EAI Endorsed Transactions
on Security and Safety
Research Article EAI.EU

Image encryption algorithm using chaotic maps and cellular automata

Lanhang Li*, Yuling Luo, Shubin Tang, Lvchen Cao and Xue Ouyang

[1]Guangxi Normal University

Abstract

Nowadays, some encryption schemes are not sensitive enough to plain-image, which leads to poor robustness and the scheme is vulnerability to attacks. By employing chaotic maps and cellular automata (CA), a novel image encryption algorithm is presented in this work to increase the sensitivity to plain-image and improve the security. Firstly, initial values of the two-dimensional Logistic-Sine-coupling map (2D-LSCM) and the Logistic-Sine-Cosine map (LSC) are calculated by the SHA-256 hash value of original image, and the process of diffusion is conducted next. Secondly, the key matrices are produced by iterating chaotic map 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 previous scrambled image is transformed into cipher-image by using CA. The experimental results and theoretical analysis prove that the proposed scheme owns good security as it can effectively resist a variety of attacks.

Keywords: Image encryption, Cellular automata, SHA-256, 2D-LSCM chaotic map, LSC chaotic map.

Received on 15 June 2020, accepted on 28 September 2020, published on 13 October 2020

doi: 10.4108/eai.21-6-2021.170238

1 Introduction

With computer science and technology developing rapidly, a mass of digital images are transmitted on the internet, which may be leaked due to the non-secure channels and lead to significant threat to information security. 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 to binary data and encrypted with Advanced Encryption Standard (AES). 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 [1–3]. Moreover, coupled chaotic map is a new system which can be acquired by conducting combination using one-dimensional (1D) chaotic maps. Thus, coupled chaotic map has more complex chaotic behaviour than its seed maps [4]. 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 [6], which is further used for image encryption [7]. There are mainly two CA-based image encryption methods, one is that the CA can be used to produce pseudo random numbers and the other is that the CA is utilized to encrypt the image in bit-level. Specifically, in [8], a key matrix is built using pseudo random numbers, which are generated by one-dimensional CA. In addition, an encryption algorithm using a secondorder life-like CA is designed in [9], the pixels are substituted by the CA. In [10], reversible cellular automata is applied to image encryption, and the pixels are confused by using CA and the histogram distribution in encrypted image is more uniform because of the excellent pseudo-randomness of CA. Therefore, in this algorithm, the CA is used to encrypt the scrambled image to reduce the correlation and further improve the security.

The following three aspects are main contributions of this algorithm: (1) In the proposed algorithm, a one-

* Corresponding author. Email: 1766511762@qq.com

dimensional 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 pseudo random sequence. (2) Initial secret keys are produced according to the hash value of original image, which certifies the generated pseudo random sequences are highly correlated with original image. (3) A new diffusion and permutation mechanism is designed in this algorithm. In the process of permutation, the key matrices are generated by the chaotic maps. The index matrices can be obtained by sorting each row or column of the key matrices, which are utilized to scramble the image.

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, and Section 5 gives a conclusion.

2 Preliminaries

2.1 Chaotic systems

The two-dimensional Logistic-Sine-coupling map (2D-LSCM) [11] can be given by

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

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

In addition, the Logistic-Sine-Cosine map (LSC) [12] is defined by

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

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

2.2 Cellular automata

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

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

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 is 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 $(01011010)_2 = (90)_{10}$.

Table 1. Different rules of one-dimensional 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	$S_i^{t+1} = S_{i-1}^t \odot S_{i+1}^t + (S_i^t \oplus S_{i+1}^t) \cdot S_{i-1}^t$

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 [13]. In this algorithm,

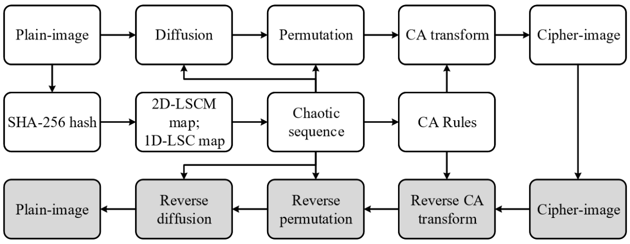


Fig.1. Flow chat of the proposed encryption algorithm.

the initial secret keys are obtained by employing the SHA-256 hash value of original image, which is calculated as *H*. *H* represents a 64 hexadecimal value since each hexadecimal denotes four binary values. Then, *H* is transformed into decimal value and is 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 $\beta = d_5 \oplus d_6 \oplus d_7 \oplus d_8$.

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 Figure 1, and the detailed encryption steps are presented next.

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

Step 2. The 2D-LSCM map 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 LSC map for $m \times n \times 8$ times. Then, the sequence *X* is reshaped into matrix X_1 with the same size as *I*. In addition, 2*m* values from sequence *Z* are selected, and quantification operations are performed by

$$\begin{cases} \boldsymbol{Y}_1 = \mathrm{mod}\left(\mathrm{floor}\left(\boldsymbol{Y} \times 10^{14}\right), 256\right), \\ \boldsymbol{R} = \mathrm{mod}\left(\mathrm{floor}\left(\boldsymbol{Z}(1:n) \times 10^{14}\right), 256\right), \\ \boldsymbol{C} = \mathrm{mod}\left(\mathrm{floor}\left(\boldsymbol{Z}(m+1:2m) \times 10^{14}\right), 256\right), \end{cases} \tag{6}$$

where $Z(1:n)$ represent the first value to the n^{th} value in sequence *Z*, $Z(m+1:2m)$ represent 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

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

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

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

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

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

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

$$C_1(x,y) = \mathrm{mod}\,(C_1(x-1, y-1) + I(x,y), 256) \oplus D(x,y), \tag{9}$$

Specially, the first value $C_1(1)$ is calculated by

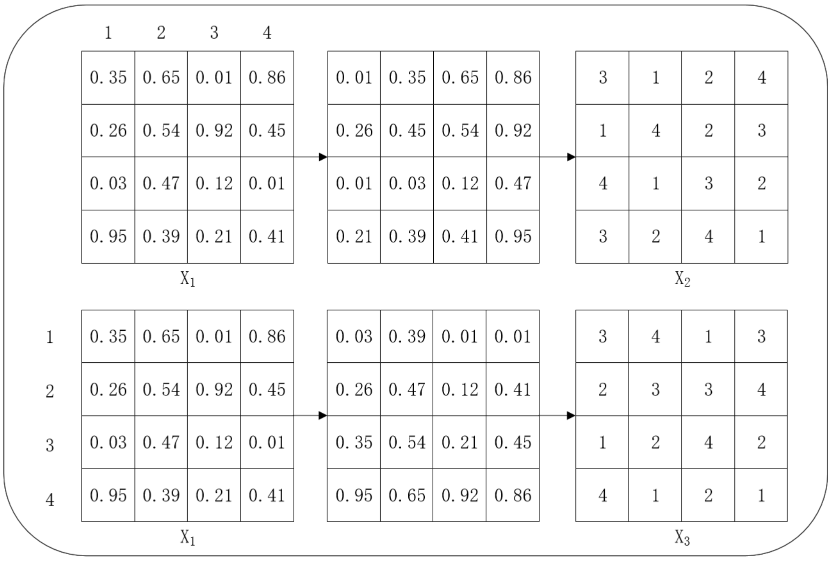
$$C_1(1) = \mathrm{mod}(I(1,1) + R(1), 256) \oplus D(1,1). \tag{10}$$

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. 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. Figure 2 shows the process of generating the index matrices for permutation when the size of image is 4 × 4.

Step 5. Cellular automata transform is adopted to obtain the final cipherimage. 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 judging the interval of the values of chaotic sequences *Z* produced by the LSC map. R_i is given by

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

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



1	2	3	4
0. 35	0. 65	0. 01	0. 86
0. 26	0. 54	0. 92	0. 45
0. 03	0. 47	0. 12	0. 01
0. 95	0. 39	0. 21	0. 41

X₁

0. 01	0. 35	0. 65	0. 86
0. 26	0. 45	0. 54	0. 92
0. 01	0. 03	0. 12	0. 47
0. 21	0. 39	0. 41	0. 95

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

X₂

1	0. 35	0. 65	0. 01	0. 86
2	0. 26	0. 54	0. 92	0. 45
3	0. 03	0. 47	0. 12	0. 01
4	0. 95	0. 39	0. 21	0. 41

X₁

0. 03	0. 39	0. 01	0. 01
0. 26	0. 47	0. 12	0. 41
0. 35	0. 54	0. 21	0. 45
0. 95	0. 65	0. 92	0. 86

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

X₃

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

What's more, the Rule numbers in the process of the CA transform are randomly selected. The chaotic sequence *Z* is obtained by iterating the LSC map and the initial value of the LSC map 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 values in C_3 is binary, and the length of C_3 is m×n×8. Then, C_3 is regarded as the inputs of the CA, and the corresponding rules of the CA transform are the sequence Z_i. The outputs can be obtained after conducting the CA transform, and the outputs are converted into decimal vector C_4. Finally, the vector C_4 is reshaped into the final encrypted image.

Obviously, the decryption process is the inverse of encryption.

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 are 512 × 512. No any useful information can be acquired from encrypted images, which are represented in Figure 3, and the plain-images can be recovered successful.

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, 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

$$NPCR = \frac{\sum_{i,j} D(i,j)}{m \times n} \times 100\%, \tag{12}$$

$$UACI = \frac{1}{m \times n}\left[\sum_{i,j} \frac{|T_1(i,j) - T_2(i,j)|}{255}\right] \times 100\%, \tag{13}$$

where m × n is the dimension of image. T_1 and T_2 are encrypted images, and their corresponding plain-images differ by one pixel. When $T_1(i,j)$ 6= $T_2(i,j)$, the difference matrix $D(i,j)$ = 1; or else, $D(i,j)$ = 0.

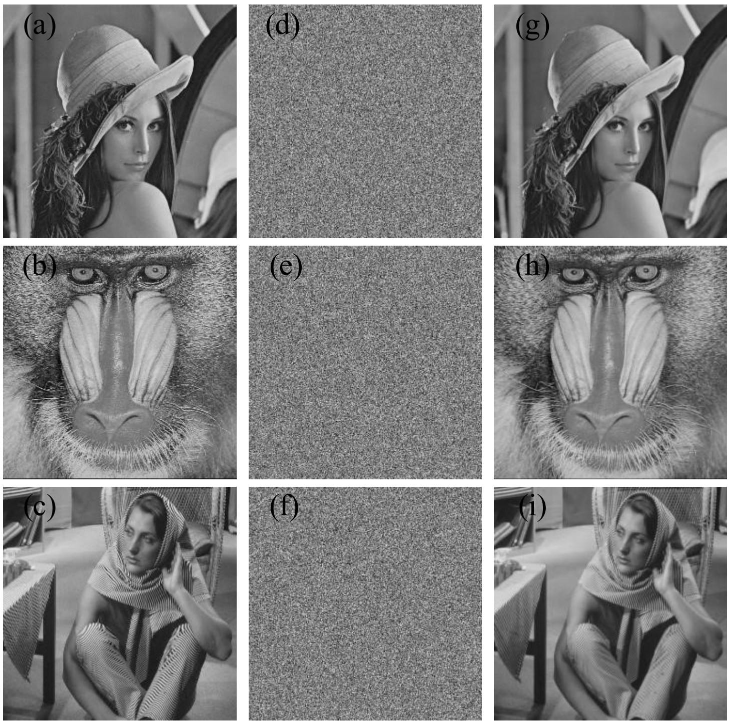


Fig.3. Experimental results. (a)-(c) original images; (d)-(f) encrypted images;(g)-(i) decrypted images.

In this test, three images are tested, and 100 pixels are selected randomly with adding one for each time. The test results and the comparison with other schemes are listed in Table 2. The results indicate that the average values are approximate to the expected value [14], which certifies that the algorithm can resist differential attack.

Table 2. *NPCR* and *UACI* of different images and the comparison results with other schemes.

Method	Image	*NPCR*(%)	*UACI*(%)
This work	“Lena”	99.6103	33.4540
	“Baboon”	99.6105	33.4643
	“Barbara”	99.6093	33.4792
[15]	“Lena”	99.59	33.41
[16]	“Lena”	99.60	33.45
[17]	“Lena”	99.61	33.32

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} [18]. The secret keys in this paper includes two control parameters r, θ, initial values x_0, x_1, y_1, and 256-bit hash value. The accuracy of the computer is limited, assuming it is 10^{-15} [19], the entire key space in this work is 2^{388}. Therefore, the proposed algorithm can defend brute force attack.

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 secret key is altered slightly. In order to prove the key sensitivity, the hash value $H(5ec042e1...f35459f5)$ is changed slightly to $H_1(4ec042e1...f35459f5)$. Figure 4(a) shows the new cipher-image when H_1 is utilized to encrypt the 512 × 512 “Lena”. In addition, Figure 4(b)-(f) show the other new encrypted images when other keys are modified. Figure 4(g)-(l) indicates that there is an enormous difference between Figure 4(a)-(f) and original encrypted image.

What’s more, the decrypted image by using an error key has an enormous difference with the correct decrypted image. Usually, this difference can be evaluated by peak signal-to-noise ratio (*PSNR*) and mean square error (*MSE*) [20], which are calculated by

$$PSNR = 10\log_{10}\frac{255^2}{MSE}, \tag{14}$$

$$MSE = \frac{1}{m \times n}\sum_{x=1}^{m}\sum_{y=1}^{n}[D(x,y) - P(x,y)]^2, \tag{15}$$

where D represents decrypted image by using 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 [21]. The key sensitivity can be also evaluated by *NPCR* and *UACI*. The results are listed in Table 3, in which certifies that the proposed algorithm has a good performance in key sensitivity.

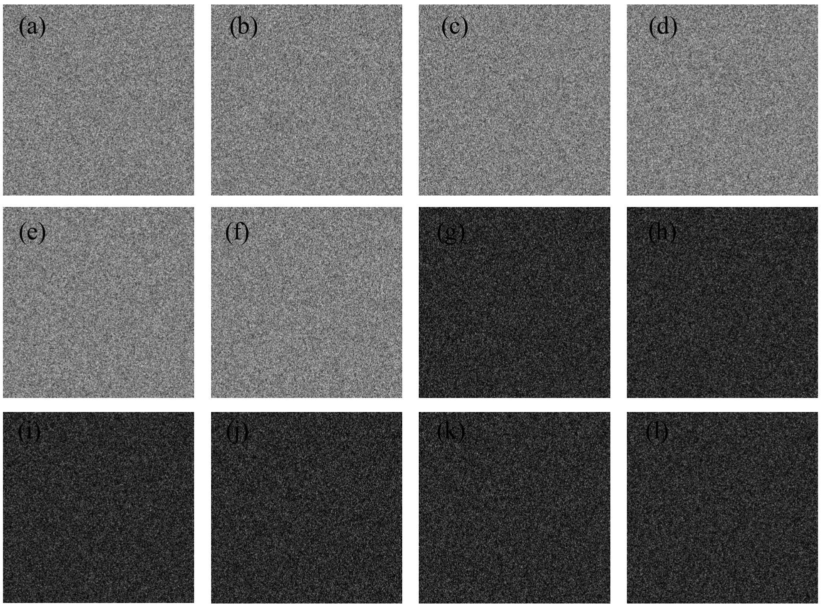


Fig.4. (a)-(f) new encrypted images when keys are modified; (g)-(l) difference images between (a)-(f) and original cipher “Lena”.

Table 3. Key sensitivity about decrypted image.

The wrong keys	*NPCR*(%)	*UACI*(%)	*MSE*	*PSNR*
H_1	99.6078	29.1718	8132.2408	9.0287
$\theta' = \theta + 10^{-16}$	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

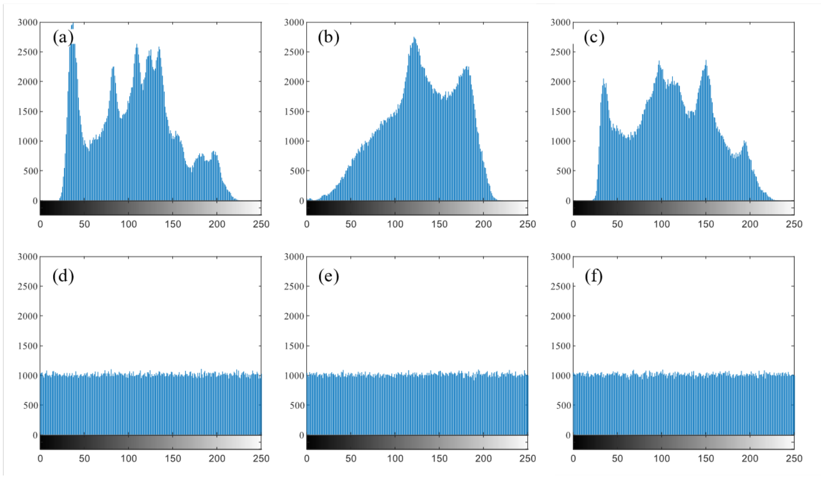


Fig.5. (a)-(c) Histograms of “Lena”, “Baboon”, and “Barbara”. (d)-(f) Histograms of their cipher-images.

4.4 The histogram analysis

Histogram of an image shows each pixel value’s distribution [22]. If the histogram is not uniform, the exposed information may be utilized by attackers. Figure 5 displays the histograms of encrypted images, which are balance distribution. Hence, the proposed algorithm is resistant against statistical attack.

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 original image contains a lot of redundant information. Hence, a secure encryption algorithm ought to eliminate these correlations. The correlation coefficients cor_{xy} can be calculated by

$$\mathrm{cor}_{xy} = \frac{\sum_{i=1}^{N}(x_i-\omega_x)(y_i-\omega_y)}{\sqrt{\sum_{i=1}^{N}(x_i-\omega_x)^2(y_i-\omega_y)^2}}, \qquad (16)$$

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 6 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 and the comparison results. The correlation coefficients of encrypted images in this paper are much nearer to zero than other methods. So the proposed algorithm is able to eliminate the correlation effectively.

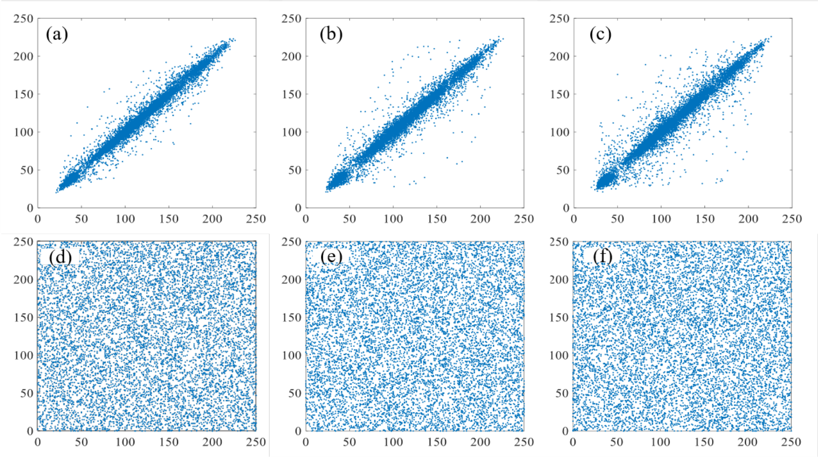


Fig.6. Correlation distribution. (a)-(c) correlation distributions in three directions of original “Lena”; (d)-(f) correlation distributions in in three directions of encrypted “Lena”.

Table 4. Correlation coefficient of original and encrypted images.

Method	Image	Original			Encrypted		
		HL	VL	DL	HL	VL	DL
Proposed	“Lena”	0.9850	0.9782	0.9633	0.0065	0.0051	-0.0005
	“Baboon”	0.7115	0.8511	0.6839	0.0046	0.0036	0.0027
	“Barbara”	0.9688	0.8933	0.8568	0.0033	-0.0026	0.0020
[23]	“Lena”	0.9771	0.9631	0.9490	0.0925	0.0430	0.0533
[24]	“Lena”	0.9503	0.9755	0.9275	-0.0226	0.0041	0.0368
[5]	“Baboon”	0.7508	0.8562	0.7153	-0.0061	0.0130	0.0017

4.6 Information entropy analysis

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

$$H(s)=\sum_{i=0}^{2^n-1} p\left(s_i\right) \log_2 \frac{1}{p\left(s_i\right)}, \qquad (17)$$

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 [8]. The nearer the entropy is to eight of cipher-image, the higher security of the scheme [25]. The results for three tested images and the comparison for “Lena” with different methods are listed in Table 5. The information entropies of encrypted image in this paper are close to eight, and higher than the other three methods.

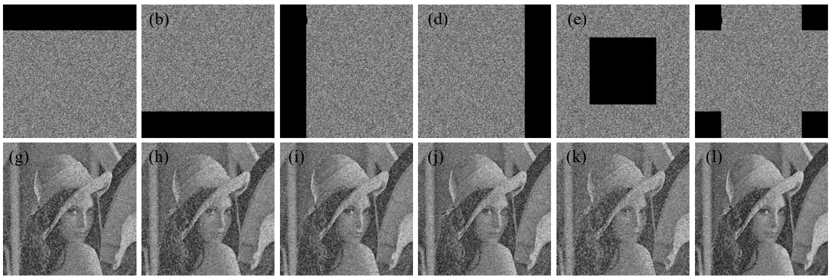


Fig.7. Data loss attack. (a)-(f) encrypted images with different degrees of data loss; (g)-(l) decrypted images.

Table 5. Information entropies of different images and the comparison results.

Method	Original	Encrypted
Proposed (Lena)	7.38712	7.99927
Proposed (Baboon)	7.35794	7.99933
Proposed (Barbara)	7.46642	7.99934
[26]	-	7.99921
[27]	-	7.99925
[28]	-	7.99923

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 [29]. Therefore, it is essential for the cryptosystem to have the ability to resist data loss and noise attacks [30].

The cipher-image "Lena" with different cropped parts and the decrypted results are displayed in Figure 7. 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 the data loss attack effectively.

In addition, the Gaussian noise(GN) is tested, which can be added to the encrypted image C by

$$C' = C + kG_N, \tag{18}$$

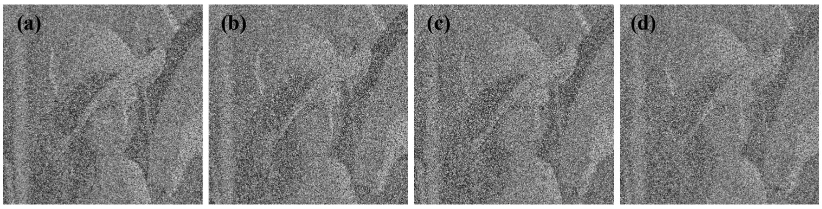


Fig.8. Decrypted images under different GN intensities: (a)-(d): k=5,10,20,30.

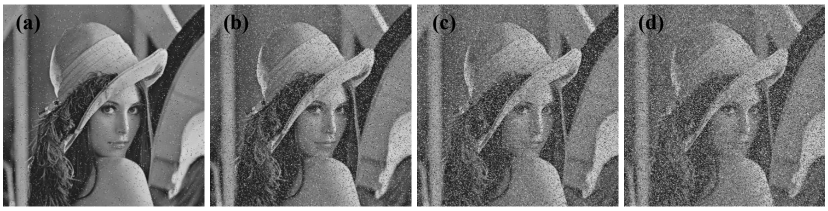


Fig.9. Decrypted images under different SPN densities (a):1%; (b):5%; (c):10%; (d):15%.

where C' is new encrypted image after adding GN, k is noise intensity, and G_N is the standard GN, and. The decrypted results after adding GN are represented in Figure 8, 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 are given in Figure 9. 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 map and the LSC map 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 [31]. Figure 10 shows the cipher-images of all black and white, and their histograms. In addition, Table 6 shows the values of 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.

5 Conclusion

Based on chaotic maps and cellular automata, a novel image encryption algorithm is proposed in this work. Firstly, initial secret keys of the chaotic maps are calculated by using the SHA-256 hash value of the original image, which leads to high key sensitivity. Then, the plain-image is diffused and scrambled by using the proposed encryption method. The final encrypted image can be acquired by transforming the scrambled image using CA. Experimental results and security analysis certify the proposed scheme has good performance in key sensitivity, ideal information entropy, and it can resist a variety of attacks, i.e., noise and data loss attacks. All these demonstrate that the proposed algorithm is suited for multimedia communication.

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

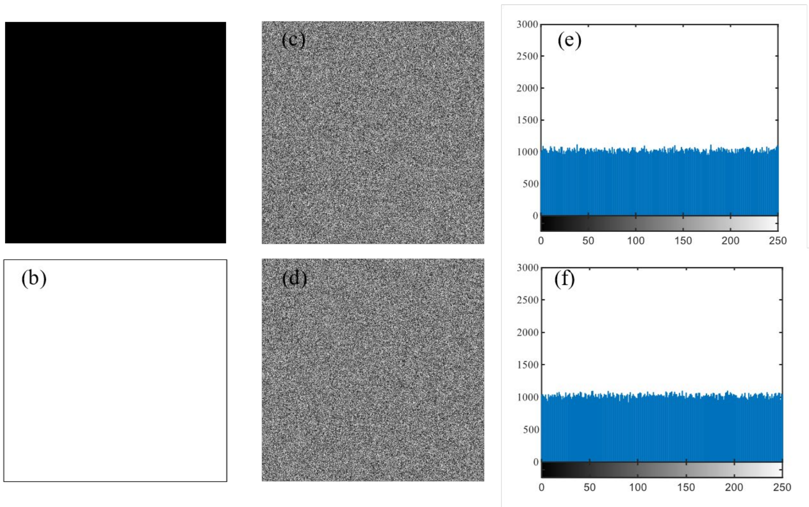


Fig.10. (a)-(b) images of all black and white, (c)-(d) encrypted images, (e)-(f) histograms of (c)-(d).

References

[1] Luo, Y., Zhou, R., Liu, J., Qiu, S., Cao, Y.: An efficient and self-adapting colourimage encryption algorithm based on chaos and interactions among multiple layers. Multimedia Tools and Applications **77**(20), 26,191–26,217 (2018)

[2] Luo, Y., Zhou, R., Liu, J., Cao, Y., Ding, X.: A parallel image encryption algorithm based on the piecewise linear chaotic map and hyper-chaotic map. Nonlinear Dynamics **93**(3), 1165–1181 (2018)

[3] Luo, Y., Ouyang, X., Liu, J., Cao, L.: An Image Encryption Method Based on Elliptic Curve Elgamal Encryption and Chaotic Systems. IEEE Access **7**(1), 38,507– 38,522 (2019)

[4] Zhou, Y., Bao, L., Chen, C.L.P.: A new 1D chaotic system for image encryption. Signal Processing **97**(1), 172–182 (2014)

[5] Chai, X., Gan, Z., Yang, K., Chen, Y., Liu, X.: An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations. Signal Processing: Image Communication **52**(1), 6–19 (2017)

[6] Wolfram, S.: Random sequence generation by cellular automata. Advances in Applied Mathematics **7**(2), 123–169 (1986)

[7] Ping, P., Xu, F., Wang, Z.: Image encryption based on non-affine and balanced cellular automata. Signal Processing **105**(1), 419–429 (2014)

[8] Niyat, A.Y., Moattar, M.H., Torshiz, M.N.: Color image encryption based on hybrid hyper-chaotic system and cellular automata. Optics and Lasers in Engineering **90**(1), 225–237 (2017)

[9] Ping, P., Wu, J., Mao, Y., Xu, F., Fan, J.: Design of image cipher using life-like cellular automata and chaotic map. Signal Processing **150**(1), 233–247 (2018)

[10] Su, Y., Wo, Y., Han, G.: Reversible cellular automata image encryption for similarity search. Signal Processing: Image Communication **72**(1), 134–147 (2019)

[11] Hua, Z., Jin, F., Xu, B., Huang, H.: 2D Logistic-Sine-coupling map for image encryption. Signal Processing **149**(1), 148–161 (2018)

[12] Hua, Z., Zhou, Y., Huang, H.: Cosine-transform-based chaotic system for image encryption. Information Sciences **480**(1), 403–419 (2019)

[13] Wu, X., Wang, K., Wang, X., Kan, H., Kurths, J.: Color image DNA encryption using NCA map-based CML and one-time keys. Signal Processing **148**(1), 272–287 (2018)

[14] Hu, T., Liu, Y., Gong, L.H., Ouyang, C.J.: An image encryption scheme combining chaos with cycle operation for DNA sequences. Nonlinear Dynamics **87**(1), 51–66 (2017)

[15] Chai, X., Chen, Y., Broyde, L.: A novel chaos-based image encryption algorithm using DNA sequence operations. Optics and Lasers in Engineering **88**(1), 197–213 (2017)

[16] Wang, X., Liu, C.: A novel and effective image encryption algorithm based on chaos and DNA encoding. Multimedia Tools and Applications **76**(5), 6229–6245 (2017)

[17] Sun, S.: Chaotic image encryption scheme using two-by-two deoxyribonucleic acid complementary rules. Optical Engineering **56**(11), 1–9 (2017)

[18] Alvarez, G., Li, S.: Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems. International Journal of Bifurcation and Chaos **16**(8), 2129–2151 (2006)

[19] Lambi´c, D.: Cryptanalyzing a novel pseudorandom number generator based on pseudorandomly enhanced logistic map. Nonlinear Dynamics **89**(3), 2255–2257 (2017)

[20] Chai, X., Gan, Z., Yuan, K., Lu, Y., Chen, Y.: An image encryption scheme based on three-dimensional Brownian motion and chaotic system. Chinese Physics B **26**(2), 1674–1056 (2017)

[21] Luo, Y., Du, M., Liu, J.: A symmetrical image encryption scheme in wavelet and time domain. Communications in Nonlinear Science and Numerical Simulation **20**(2), 447–460 (2015)

[22] Wu, X., Wang, K., Wang, X., Kan, H., Kurths, J.: Color image DNA encryption using NCA map-based CML and one-time keys. Signal Processing **148**(1), 272–287 (2018)

[23] Ye, G., Huang, X., Zhang, Y., Wang, Z.: A self-cited pixel summation based image encryption algorithm. Chinese Physics B **26**(1), 1–8 (2017)

[24] Xu, L., Gou, X., Li, Z., Li, J.: A novel chaotic image encryption algorithm using block scrambling and dynamic index based diffusion. Optics and Lasers in Engineering **91**(1), 41–52 (2017)

[25] Wang, X., Wang, S., Zhang, Y., Luo, C.: A one-time pad color image cryptosystem based on SHA-3 and multiple chaotic systems. Optics and Lasers in Engineering **103**(1), 1–8 (2018)

[26] Zhou, Y., Bao, L., Chen, C.L.P.: A new 1D chaotic system for image encryption. Signal Processing **97**(1), 172–182 (2014)

[27] Zhang, W., Yu, H., Zhao, Y., Zhu, Z.: Image encryption based on three-dimensional bit matrix permutation. Signal Processing **118**(1), 36–50 (2016)

[28] Zhang, L.Y., Hu, X., Liu, Y., Wong, K.W., Gan, J.: A chaotic image encryption scheme owning temp-value feedback. Communications in Nonlinear Science and Numerical Simulation **19**(10), 3653–3659 (2014)

[29] Luo, Y., Lin, J., Liu, J., Wei, D., Cao, L., Zhou, R., Cao, Y., Ding, X.: A robust image encryption algorithm based on Chua's circuit and compressive sensing. Signal Processing **161**(1), 227–247 (2019)

[30] Chen, J., Zhu, Z., Fu, C., Zhang, L., Yu, H.: Analysis and improvement of a doubleimage encryption scheme using pixel scrambling technique in gyrator domains. Optics and Lasers in Engineering **66**(1), 1–9 (2015)

[31] Luo, Y., Cao, L., Qiu, S., Lin, H., Harkin, J., Liu, J.: A chaotic map-control-based and the plain image-related cryptosystem. Nonlinear Dynamics **83**(4), 2293–2310 (2016)