2]$.

Secondly, the pixels of first row excluded the first position are decrypted by

$$I(1, x) = \bmod\ (C_1(1, x) \oplus D(1, x) - R(x), 256)\,, \tag{13}$$

where $x \in [n, 2]$. Similarly, the pixels of first column excluded the first position are decrypted by

$$I(y, 1) = \bmod\ (C_1(y, 1) \oplus D(y, 1) - C(y), 256)\,, \tag{14}$$

where $y \in [m, 2]$. Finally, the last decrypted pixel is obtained by

$$I(1, 1) = \bmod\ (C_1(1, 1) \oplus D(1, 1) - R(1), 256)\,. \tag{15}$$

Then, the decrypted image I is recovered.

4 Experimental results and security analysis

Experimental results are displayed in this part and security analysis is discussed in detail. Three standard images are tested, and the size of them is 512 × 512. No useful information can be acquired from encrypted images, which are represented in Fig. 5, and the plain-images can be successfully recovered.

4.1 Differential attack

After a tiny modification is made for plain-image, hackers can obtain one new image and encrypt the modified image. Then, attackers may compare the difference between two encrypted images to seek out secret keys. Generally, the number of pixel change rate ($NPCR$) and unified average changing intensity ($UACI$) are selected to assess the capacity for withstanding the differential attack, which are defined by [4]

$$NPCR = \frac{\sum_{i,j} D(i, j)}{m \times n} \times 100\%, \tag{16}$$

$$UACI = \frac{1}{m \times n}\left[\sum_{i,j} \frac{|T_1(i, j) - T_2(i, j)|}{255}\right] \times 100\%, \tag{17}$$

where $m \times n$ is the dimension of image. T_1 and T_2 are encrypted images, and their corresponding plain-images differ by one pixel. When $T_1(i, j) \neq T_2(i, j)$, the difference matrix $D(i, j) = 1$; or else, $D(i, j) = 0$.

In this test, three images are tested, and 100 pixels are selected randomly with adding one for each time. The test results are listed in Table 2. The results indicate that the average values are approximate to the expected value [11], which certifies that the algorithm can resist differential attack.

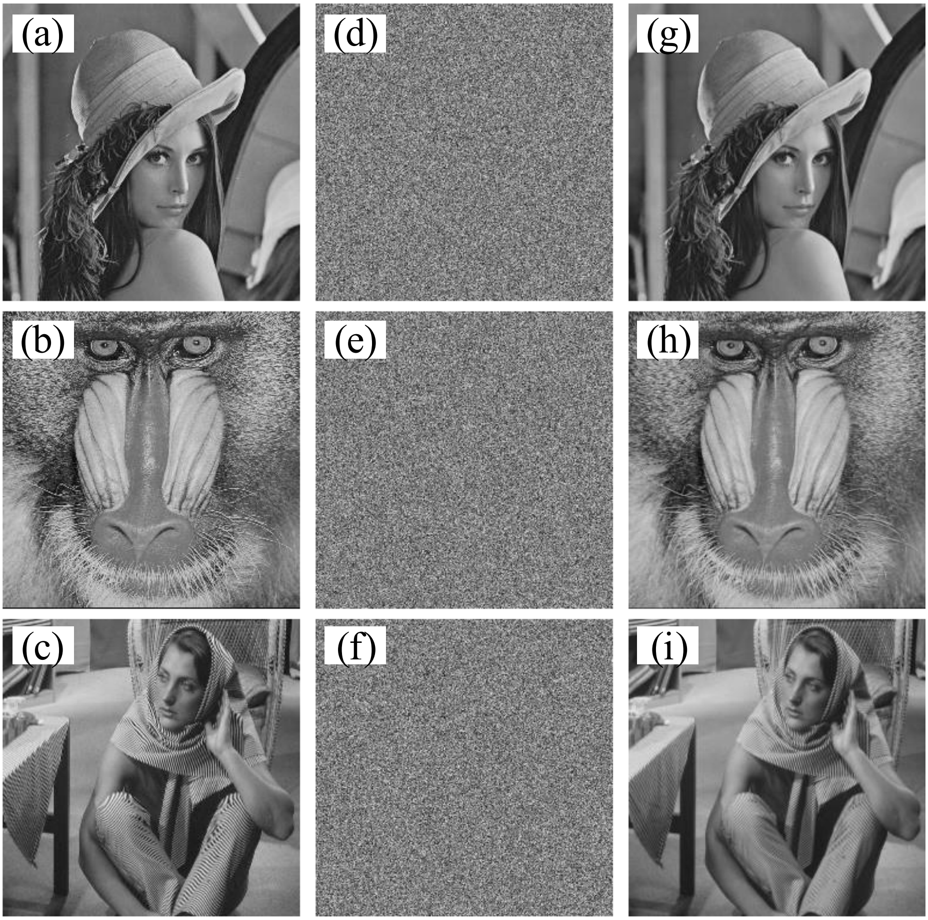


Fig. 5 Experimental results. **a-c** original images; **d-f** encrypted images; **g-i** decrypted images

4.2 Key space

Key space ought to be large enough to withstand brute force attack, and the key space in an encryption scheme is usually required to reach 2^{100} [2]. The secret keys in this paper includes two control parameters r, θ, initial values x_1, y_1, z_1. The accuracy of the computer is limited, assuming it is 10^{-15} [14], the whole key space in this work is 2^{260}. Therefore, the proposed algorithm can defend against brute-force attack.

Table 2 *NPCR* and *UACI* of different images

Image	*NPCR*(%)	*UACI*(%)
Lena	99.6103	33.4540
Baboon	99.6105	33.4643
Barbara	99.6093	33.4792

4.3 Key sensitivity

Encryption method ought to be sensitive enough to the encryption key, which means that an enormous variation will be taken place in cipher-image when the secret key is altered slightly. To prove the key sensitivity, the initial parameter θ is changed slightly to $\theta' = \theta + 10^{-16}$.

Figure 6c shows the new cipher-image when θ' is utilized to encrypt the 512×512 "Lena". In addition, Fig. 6d-g show the other new encrypted images when other keys are modified. Figure 6h-l indicate that there is an enormous difference between Fig. 6c-g and b.

Figure 7a shows the decrypted image using correct keys, and Fig. 7b-f display wrong decrypted images by using modified keys. The results show that the plain-image can't be recovered correctly using modified keys even the keys are changed slightly.

What's more, the decrypted image by using an error key has an enormous difference from the correct decrypted image. Usually, this difference can be evaluated by the peak signal-to-noise ratio ($PSNR$) and mean square error (MSE) [6], which are calculated by [8]

$$PSNR = 10\log_{10}\frac{255^2}{MSE}, \tag{18}$$

$$MSE = \frac{1}{m \times n}\sum_{x=1}^{m}\sum_{y=1}^{n}[D(x, y) - P(x, y)]^2, \tag{19}$$

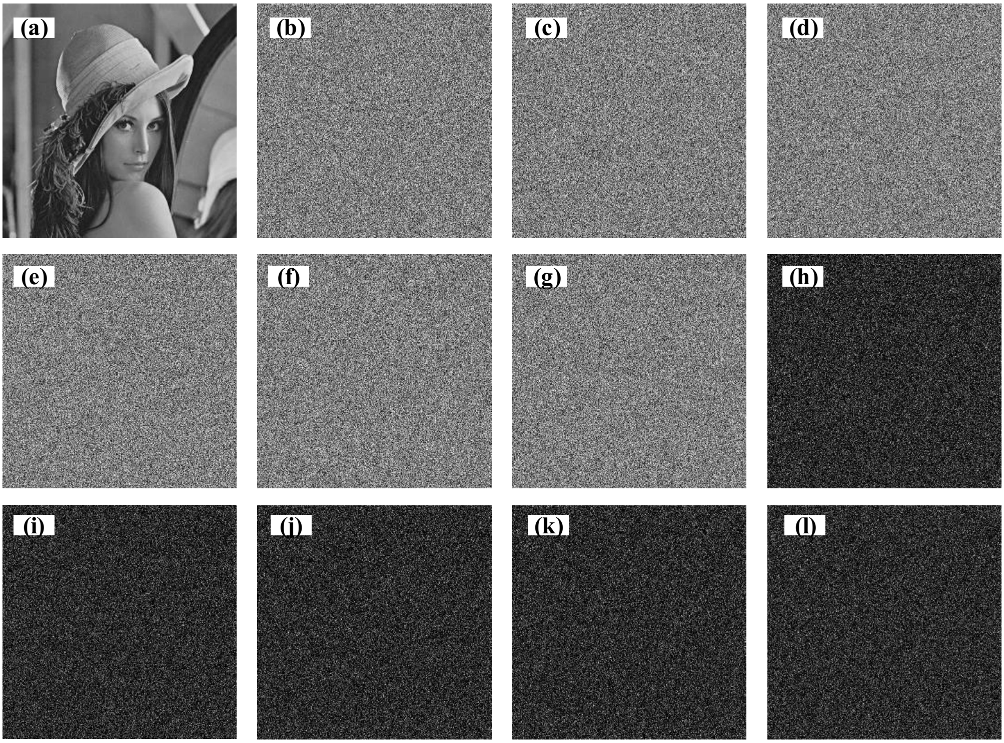


Fig. 6 **a** plain-image of "Lena"; **b** cipher-image with original keys; **c-g** cipher-images with modified keys; **h-l** difference images between **b** and **c-g**

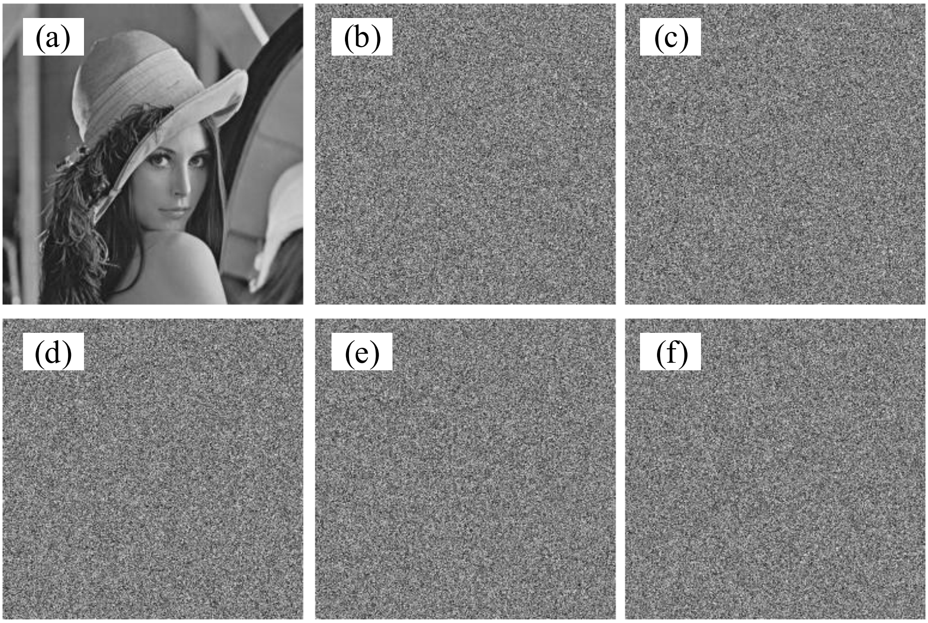


Fig. 7 **a** decrypted image using original keys; **b**-**f** decrypted image using modified keys

where D represents decrypted image by using a wrong key and P is the original image. A small value of $PSNR$ demonstrates that there is a great difference between image P and D [15]. The key sensitivity can be also evaluated by $NPCR$ and $UACI$. The results are listed in Table 3, which certifies that the proposed algorithm has good performance in key sensitivity.

4.4 The histogram analysis

Histogram of an image shows each pixel value's distribution [38]. If the histogram is not uniform, the exposed information may be utilized by attackers. Figure 8 displays the histograms of encrypted images, which are balanced distribution. Hence, the proposed algorithm is resistant to statistical attack.

Table 3 Key sensitivity about decrypted image

The wrong keys	*NPCR*(%)	*UACI*(%)	*MSE*	*PSNR*
$\theta' = \theta + 10^{-16}$	99.6151	29.1550	8111.2823	9.0399
$x_1' = x_1 + 10^{-16}$	99.6147	29.2031	8142.8512	9.0230
$y_1' = y_1 + 10^{-16}$	99.5853	29.1941	8132.8688	9.0284
$r' = r + 10^{-16}$	99.5865	29.0433	8082.0390	9.0556
$z_1' = z_1 + 10^{-16}$	99.5983	29.1938	8141.5808	9.0237

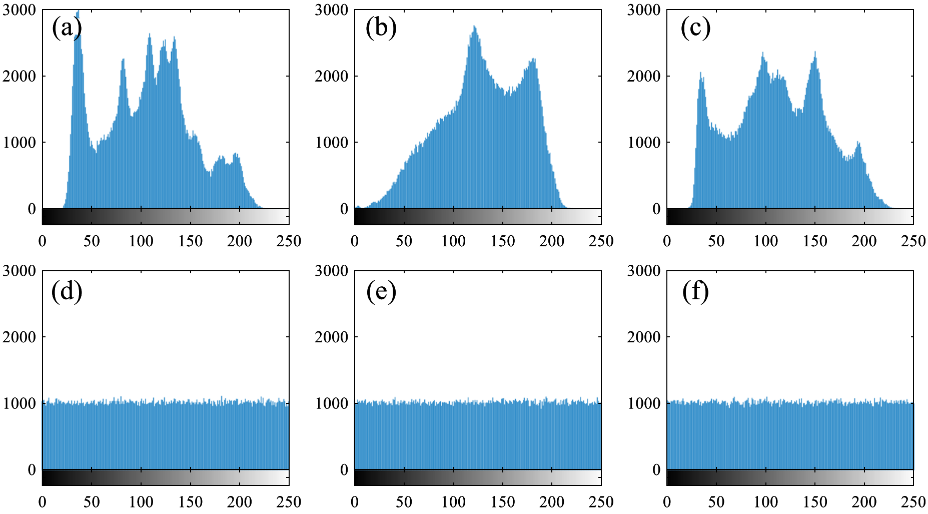


Fig. 8 **a-c** Histograms of “Lena”, “Baboon”, and “Barbara”. **d-f** Histograms of their cipher-images

4.5 Correlation analysis

The strong correlation in horizontal(HL), vertical(VL), and diagonal(DL) directions exist commonly between two adjacent pixels, which indicates that the original image contains a lot of redundant information. Hence, a secure encryption algorithm ought to eliminate these

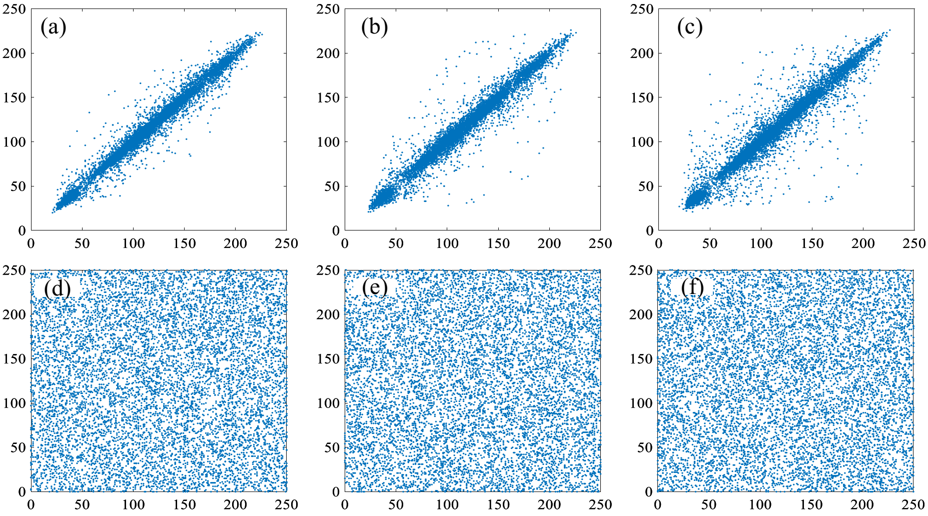


Fig. 9 Correlation distribution. **a-c** correlation distributions in three directions of original “Lena”; **d-f** correlation distributions in in three directions of encrypted “Lena”

Table 4 Correlation coefficient of original and encrypted images

Image	Original			Encrypted		
	HL	VL	DL	HL	VL	DL
Lena	0.9850	0.9782	0.9633	0.0065	0.0051	−0.0005
Baboon	0.7115	0.8511	0.6839	−0.0038	0.0037	0.0005
Barbara	0.9688	0.8933	0.8568	0.0021	−0.0020	0.0025

correlations. The correlation coefficients cor_{xy} can be calculated by [7]

$$\mathrm{cor}_{xy} = \frac{\sum_{i=1}^{N}\left(x_i-\omega_x\right)\left(y_i-\omega_y\right)}{\sqrt{\sum_{i=1}^{N}\left(x_i-\omega_x\right)^2\left(y_i-\omega_y\right)^2}}, \tag{20}$$

where $\omega_x = \sum_{i=1}^{N} x_i$, $\omega_y = \sum_{i=1}^{N} y_i$, x_i and y_i represent gray values of adjacent pixels in an image. Figure 9 displays the correlation of adjacent pixels in three directions of “Lena” and its encrypted image, respectively. In addition, Table 4 gives the correlation coefficients of test images. The correlation coefficients of encrypted images in this paper are near to zero. So the proposed algorithm is able to eliminate the correlation effectively.

4.6 Information entropy analysis

The randomness of information can be described by information entropy, which can be calculated by [7]

$$H(s) = \sum_{i=0}^{2^n-1} p\left(s_i\right)\log_2\frac{1}{p\left(s_i\right)}, \tag{21}$$

where $p(s_i)$ is the probability of s_i. In a fully uniform image with 2^8 gray level, the expected value of entropy is eight [23]. The nearer the entropy is to eight of cipher-image, the higher security of the scheme [32]. In the proposed scheme, “Lena”, “Baboon”, “Barbara”, and the corresponding cipher-images are tested. The results for three tested images are listed in Table 5. The information entropies of the encrypted image in this paper are close to eight, which prove that the proposed scheme has good randomness.

4.7 Data loss and noise attack analysis

The information may be lost due to the effect of congestion network or noise when the cipher-image is transmitted over internet [20]. Therefore, it is essential for the cryptosystem to have the ability to resist data loss and noise attacks [9].

Table 5 Information entropies of different images and the comparison results

Image	Original	Encrypted
Lena	7.38712	7.99927
Baboon	7.35795	7.99933
Barbara	7.46642	7.99934

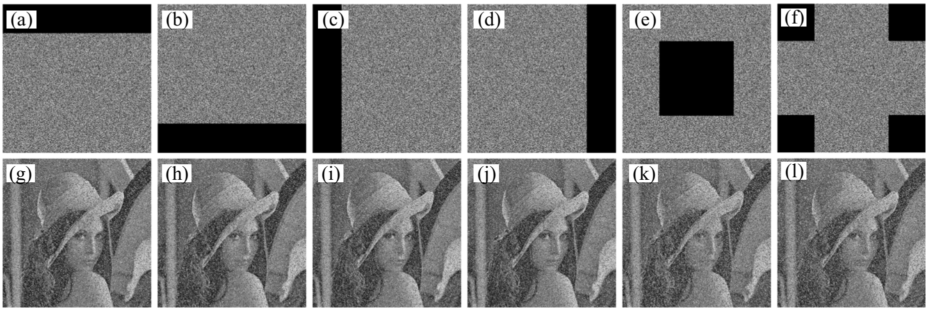


Fig. 10 Data loss attack. **a-d** cipher image with 19.5% data loss, **e-f** cipher image with 25% data loss; **g-l** corresponding decrypted images

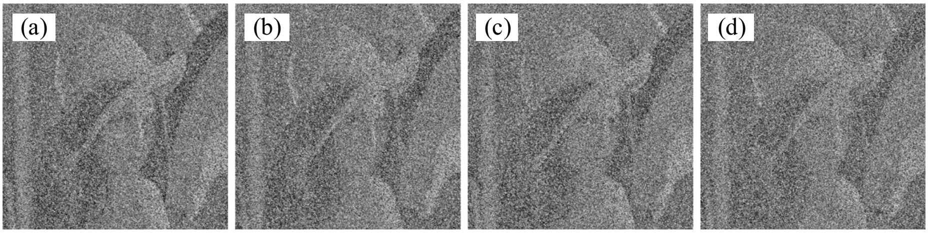


Fig. 11 Decrypted images under different GN intensities: **a-d**: *k*=5,10,20,30

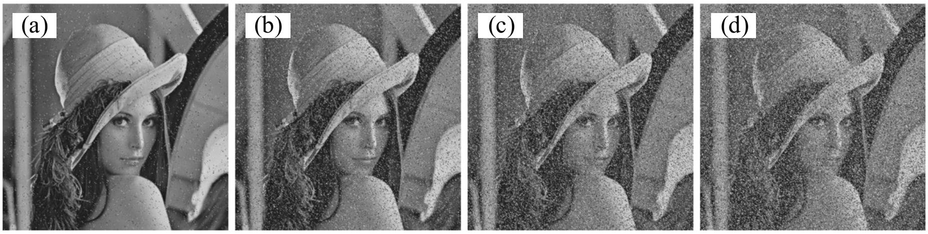


Fig. 12 Decrypted images under different SPN densities **a**: 1%; **b**: 5%; **c**: 10%; **d**: 15%

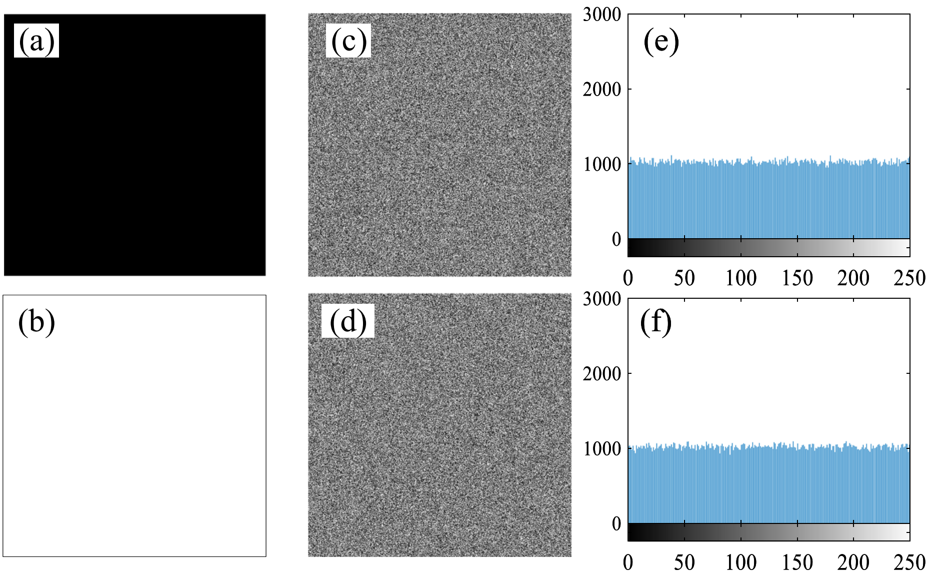


Fig. 13 **a-b** images of all black and white, **c-d** encrypted images, **e-f** histograms of **c-d**

Table 6 Performance evaluation results of special images

Image	Correlation coefficients			Entropy
	Horizontal	Diagonal	Vertical	
All black	0.0028	0.0012	−0.0033	7.99928
All white	−0.0030	0.0031	−0.0027	7.99929

Table 7 Encryption speed analysis

Process	Diffusion	Permutation	CA transform	Other	Total
Time(s)	0.369265	0.009421	0.191673	0.783367	1.353726

Table 8 Comparison result with other schemes

Method	Image	*NPCR*(%)	*UACI*(%)	Entropy	Correlation coefficients			Time(s)
					HL	VL	DL	
This work	Lena(256 × 256)	99.6108	33.4922	7.9970	0.0026	0.0064	−0.0086	0.3572
	Lena(512 × 512)	99.6103	33.4540	7.9993	0.0065	0.0051	−0.0005	1.3537
[33]	Lena(256 × 256)	99.5956	33.5512	7.9975	−0.0037	−0.0029	0.0047	1.172
[29]	Lena(256 × 256)	99.6105	33.4631	7.9972	0.0068	−0.0224	−0.0259	1.05
[30]	Lena(256 × 256)	99.6002	33.4592	7.9976	−0.0017	−0.0004	0.0028	7.14
[21]	Lena(512 × 512)	99.6113	33.4682	7.9993	0.0019	−0.0024	0.0011	4.7339
[39]	Lena(512 × 512)	99.6114	33.4523	7.9992	−0.0381	−0.0291	0.0027	–
[22]	Lena(512 × 512)	99.6881	37.5600	7.9993	0.0015	0.0043	0.0023	3.007
[1]	Lena(512 × 512)	99.6129	33.4127	7.9993	0.0013	0.0060	−0.0084	–
[31]	Lena(512 × 512)	99.6074	33.4792	7.9993	−0.0014	0.0012	0.0010	–
[3]	Lena(512 × 512)	99.6258	33.4153	7.9993	0.0014	0.0020	0.0012	0.964
[34]	Lena(512 × 512)	99.6043	33.4642	7.9992	0.0033	0.0022	−0.0035	-

The cipher-image “Lena” with different cropped parts and the decrypted results are displayed in Fig. 10. The recovered images can still be identified in spite of there are a large number of data is lost in cipher-images. Hence, the proposed algorithm can resist data loss attack effectively.

In addition, the Gaussian Noise(GN) is tested, which can be added to the encrypted image C by [17]

$$C' = C + kG_N, \tag{22}$$

where C' is a new encrypted image after adding GN, k is noise intensity, and G_N is the standard GN. The decrypted results after adding GN are represented in Fig. 11, in which the recovered images can be recognized when $k = 30$. Therefore, the proposed algorithm is resistant to GN attack.

What’s more, the decrypted image after adding Salt and Pepper Noise(SPN) with different densities is given in Fig. 12. The results indicate that the encryption scheme is capable of resisting SPN attack.

4.8 Known/chosen plaintext attack analysis

Initial values of the 2D-LSCM and the LSCM in the proposed algorithm are calculated by SHA-256 hash value of plain-image, which indicates that the encrypted image is highly sensitive to the plain-image. What’s more, attackers usually choose a special image such as all black and the secret key may be found according to the chosen-plaintext attack [16]. Figure 13 shows the cipher-images of all black and white, and their histograms. In addition, Table 6 shows the entropies and correlation coefficients of cipher-images. The results indicate that the attackers cannot get any valuable information from the encrypted images. Hence, the proposed scheme can withstand the known/chosen plaintext attack.

4.9 Encryption speed analysis and comparison

The encryption efficiency of the algorithm is an important index to measure the encryption performance of the cryptographic system. The experimental environment is based on MATLAB R2018b with Intel Core i3-8100 CPU @ 3.6GHz, 8G RAM. The operating system is Windows 10. “Lena” image with the size of 512 × 512 is encrypted, and the time of each process and the total encryption time are listed in Table 7.

In addition, recent methods are compared with the proposed algorithm, which is listed in Table 8. In Table 8, the entropy in the proposed method is lower than [29, 30, 33]. However, the encryption efficiency of this scheme is much faster than [29, 30, 33] when the size of test image is 256 × 256. Further, the encryption speed of proposed algorithm is slower than [3], but it is faster than [21] and [22]. Finally, the proposed scheme has similar performance compared with other algorithms in other respects, including $NPCR$, $UACI$, entropy, and correlation coefficients. Therefore, the proposed algorithm has good safety and effectiveness.

5 Conclusion

Based on chaotic maps and CA, a novel image encryption algorithm is proposed in this paper. Firstly, initial secret keys of the chaotic maps are calculated by using the SHA-256 hash value of the original image, which guarantees high key sensitivity. Then, the plain-image is diffused and scrambled. The final encrypted image is obtained by transforming the scrambled image using CA. Experimental results and security analysis demonstrate the proposed scheme has good performance in key sensitivity, information entropy, and it can resist attacks, i.e., noise and data loss attacks. In the future, the robustness of resisting noise attack will be increased, and the effectiveness of this algorithm can be improved further.

Acknowledgements This research was supported by the National Natural Science Foundation of China under Grant 61801131, the funding of Overseas 100 Talents Program of Guangxi Higher Education, 2018 Guangxi One Thousand Young and Middle-Aged College and University Backbone Teachers Cultivation Program.

References

1. Abbasi AA, Mazinani M, Hosseini R (2020) Evolutionary-based image encryption using biomolecules operators and non-coupled map lattice. Optik 219(1):164949
2. Alvarez G, Li S (2006) Some basic cryptographic requirements for Chaos-Based cryptosystems. Int J Bifurc Chaos 16(8):2129–2151
3. Babaei A, Motameni H, Enayatifar R (2020) A new permutation-diffusion-based image encryption technique using cellular automata and DNA sequence. Optik 203(1):164000
4. Chai X, Chen Y, Broyde L (2017a) A novel chaos-based image encryption algorithm using DNA sequence operations. Opt Lasers Eng 88(1):197–213
5. Chai X, Gan Z, Yang K, Chen Y, Liu X (2017b) An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations. Signal Process Image Commun 52(1):6–19
6. Chai X, Gan Z, Yuan K, Lu Y, Chen Y (2017c) An image encryption scheme based on three-dimensional Brownian motion and chaotic system. Chin Phys B 26(2):1674–1056
7. Chai X, Bi J, Gan Z, Liu X, Zhang Y, Chen Y (2020) Color image compression and encryption scheme based on compressive sensing and double random encryption strategy. Signal Process 176(1):107684
8. Chai X, Wu H, Gan Z, Han D, Zhang Y, Chen Y (2021) An efficient approach for encrypting double color images into a visually meaningful cipher image using 2D compressive sensing. Inf Sci 556(1):305–340
9. Chen J, Zhu Z, Fu C, Zhang L, Yu H (2015) Analysis and improvement of a double-image encryption scheme using pixel scrambling technique in gyrator domains. Opt Lasers Eng 66(1):1–9
10. Dennunzio A, Formenti E, Manzoni L, Margara L, Porreca AE (2019) On the dynamical behaviour of linear higher-order cellular automata and its decidability. Inf Sci 486:73–87. https://doi.org/10.1016/j.ins.2019.02.023
11. Hu T, Liu Y, Gong LH, Ouyang CJ (2017) An image encryption scheme combining chaos with cycle operation for DNA sequences. Nonlinear Dyn 87(1):51–66
12. Hua Z, Jin F, Xu B, Huang H (2018) 2D Logistic-Sine-coupling map for image encryption. Signal Process 149(1):148–161
13. Hua Z, Zhou Y, Huang H (2019) Cosine-transform-based chaotic system for image encryption. Inf Sci 480(1):403–419
14. Lambiċ D (2017) Cryptanalyzing a novel pseudorandom number generator based on pseudorandomly enhanced logistic map. Nonlinear Dyn 89(3):2255–2257
15. Luo Y, Du M, Liu J (2015) A symmetrical image encryption scheme in wavelet and time domain. Commun Nonlinear Sci Numer Simul 20(2):447–460
16. Luo Y, Cao L, Qiu S, Lin H, Harkin J, Liu J (2016) A chaotic map-control-based and the plain image-related cryptosystem. Nonlinear Dyn 83(4):2293–2310
17. Luo Y, Tang S, Qin X, Cao L, Jiang F, Liu J (2018a) A double-image encryption scheme based on amplitude-phase encoding and discrete complex random transformation. IEEE Access 6(1):77740–77753
18. Luo Y, Zhou R, Liu J, Cao Y, Ding X (2018b) A parallel image encryption algorithm based on the piecewise linear chaotic map and hyper-chaotic map. Nonlinear Dyn 93(3):1165–1181
19. Luo Y, Zhou R, Liu J, Qiu S, Cao Y (2018c) An efficient and self-adapting colour-image encryption algorithm based on chaos and interactions among multiple layers. Multimed Tools Appl 77(20):26191–26217
20. Luo Y, Lin J, Liu J, Wei D, Cao L, Zhou R, Cao Y, Ding X (2019a) A robust image encryption algorithm based on Chua's circuit and compressive sensing. Signal Process 161(1):227–247

21. Luo Y, Ouyang X, Liu J, Cao L (2019b) An image encryption method based on elliptic curve elgamal encryption and chaotic systems. IEEE Access 7(1):38507–38522
22. Mondal B, Singh S, Kumar P (2019) A secure image encryption scheme based on cellular automata and chaotic skew tent map. J Inf Secur Appl 45(1):117–130
23. Niyat AY, Moattar MH, Torshiz MN (2017) Color image encryption based on hybrid hyper-chaotic system and cellular automata. Opt Lasers Eng 90(1):225–237
24. Pak C, Huang L (2017) A new color image encryption using combination of the 1D chaotic map. Signal Process 138(1):129–137
25. Parvin Z, Seyedarabi H, Shamsi M (2016) A new secure and sensitive image encryption scheme based on new substitution with chaotic function. Multimed Tools Appl 75(1):10631–10648
26. Ping P, Xu F, Wang Z (2014) Image encryption based on non-affine and balanced cellular automata. Signal Process 105(1):419–429
27. Ping P, Wu J, Mao Y, Xu F, Fan J (2018) Design of image cipher using life-like cellular automata and chaotic map. Signal Process 150(1):233–247
28. Su Y, Wo Y, Han G (2019) Reversible cellular automata image encryption for similarity search. Signal Process Image Commun 72(1):134–147
29. ur Rehman A, Xiao D, Kulsoom A, Hashmi MA, Abbas SA (2019) Block mode image encryption technique using two-fold operations based on chaos, MD5 and DNA rules. Multimed Tools Appl 78(7):9355–9382
30. Wang X, Guan N (2020a) A novel chaotic image encryption algorithm based on extended Zigzag confusion and RNA operation. Opt Laser Technol 131(1):106366
31. Wang X, Guan N (2020b) Chaotic image encryption algorithm based on block theory and reversible mixed cellular automata. Opt Laser Technol 132(1):106501
32. Wang X, Wang S, Zhang Y, Luo C (2018a) A one-time pad color image cryptosystem based on SHA-3 and multiple chaotic systems. Opt Lasers Eng 103(1):1–8
33. Wang X, Zhu X, Wu X, Zhang Y (2018b) Image encryption algorithm based on multiple mixed hash functions and cyclic shift. Opt Lasers Eng 107(1):370–379
34. Wang X, Xue W, An J (2020) Image encryption algorithm based on Tent-Dynamics coupled map lattices and diffusion of Household. Chaos Solitons Fractals 141(1):110309
35. Wolfram S (1986) Random sequence generation by cellular automata. Adv Appl Math 7(2):123–169
36. Wu X, Kurths J, Kan H (2018a) A robust and lossless DNA encryption scheme for color images. Multimed Tools Appl 77(10):12349–12376
37. Wu X, Wang K, Wang X, Kan H, Kurths J (2018b) Color image DNA encryption using NCA map-based CML and one-time keys. Signal Process 148(1):272–287
38. Wu X, Wang K, Wang X, Kan H, Kurths J (2018c) Color image DNA encryption using NCA map-based CML and one-time keys. Signal Process 148(1):272–287
39. Zhou M, Wang C (2020) A novel image encryption scheme based on conservative hyperchaotic system and closed-loop diffusion between blocks. Signal Process 171(1):1–14
40. Zhou Y, Bao L, Chen CLP (2014) A new 1D chaotic system for image encryption. Signal Process 97(1):172–182

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