Nonlinear Dyn
https://doi.org/10.1007/s11071-022-08021-1

ORIGINAL PAPER

A robustness-improved image encryption scheme utilizing Life-liked cellular automaton

Wenrui Lv · Junxin Chen · Xiuli Chai · Chong Fu

Received: 29 June 2022 / Accepted: 15 October 2022

Abstract Image encryption is considered as an effective method to protect image against revealing. As a discrete dynamic system, the Life-liked cellular automaton (CA) has good chaotic performance and has been applied for image encryption. This paper proposes a robust block encryption scheme, based on reversible Life-liked CA with balanced rule. The proposed method adopts classic confusion–diffusion structure on block level and reversible Life-liked CA within block. An effect permutation method is developed to reduce the iteration rounds of whole system, while the diffusion module adopts reversible Life-liked CA with balanced rule to encrypt the blocks to noise-like ones. Performance analyses show the proposed scheme have good cryptographic features, satisfactory security for defeating common attacks and robustness to resist data loss and random noise.

Keywords Image encryption · Life-liked cellular automaton · Chaotic system · Block encryption

W. Lv · C. Fu (✉)
School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
e-mail: fuchong@mail.neu.edu.cn

W. Lv
e-mail: 2101662@stu.neu.edu.cn

J. Chen
College of Medicine and Biological Information Engineering, Northeastern University, Shenyang 110819, China
e-mail: junxinchen@ieee.org

X. Chai
School of Artificial Intelligence, Henan University, Zhengzhou 450046, China
e-mail: chaixiuli@henu.edu.cn

C. Fu
Engineering Research Center of Security Technology of Complex Network System, Ministry of Education, Shenyang, China

1 Introduction

With the dramatic progresses of communication technologies, huge of digital images have been transmitted on the public networks and shared in mobile phones. The security of these images not only involves individual privacy, but also relates to society and government security. As one of the most effective privacy protection methods, image encryption has attracted worldwide attentions. Among the image encryption proposals, two different nonlinear dynamics, i.e., cellular automaton (CA) and chaotic system, have been widely adopted. Both the dynamic systems have many advantages for image encryption, such as sensitivity to initial condition, unpredictable evolution and random behavior.

There are many famous one-dimensional (1D) chaotic maps which have been adopted for image encryption, such as logistic map [1], sine map [2] and tent map [3]. Though they are simple to be implemented, they also have drawbacks. In [4], Li et al. stated that when a chaotic system is implicated on a finite precision platform, chaos degradation will happen and the chaotic system may degrade to a periodic

Published online: 02 November 2022

one. In addition, Hua et al. [5] presented the average cycle length of some chaotic maps under different precisions. Some weak chaotic systems have been demonstrated to have security problems [6,7], and in this direction, many encryption schemes have been cracked [8,9]. More complex chaotic maps are needed to improve the security of encryption scheme. On the other hand, the CA has advantages over chaotic systems. As a time and space discrete dynamic system, it can avoid chaos degradation on different finite precision platforms. In addition, it can be easily implicated in encryption scheme without much additional hardware and software complexity. Many encryption schemes based on CA have been proposed. In [10], Wolfram first applied CA in cryptography. The works of [11–13] applied CA as pseudorandom number generator (PRNG), while the proposals in [14,15] adopted state attractor of some CAs to encrypt images. Jhon Conway introduced the well-known Game of Life [16], which is a kind of 2D CA with more complex chaotic behaviors, based on it, many researchers have developed a series of CA, called Life-liked CA and used them for image encryption [12,17,18].

As for encryption framework, many algorithms adopt the classic confusion–diffusion [19] architecture. However, this structure has inherent drawbacks, and the opponents can crack the encryption scheme by plaintext attack no matter which chaos system is adopted [20]. Researchers have proposed many methods to improve the security against plaintext attacks. However, some of these methods sacrifice so much robustness that they cannot defeat data loss and random noise. Wang et al. [21] proposed an algorithm with variable control parameters, which is able to bring unique chaos sequence for different plaintext. Though this method can resist plaintext analysis, slight error in the cipher pixel could make the original plain image fail to be recovered. The studies [17,22,23] adopted reversible CA (RCA) for diffusion, which could defeat plaintext attacks. However, because of the good diffusion property of CA, these methods also lack of robustness. Taking the method in [17] as an example, Ping et al. divide the whole permuted image into 2 binary images as the initial cell states of a second-order RCA and then iterate the RCA certain steps to produce the cipher image. Due to the satisfactory chaotic performance of RCA, only one-cell state modification could influence entire space. However, if one state in cipher image is changed, the whole recovered image could be interfered. Therefore, if a method adopts RCA to encrypt whole image, its robustness is limited.

In this work, we propose a block encryption scheme, based on the classic permutation–diffusion structure. The plaintext is first divided as a series of blocks with size 8×8, and then, the permutation is implemented among blocks, i.e., shuffle the image blocks yet keep the pixels within each block unchanged. The permutation sorts all the blocks by random numbers and specially ensure that the last block in image sorted as the first one. The diffusion module adopts reversible Life-liked CA with balanced rule in block-wise. Each bit plane's bit states are the initial RCA's configuration and then will be iterated several steps with a balanced rule to yield the encrypted block. The 2D logistic–sine coupling map (2D-LSCM) is employed to generate the random numbers used in the encryption process. A mechanism is developed to share chaos variables between the permutation and diffusion modules, so that the required chaotic iterations are reduced and the cryptosystem's efficiency is further improved. We find that a proper designed permutation method could make sure that 2 permutation–diffusion rounds are sufficient to achieve a satisfactory diffusion performance. Based on this concept, we propose the permutation method with the ability to shuffle image pixels and sort the last block as the first one, so as to greatly improve the efficiency of encryption system. In diffusion phase, we employ Life-liked CA due to its good chaos performance and implement it in block-wise. The CA can ensure the security and diffusion performance within a block. On the other hand, an error in cipher image will be restricted within one block rather than affect the whole image so as to improve the robustness of the cryptosystem. Therefore, this method can greatly overcome the drawback, lack of robustness, in [17]. The simulation results and performance analysis indicate that the proposed cryptosystem can encrypt various types of plain image into noise-like ciphertexts, and it has sufficient security against various attacks and robustness against data loss or random noise.

The remaining parts in this paper are organized as follows. Section 2 introduces some related work, such as 2D-LSCM and reversible Life-liked CA. Section 3 is the main body and presents the proposed algorithm, while Sect. 4 analyzes the computation complexity and the ability to encrypt other image types in theory and Sect. 5 shows the simulation results of the scheme.

Table 1 Descriptions of main abbreviations

Abbreviation	Description
CA	Cellular automaton
RCA	Reversible cellular automaton
2D-LSCM	2D Logistic–sine coupling map
NPCR	Number of pixel change rate
UACI	Unified average changed intensity
MSE	Mean square error
PSNR	Peak signal-to-noise ratio

The security and robustness analyses are carried out in Sect. 6, and Sect. 7 concludes our work.

2 Preliminaries

This section introduces three significant models utilized in this paper and illustrates their advantages in image encryption. Some abbreviations used in this paper are listed in Table 1.

2.1 Two-dimensional logistic–sine coupling map

In [5], Hua et al. introduce the 2D-LSCM and illustrate its good chaotic property. The 2D-LSCM is an expanding map of two basic 1D chaotic maps, the logistic map [1] and the sine map [2]. The logistic map is defined as

$$x_{i+1} = 4 \cdot \eta \cdot x_i \cdot (1 - x_i),$$

where the control parameter $\eta \in [0, 1]$. The sine map is defined as

$$x_{i+1} = \beta \cdot sin(\pi x_i),$$

where the control parameter β is also in the interval [0, 1]. However, these two chaotic maps have many weaknesses, such as simple behaviors and frail chaotic intervals. Therefore, to achieve sufficient security of a encryption scheme, more complex chaotic system is expected. The 2D-LSCM is obtained by coupling the two maps to have quite complex chaos, and it is defined by

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

where θ is the control parameter and $\theta \in [0, 1]$. The 2D-LSCM has complex chaotic behavior can account for three aspects. First, it couples the logistic and sine map together, the complexity of the two maps are mixed. Then it performs a sine transform to the coupling result. At last, it expands the dimension to 2D.

Hua et al. [5] evaluate the chaos performance of 2D-LSCM in terms of chaos trajectory, Lyapunov exponent [24], Kolmogorov entropy [25] and dynamical degradation. It is shown that 2D-LSCM has chaotic behavior when $\theta \in (0, 1)$, and it would have hyperchaotic behavior when $\theta \in (0, 0.34) \cup (0.67, 1)$. In addition, the output sequences of 2D-LSCM can pass all the subtests in National Institute of Standards and Technology (NIST) SP800-22 [26]; it indicates that 2D-LSCM can generate random sequences that are suitable for image encryption.

2.2 Life-liked CA with balanced rules

The CA is an abstract dynamic system with discrete and finite states. Due to its local rules, the CA could present complex chaotic behavior after certain discrete time evolution.

2.2.1 Life-liked CA

Life-liked CA is a certain type of CA; generally, it can be described as $CA = \{C, S, V, f\}$ [17].

- C is a two-dimensional cell space and can be described as $C = \{(i, j) | \ 1 \le i \le m, 1 \le j \le n\}$.
- S represents the cell's states, and $S = \{1, 0\}$ generally refers to the survival and death of a cell, respectively.
- V is the set of a cell's neighbors. In two-dimensional cell place, a cell generally has two types neighborhoods, von Neumann neighborhood and Moore neighborhood. Moore type is generally utilized in Life-liked CA, and V set contains 8 surrounding neighbors.
- f is the local function, also called as the local rule. It determines the state of one cell in the next iteration step, according to its present and the neighbors' states. A local function of Life-liked CA can be described by Eq. (2), where $S^t_{i,j}$ represents the state of a cell at position (i, j) at the tth discrete time.

$$S^{t+1}_{i,j}$$

$$= f(S^t_{i-1,j-1}, S^t_{i-1,j}, S^t_{i-1,j+1}, S^t_{i,j-1}, S^t_{i,j}, S^t_{i,j+1}, S^t_{i+1,j-1}, S^t_{i+1,j}, S^t_{i+1,j+1}). \quad (2)$$

In this paper, all cells in the space follow the same local rule f. Thus, the rule is a function determining the iteration method of a CA system.

In addition, because the CA is implemented on a finite computation system, the boundary condition should be considered. Generally, periodic boundary is utilized, because it can simulate the infinite situation without losing any properties. The periodic boundary in this paper is described as Eq. (3), where i, j, u, v represent the coordinates in cell space, and m, n are the size of space.

$$S^t_{i,j} = S^t_{u,v} \quad i = \mathrm{mod}(u-1, m+1),\ j = \mathrm{mod}(v-1, n+1). \quad (3)$$

The local rule of a Life-liked CA is described as Bx/Sy, where B and S, respectively, represent "Birth" and "Survival," the two behavior of a cell. In addition, because Life-liked CA utilizes Moore neighborhood, the central one and its neighbors are 9 cells. The variable sets x and y refer to the number of survival numbers of the 9 cells. The rule means that if the central cell's state is "death" at step t, and there are x survival neighbors around it, at the next step, it will "reborn." Similarly, if the central cell's state is "survival," and there are y survival neighbors around it, at the next step, it will keep alive. For example, John Conway proposed the famous Life-liked CA "Game of Life," which can be described as "B3/S23." The set $x = \{3\}$, $y = \{2, 3\}$, which means a death central cell has 3 survival neighbors, it will reborn at next step; and a survival central cell, only its survival neighbors are 2 or 3, and it will keep alive.

2.2.2 Balanced rules

After iterating certain steps following the local rules, the ratio between survival and dead cells are almost stable, even though the space distribution of them is chaotic. Therefore, the local rule can control final ratio between survival and dead cells. Generally, in each bit plane of a cipher image, the distributions of bit "0" and "1" are almost uniform. In other words, the percentage of each bit is around 50% [27]. Thus, if a local rule can control the percentage of cells' states approximately equal, this local can be utilized in image encryption [17].

In [17], Ping et al. propose a method to evaluate whether a local rule is balanced. The method can be summarized as follows.

Step 1. According to the local rule, list all the birth and death cases in the sets B, D.

Step 2. Calculate the number of birth and death events with Eq. (4), where a_i is a Boolean value. If i belongs to the set, $a_i = 1$; otherwise, $a_i = 0$.

$$T = \sum_{i=0}^{8} C_8^i \times a_i. \quad (4)$$

Step 3. Only when the birth event number T_b equals to death event number T_d, the Life-liked CA is balanced.

For example, the Life-liked CA with local rule $B1357/S02468$, the set $B = \{1, 3, 5, 7\}$, $D = \{1, 3, 5, 7\}$. Thus, the event value $T_b = T_d = 128$, the CA is balanced.

Figure 1 presents the evolution process of a Life-liked CA with local rule $B1357/S02468$, whose size is 128×128. Take Fig.1a–d; for example, the percentage of alive cell in initial CA is 9.72%; then, the percentage rapidly tends to 50%. At steps 2 and 5, the percentage are, respectively, 42.68% and 49.66%. Finally, at step 100, it comes to 49.95%. As to Fig. 1e–h and i–l, they, respectively, present the evolution process with initial 50 and 90% alive cells. At the step 100, all the CAs approximately have 50% alive cells. Therefore, no matter the initial percentage of alive cells, after several steps' iteration, the final amount distribution is uniform.

2.3 Reversible Life-liked CA

Because of the complex chaos performance of CA system, a Life-liked CA is generally not reversible. This means the evolution process would be lost. Therefore, a reversible CA is needed for the propose of recovering the original information in an encryption scheme. In [17], a second-order CA is defined as

$$S^{t+1} = F(S^t) \oplus S^{t-1}, \quad (5)$$

where S^t represents whole CA's configuration at time step t, F is local rule of CA which is used to update the configuration and symbol $\oplus$ is XOR operation. This formula means the configuration at step $t+1$ not only

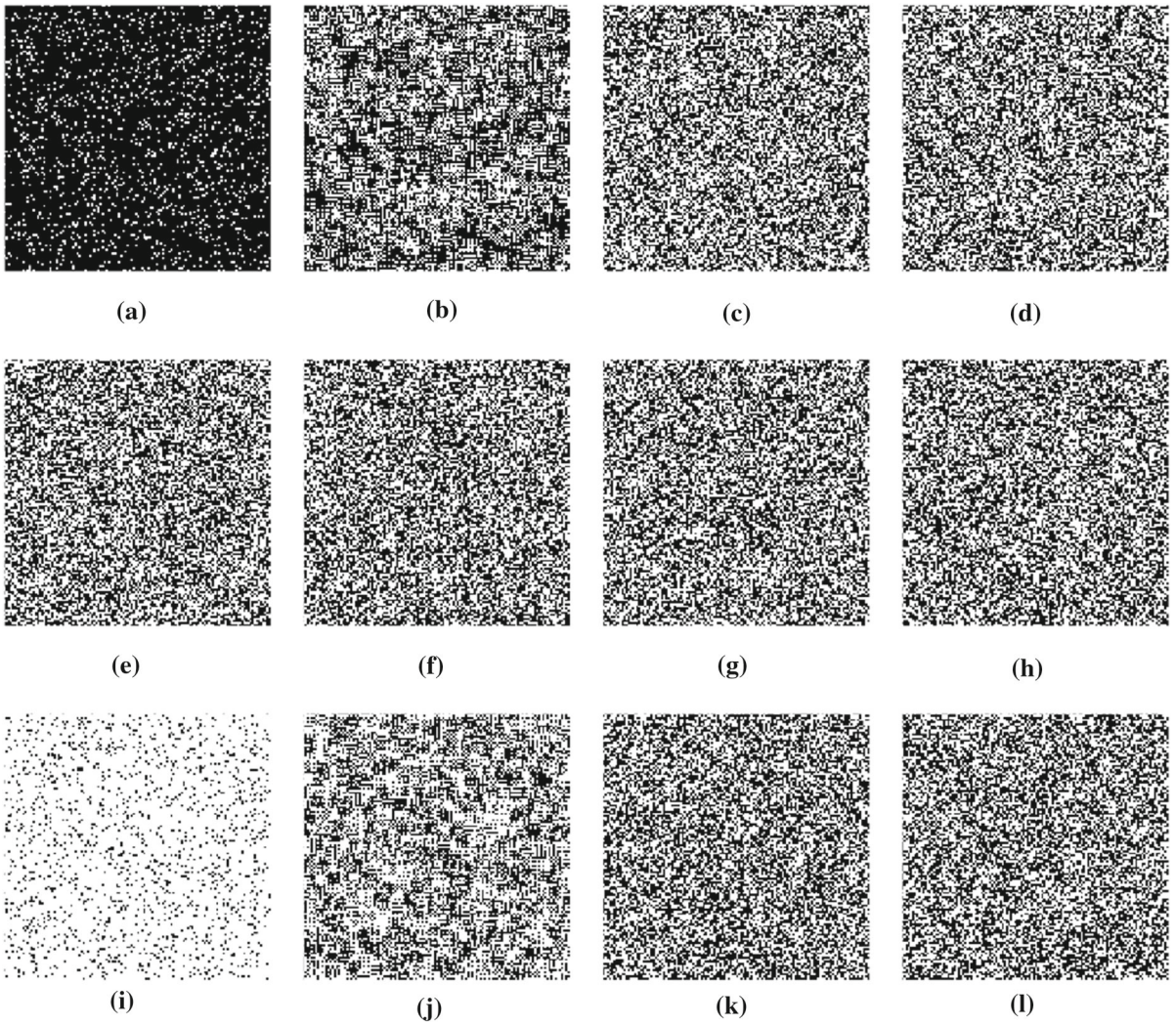

Fig. 1 Evolution process of balanced CA: **a**, **e**, **i** initial images with alive cells' percentages of 10, 50 and 90%; **b**, **f**, **g** evolution images at step 2; **c**, **g**, **k** evolution images at step 5; **d**, **h**, **l** evolution images at step 100

influenced by the state at step t, but also by the state at step $t-1$. Similarly, the state at time $t-1$ could be obtained by

$$S^{t-1} = F(S^t) \oplus S^{t+1}.$$

In addition, Eq. (5) can extend to higher order. In [22], a fourth-order CA is developed as

$$S^{t+1} = F(S^t) \oplus F(S^{t-1}) \oplus F(S^{t-2}) \oplus S^{t-3}, \qquad (6)$$

and the reverse process is defined as

$$S^{t-3} = F(S^{t-2}) \oplus F(S^{t-1}) \oplus F(S^t) \oplus S^{t+1}.$$

Because of the chaotic property of Life-liked CA, any slight difference in initial configuration could account for extreme difference in final states. However, this property also decrease the robustness of an encryption system, because a slight difference in the final configuration may make the reverse process fail to recover the original image.

Figure 2 shows the robustness analysis of a third-order reversible Life-liked CA whose size is 64×64. Figure 2a, f and k shows the initial third-order CA configurations. The second column Fig. 2b, g and l is the corresponding iterated results of the first column. In these 3 images, only one random cell in Fig. 2g is selected and reversed, generating the third column Fig. 2c, h, m. The fourth column images show the dif-

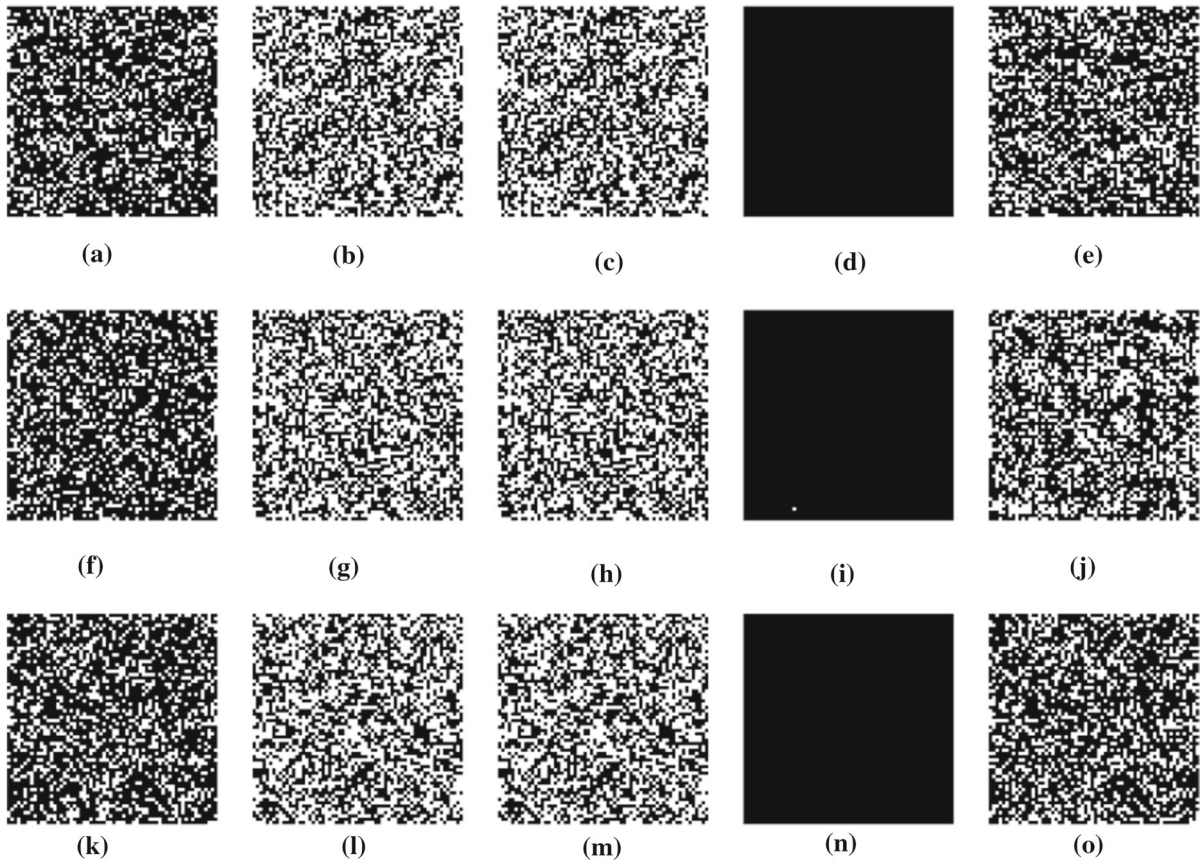

Fig. 2 Robustness analysis of a third-order Life-liked CA: **a**, **f**, **k** the initial configuration; **b**, **g**, **l** the evolution results; **c**, **h**, **m** introduce one state difference; **d**, **i**, **n** the difference between **b**, **g**, **l** and **c**, **h**, **m**; **e**, **j**, **o** the reversed result of **c**, **h**, **m**

ference between second and third column images. As for the fifth column images, they are the results recovered from the third images, and they are much different from the first column images. It is clear that only an one-cell difference is introduced to the final evolved image, and the recovered images have obvious difference. Thus, if the reversible Life-liked CA is directly applied for image encryption [17], the cryptosystem could lack of robustness.

3 The proposed method

This section introduces the modules of the proposed image encryption scheme, including preprocessing, permutation and diffusion.

As it is demonstrated in Sect. 2.3, the reversible Life-liked CA has good diffusion property, but it lacks robustness. Thus, this proposed scheme adopts block encryption to increase the robustness of system. In our proposal, a block consists of 8×8 adjacent pixels. Figure 3 shows the whole structure of the scheme. The chaos variable generator adopts 2D-LSCM chaotic map introduced in Sect. 2.1. In [28], it is indicated that the chaotic system iteration process has the highest computational complexity in an image encryption scheme. Therefore, it can improve the efficiency of whole system if the chaotic variables are utilized properly.

3.1 Chaos variables generator

In this module, chaos variables are generated for encryption referring to a secret key which is a 256-bit sequence. The chaos variable sequence is generated by the following steps.

Step 1. Calculate the number of blocks in the image. In this scheme, the 8×8 pixel block is adopted,

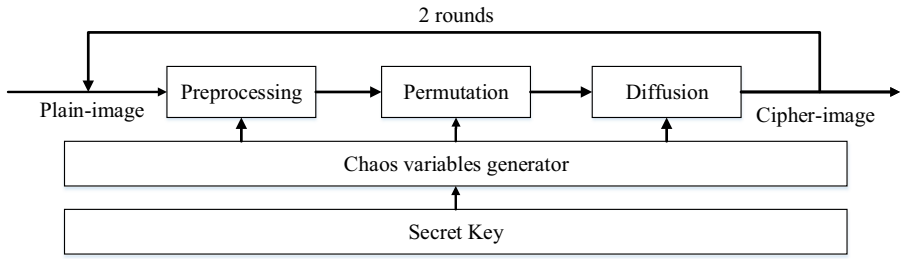

Fig. 3 Structure of proposed encryption system

and if the image's size cannot be divisible by 8, the image should firstly fill some pixels. The number of blocks in the image whose size is of $M \times N$ is

$$L = \lceil M/8 \rceil \times \lceil N/8 \rceil.$$

Step 2. Calculate the initial values of the 2D-LSCM system. There are 5 parameters $\{x_0, y_0, r, a_1, a_2\}$ obtained from the key K, where (x_0, y_0) are initial coordinates in the 2D-LSCM, r is the control parameter and a_1, a_2 are perturbation values to change r in the two encryption rounds. The variables x_0, y_0, r are, respectively, extracted from a non-overlapping 64-bit sequence in K. Suppose that a 64-bit sequence is denoted as $b_1 b_2 ... b_{64}$, a variable u can be obtained by

$$u = \left(\sum_{i=1}^{32} b_i 2^{-i} \right) \times \left(\sum_{i=33}^{64} b_i 2^{-(i-32)} \right).$$

The variables a_1, a_2 are two integers, whose values v are obtained from a 32-bit string $b_1 b_2 ... b_{32}$ by

$$v = \sum_{i=1}^{32} b_i 2^{32-i}.$$

To ensure a one-bit modification could account for totally different chaos variable sequence, r_i is obtained from $\{a_1, a_2\}$ by

$$\begin{cases} r_1 = a_1 + a_2 \\ r_2 = a_1 \oplus a_2 \end{cases}.$$

At the ith round, the control parameter θ of the 2D-LSCM is extracted by

$$\theta_i = \mathrm{mod}((r \times r_i), 1).$$

Step 3. Iterate the 2D-LSCM system for generating sufficient chaos variables for each block. Firstly, 30-round iteration is conducted to avoid initial effect of the chaotic system. In this scheme, each block needs 9 chaos variables in the encryption steps. For one block, iterate the 2D-LSCM system 5 rounds to generate the chaos variables $\{x_1, y_1 ... x_5, y_5\}$. Specially, the ninth chaos number utilized in the block is $(x_5 + y_5)/2$.

3.2 Preprocessing

In this module, an image is filled to make its width and height both can be divisible by 8. In addition, because the Life-liked CA cannot encrypted block with identical state "0" or "1," it is critical to add some disturbing pixels. The preprocessing can be described as follows.

Step 1. Fill the plain image. Suppose that the image's height is M, and width is N. Fill the image's height to M' and width to M' with pixel value "0." The values of M', N' are calculated by

$$\begin{cases} M' = \lceil m/8 \rceil \times 8 \\ N' = \lceil n/8 \rceil \times 8 \end{cases}.$$

Step 2. In each block, 4 disturbing pixels are added. With the corresponding chaos variable sequence, the second–fifth variables determine the position of disturbing pixels, and the 6th–9th ones determine the values. Each block can be transformed to a 64-pixel one-dimensional sequence, and the position and value of disturbing pixels are, respectively, defined by Eqs. (7)–(8),

$$pos = \mathrm{mod}\left(\lfloor x(i) \times 10^{14} \rfloor, 64 \right), \tag{7}$$

$$pix = \mathrm{mod}\left(\lfloor y(i) \times 10^{14} \rfloor, 256 \right), \tag{8}$$

where $x(i)$ and $y(i)$ are chaos variables for generating the ith disturbing pixel's position and value, respectively. Therefore, the pixel after preprocessing can be described as

$$B(pos) = \mathrm{mod}\left(B(pos) + pix, 256 \right),$$

where $B(i)$ represents the ith pixel value in a block. Step 3. Repeat Step 2 until all the blocks are preprocessed.

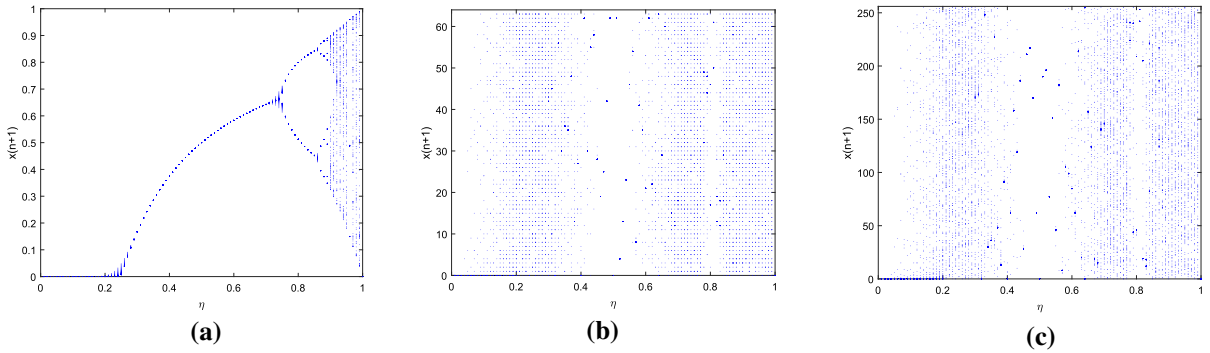

Fig. 4 Comparison bifurcation diagram of original chaos map and the map using modulars: **a** bifurcation diagram of original 1D logistic map; **b** bifurcation diagram of enhanced logistic map using *mod* 64; **c** bifurcation diagram of enhanced logistic map using *mod* 256

The literature [29] has demonstrated that modular operation can enhance the randomness of chaotic system. They illustrate the *mod* 255 and *mod* 1023 functions can broaden the spectrum of chaotic maps, which helps to improve the randomness of chaos system. Similarly, in this phase, we adopt *mod* 64 and *mod* 256 functions. These two modular operation also can enhance the robustness of chaotic system. Figure 4 shows the bifurcation diagram of 1D logistic map and the map using *mod* 64 and *mod* 256 functions. Figure 4a shows the original bifurcation diagram of logistic map, and it is clear that the map only has a narrow parameter range when the map is chaotic. Figure 4b, c shows bifurcation diagram of enhanced logistic map using *mod* 64 and *mod* 256 functions. It is evident that the spectrum of enhanced map is broadened; therefore, the modular operations in this phase can enhance chaotic performance.

3.3 Permutation

In this module, we propose a block permutation algorithm which can improve the robustness and efficiency of the encryption system. In a cryptosystem, for improving the diffusion performance, a common method is to increase the encryption rounds of the whole system. In most cases, 2-round encryption is generally insufficient to achieve a satisfactory diffusion performance. An example of the relationship between permutation and diffusion performance is illustrated in Fig. 5. In this example, the permutation phase adopts Arnold map, and diffusion phase conducts pixel-by-pixel encryption. However, after 2-round encryption, the first pixel in cipher image is not influenced by diffusion effect. That is because we cannot ensure the diffused cipher pixels of the first encryption round can be spread to the start of the permutation ciphertext in the second round.

Previous works have attempted to introduce certain diffusion effects in the permutation phase. Such a concept would improve the diffusion performance, yet it will decrease the robustness of system. We notice that the last pixel will be certainly influenced in a single permutation–diffusion round no matter which of the plain pixel has a modification. Therefore, if a permutation is able to move the last pixel to the first one, 2 permutation–diffusion encryption iterations are sufficient to yield a satisfactory diffusion effect, i.e., a plain pixel's slight modification can spread to the whole ciphertext. In this direction, our permutation approach is developed as follows.

Step 1. Obtain chaos variable used in the permutation phase. In each block's chaos variables sequence, the first value is used as the label of block. And specially, the last block does not use the variable in sequence, and its label value is fixed as "0." This means the last block has the least label value among all the blocks.

Step 2. Obtain the block order in permuted image by the label value. Sort the label value by ascending order. $s(i)$ represents order of the label value of the ith plain image block's in label sequence. So, the ith block in permuted image PB_i can be described as

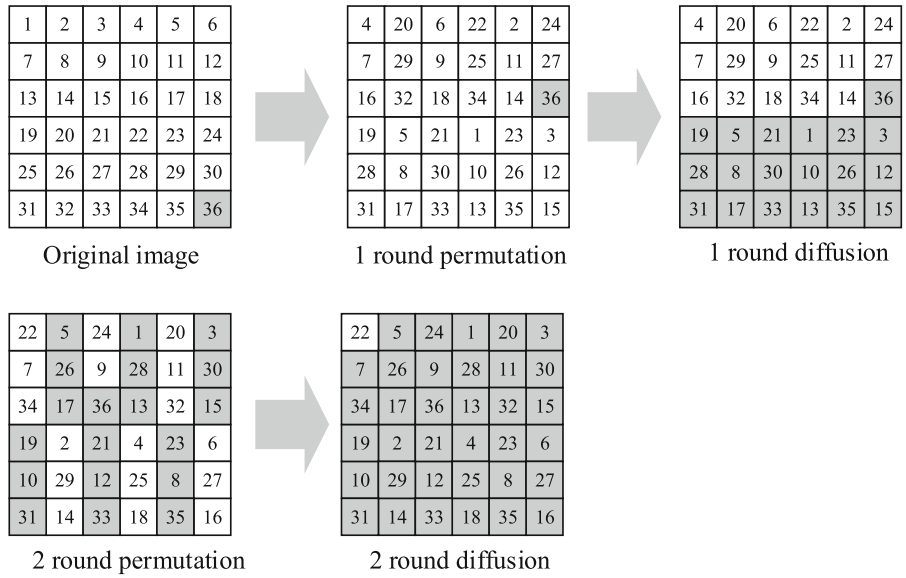

Fig. 5 Relationship between permutation and diffusion performance

$PB_{s(i)} = B_i$.

This step can be illustrated in Fig. 6, where the integers represent block order in the original image, and the decimals are value of labels in each block.

Similarly, in decryption phase, the ith block in recovered image RB_i is described as

$RB_i = PB_{s(i)}$.

As Fig. 6 shows, in plain image the block 16 is the last one, and in permuted image, the block 16 is moved to the first one.

3.4 Diffusion

In this module, each of the image block is encrypted to a noise-like one. Taking the 8-bit grayscale image as an example, each block can divide into 8 bit planes with size of 8×8. The diffusion phase proceeds as follows.

Step 1. Shuffle the bit planes and calculate iteration step T. The encryption order is determined by the first eight chaos variables $\{x(1), x(2)...x(8)\}$. Each bit plane P_i has a corresponding chaos variable $x(i)$. Sort the bit planes in ascending order by the variable, and obtain the shuffled bit planes $\{SP_1, SP_2...SP_8\}$.

The iteration steps T should be greater than 4, which could ensure the diffusion performance within block. On the other hand, T should not be too great to decrease the efficiency. In this scheme, it is calculated by

$$T = 10 + \mathrm{mod}\left(\lfloor x(9) \times 10^{14} \rfloor, 8\right).$$

Step 2. Encrypt the block with reversible Life-liked CA. The tth iteration can be described as Eq. (9),

$$\begin{cases} SP_8^{t+1} = F(SP_1^t) \oplus F(SP_2^t)... \oplus F(SP_7^t) \oplus SP_8^t \\ SP_i^{t+1} = SP_{i+1}^t \quad 1 \leqslant i < 8 \end{cases}. \tag{9}$$

Equation (9) is an expanding form of Eq. (5), an eighth-order reversible CA. In this scheme, $F(\cdot)$ adopts Life-liked CA rule $B1357/S02468$. Iterate Eq. (9) T times, the encrypted block can be obtained.

Similarly, the decryption equation is described as

$$\begin{cases} SP_1^{t+1} = F(SP_8^t) \oplus F(SP_7^t)... \oplus F(SP_2^t) \oplus SP_1^t \\ SP_i^{t+1} = SP_{i-1}^t \quad 1 < i \leqslant 8 \end{cases}.$$

Step 3. Block-to-block diffusion. Each pixel of the block XORs with the corresponding one in the previous encrypted block. The jth pixel in B_i block is encrypted by

$$B_i(j) = B_i(j) \oplus B_{i-1}(j).$$

Specifically, this step is skipped when encrypting the first block.

Step 4. Repeat Steps 1 to 3 until all the blocks are encrypted.

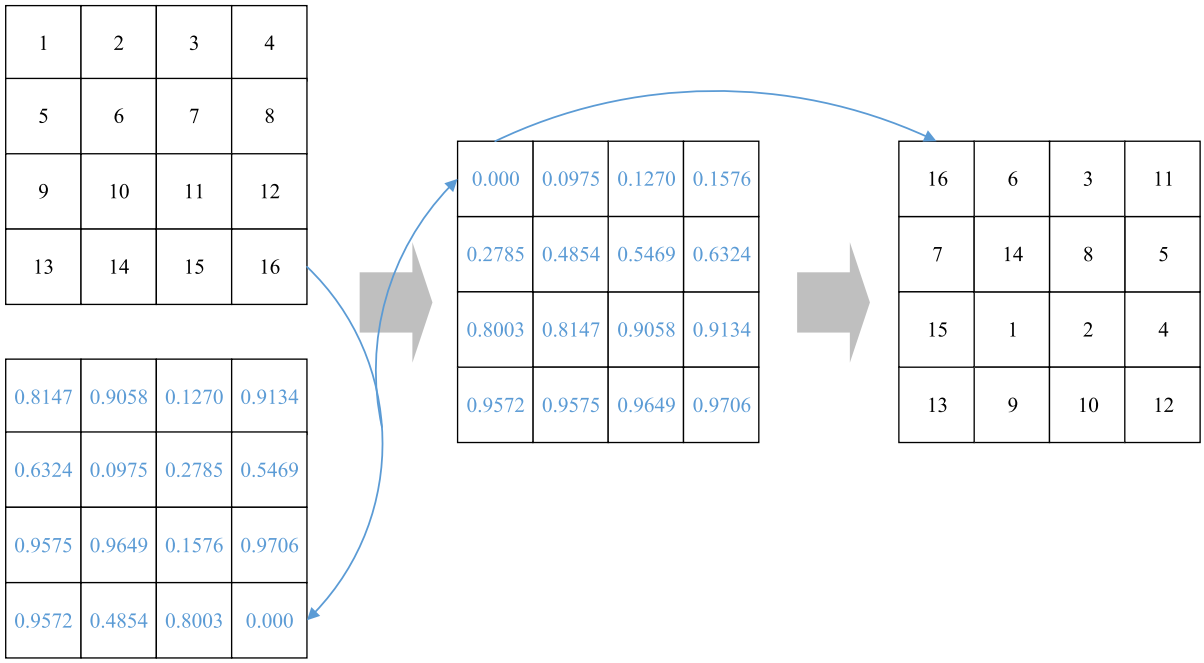

Fig. 6 A simple example of permutation method

4 Theoretical analysis

In this section, we analyze the efficiency of our proposed scheme in theory. In addition, this algorithm can expand to many image types.

Due to different hardware and software conditions, analyzing the computation complexity can be more proper than simply comparing encryption time. When encrypting an image with size of $M \times N$, 10 chaos variables are need for each block, so the computation complexity of generating chaos variables is $O(10\ \frac{M}{8}\frac{N}{8})$. In permutation phase, quick sort algorithm is recommended, so the ideal computation complexity of this phase is $O(\frac{MN}{64}\log(\frac{MN}{64}))$, which is more efficient than most permutation algorithm. In the diffusion phase, parallel computing can help to improve computation efficiency, for CA iteration can be completed in a constant time. Therefore, the complexity of this phase by parallel computing is $O(14\frac{MN}{64})$. The whole algorithm need 2-round permutation–diffusion; in conclusion, the complexity is $2O(10\ \frac{MN}{64} + \frac{MN}{64}\log(\frac{MN}{64}) + 14\frac{MN}{64})$.

As mentioned above, we proposed algorithm is aimed at encrypting 8-bit gray image, which is one basic type of images. Because in the diffusion phase, the CA configurations is generated from each bit plains, which can be convenient for expanding to different type images. For example, the transforming image elements to structured data based on BP neural network have more image dimension. And X-ray image denoising based on wavelet transform and median filter may have more pixel accuracy. In short, these images use more bits to describe one pixel. A possible expanding method is adopting RCA with more higher order. Taking structured data based on BP neural network as example, if the data has d dimensions, and each dimension has 8 bits, a eighth-order RCA system can be used to encrypt the data. Therefore, the proposed algorithm can be expanded to encrypt many type images.

5 Simulation results

This section presents the simulation results of the image encryption scheme. All the images utilized in the experiment are collected from the USC-SIPI image database.[1]

Figure 7 shows the plain images, cipher images and their histograms. For an ideal cipher image, it should

[1] https://sipi.usc.edu/database/.

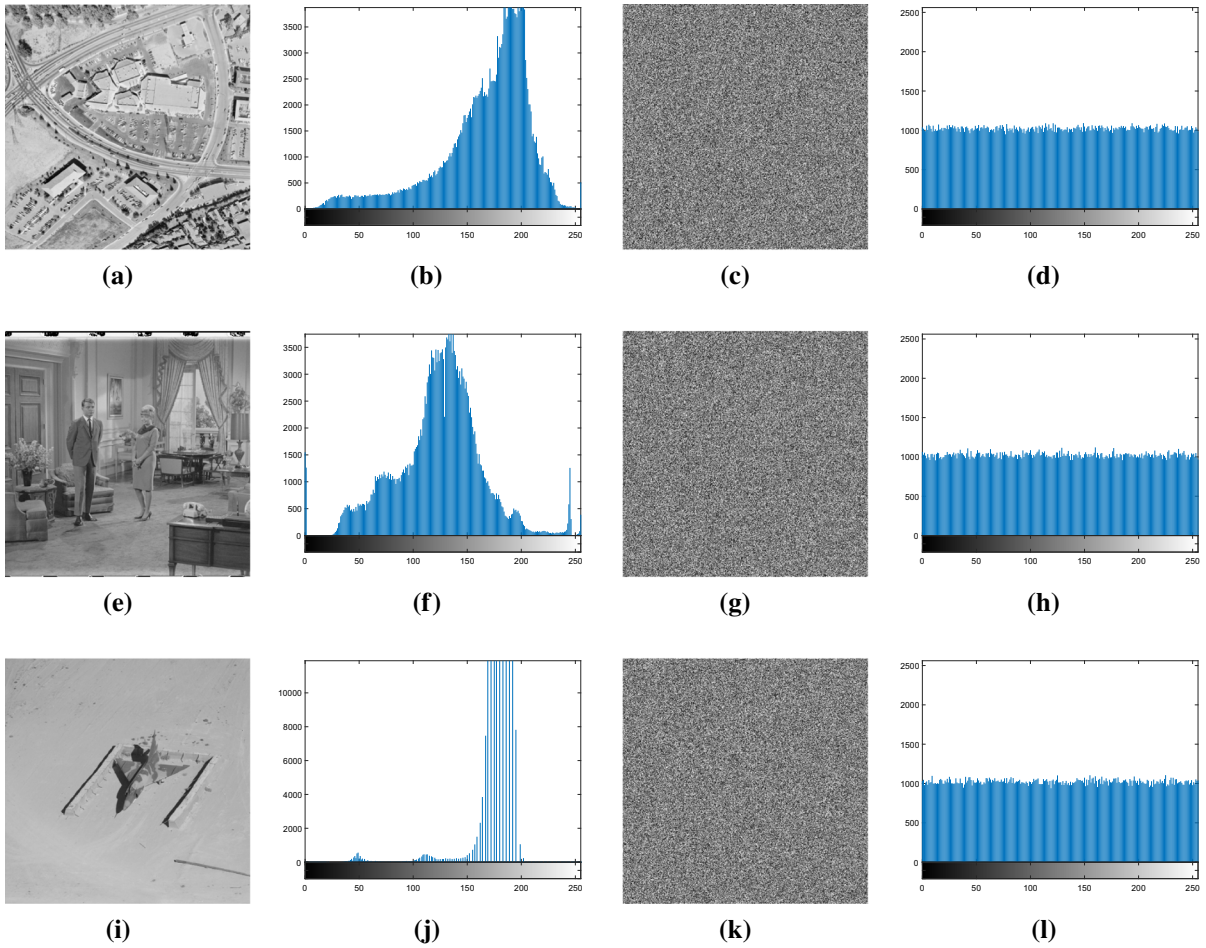

Fig. 7 Simulation results: **a**, **e**, **i** the original plain images; **b**, **f**, **j** histograms of plain images; **c**, **g**, **k** the cipher images; **d**, **h**, **l** the histograms of cipher images

Table 2 Results of Chi-square test

Image	Aerial	Airfield	Baboon	Barb	Boat	Couple	House	Plane	Peppers	Tank
χ^2_{test}	291.1211	237.9199	252.9844	231.3828	289.3418	215.9961	259.5254	254.1230	246.6504	244.2344

be random-like, and its pixels should be uniformly distributed. Therefore, the opponent cannot obtain any useful information by analyzing the distribution of pixels in a cipher image. In addition, the Chi-square test can quantitatively evaluate the uniformity of distribution of pixels in cipher images [30]. The Chi-square is defined as

$$\chi^2 = \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i},$$

where k is the gray level of an image, e_i denotes the expected frequency of each gray level, while o_i denotes the observed frequency. For an 8-bit gray cipher image, when its Chi-square test value $\chi^2_{test} < \chi^2_{255,0.05} = 293$, it can be considered that the image can pass the Chi-square test. Table 2 shows the values of Chi-square test. The results well demonstrate that the all cipher images are random-like, and the pixels in them are uniformly distributed.

6 Security and robustness analysis

This section demonstrates the security and robustness of the proposed image encryption scheme. It is demonstrated that an cryptosystem should have good security performance in terms of adjacent pixel correlation, ability of resisting differential attack, key's sensitivity, local Shannon entropy and plaintext attack resistance.

6.1 Adjacent pixel correlation

Generally, there are strong correlations between the adjacent pixels in a plain image, especially in horizontal, vertical and diagonal direction. The strong correlations could reveal the information in an image; thus, it is essential to weaken the adjacent pixel correlations for a good image encryption scheme. The correlation coefficients between adjacent pixels are calculated by Eqs. (10)–(12), where x and y are values of the two adjacent pixels, $E(x)$ is the mathematical expectation of x, $D(x)$ represents the variance of x and N denotes the number of sampling pixels. When r_{xy} is close to 0, x and y have a weak correlation, while r_{xy} is close to -1 or 1, which indicates a high correlation between x and y.

$$r_{xy} = \frac{E(x - E(x))E(y - E(y))}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{10}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{11}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2. \tag{12}$$

For testing the proposed image encryption scheme, 3000 pairs of adjacent pixels are sampled from the plain image and cipher image. Then the correlation coefficients in the horizontal, vertical and diagonal directions are, respectively, calculated. Figure 8 presents the results of image *Couple* and its cipher image. It is clear that the pixel pairs in plain image almost distribute on the diagonal line or the near area of the phase plane, which indicates that the adjacent pixels in plain image have strong correlation. In contrast, the pixel pairs in cipher image are randomly distributed in the whole phase plane. This means that the correlation between adjacent pixel pairs is very weak in a cipher image. Numerically, Table 3 shows the correlation coefficients of the testing images. As indicated, the proposed scheme can effectively weaken the correlation between adjacent pixels.

Table 3 Correlation comparison between adjacent pixels of plain images and cipher images

Image	Direction	Plain image	Cipher image
Aerial	Horizontal	0.851460	− 0.001197
	Vertical	0.896130	− 0.012249
	Diagonal	0.786035	0.003010
Airfield	Horizontal	0.943441	0.001180
	Vertical	0.940430	0.002668
	Diagonal	0.911838	− 0.009909
Baboon	Horizontal	0.760645	− 0.000114
	Vertical	0.866799	0.004721
	Diagonal	0.709949	0.000822
Barb	Horizontal	0.956620	0.000330
	Vertical	0.860241	0.000477
	Diagonal	0.879735	− 0.006834
Boat	Horizontal	0.979420	− 0.006870
	Vertical	0.952837	− 0.000478
	Diagonal	0.939454	0.000972
Couple	Horizontal	0.896929	0.005044
	Vertical	0.929593	0.000148
	Diagonal	0.841376	0.004560
House	Horizontal	0.974025	− 0.007200
	Vertical	0.974025	− 0.000892
	Diagonal	0.974025	− 0.008677
Plane	Horizontal	0.948498	− 0.002385
	Vertical	0.945446	0.001788
	Diagonal	0.936010	0.002786
Tank	Horizontal	0.929263	− 0.003291
	Vertical	0.943199	0.003372
	Diagonal	0.910912	− 0.004666

6.2 Resistance against differential attack

Differential attack is a popular model to crack an encryption scheme. It refers to extract clues by analyzing the difference between two cipher images which are encrypted by two plain image with only one-bit difference between them, then the opponent try to establish a mapping relationship between plain image and cipher image. To resist against differential attack, the encryption scheme should have certain diffusion performance. The performance means that any slight difference in

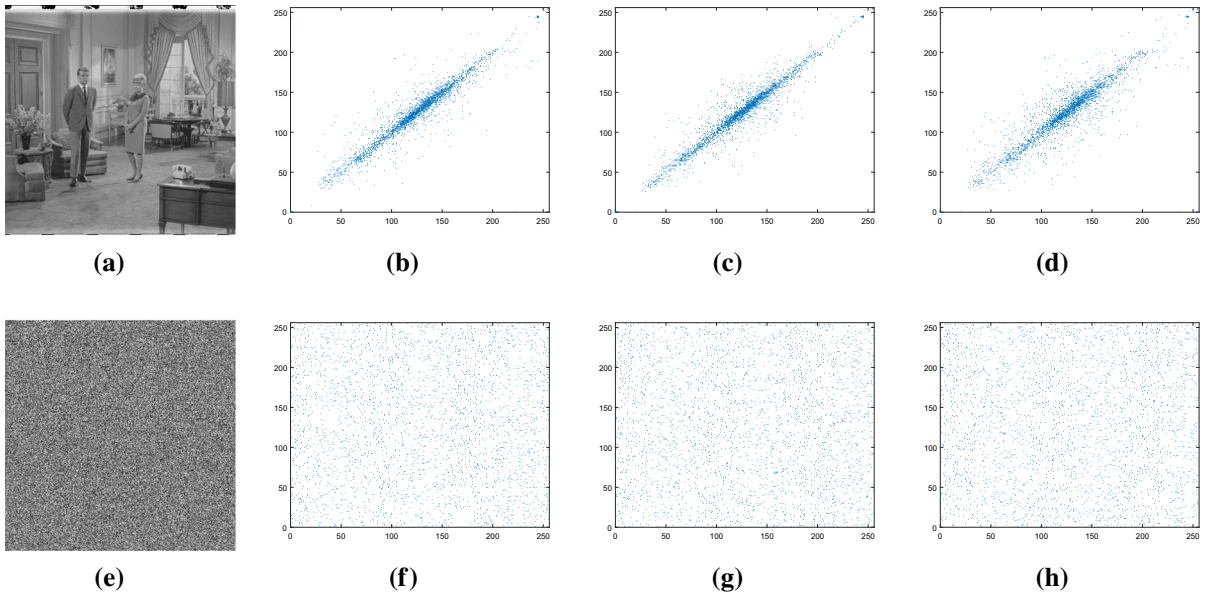

Fig. 8 Correlation plot analysis: **a**, **e** the plain image and cipher image; **b**, **f** adjacent pixel correlation in horizontal; **c**, **g** adjacent pixel correlation in vertical; **d**, **h** adjacent pixel correlation in diagonal

plain image should account for totally different ciphertext.

The ability of an image encryption scheme to resist differential attack is quantitatively estimated by the number of pixel change rate (NPCR) and unified average changed intensity (UACI). These two values are calculated by Eqs. (13)–(15).

$$NPCR(C_1, C_2) = \sum_i \sum_j \frac{D(i,j)}{M \times N} \times 100\%, \tag{13}$$

$$UACI(C_1, C_2) = \frac{1}{M \times N} \times \sum_i \sum_j \frac{|C_1(i,j) - C_2(i,j)|}{L-1} \times 100\%. \tag{14}$$

$$D(i,j) = \begin{cases} 0 \; if \; C_1(i,j) = C_2(i,j) \\ 1 \; if \; C_1(i,j) \neq C_2(i,j) \end{cases}. \tag{15}$$

In these equations, C_1 and C_2 represent the ciphertexts whose corresponding plain images have only one-bit difference, M and N are the width and height of the image, and L denotes gray level of the image.

When the cipher images obtained by differential attack can pass the NPCR and UACI test, it is indicated that the two cipher images are randomly irrelevant ones; thus, the opponent cannot get any useful information from this differential attack. Furthermore, the NPCR and UACI critical values were argued in [33]. For NPCR test, there is a threshold $\mathcal{N}_\alpha^*$ related to a significance level α. When the result is greater than the threshold, it is considered to pass the NPCR test. Similarly, for UACI test, there is an interval $(\mathcal{U}_\alpha^-, \mathcal{U}_\alpha^+)$. Only in the case that the results fail into this interval, it can pass the UACI test. In [33], Wu et al. mathematically give the critical value, $\mathcal{U}_\alpha^-$, $\mathcal{U}_\alpha^+$ and $\mathcal{N}_\alpha^*$ for different size images, when the significance level α equals to 0.05, 0.01 and 0.001, respectively. In this experiment, the size of images is 512 × 512, and gray level is 256. When α=0.05, $\mathcal{N}_\alpha^*$ = 99.5893% and $(\mathcal{U}_\alpha^-, \mathcal{U}_\alpha^+)$ = (33.3730, 33.5541%).

First, one-bit difference was elaborately designed at the right bottom of all the test plain images. In addition, for confirming the influence of different pixels located at other positions, in image *Couple*, four other random coordinates are selected. The four different pixels are, respectively, located at (146, 65), (159, 51), (511, 263), (2, 259), and corresponding images are named as Couple1, Couple2, Couple3 and Couple4. Table 4 shows the NPCR and UACI test results of the proposed scheme, and all the results indicates that the algorithm can pass the NPCR and UACI tests. This means the scheme has sufficient ability to resist against the differential attack.

Table 4 Performance of resisting differential attack

Images	NPCR (%)	UACI (%)
Aerial	99.5998	33.5225
Airfield	99.6178	33.4814
Baboon	99.6037	33.4308
Barb	99.6063	33.4624
Boat	99.6014	33.4308
Couple	99.6071	33.4979
House	99.6243	33.4392
Plane	99.6326	33.4512
Peppers	99.6040	33.4170
Tank	99.6262	33.4559
Couple1	99.6143	33.5348
Couple2	99.6067	33.4846
Couple3	99.6071	33.4830
Couple4	99.6025	33.3771

6.3 Secret key analysis

An encryption scheme should have sufficient key space and should be extremely sensitive to the secret keys. The key sensitivity means that in both encryption and decryption phase, even 1 one-bit difference in secret key should bring about significant different results in the encryption or decryption procedures. For key sensitivity evaluation, we first select a secret key K_0 and then randomly select a bit to reverse, so four keys $K_1 ... K_4$ can be obtained. The involved secret keys are as follows.

- $K_0 = EFC796D47FDFFFE9AB7DF3DFFF3CE7AFDEFEFC6977757FC9DA69D93F4D76FC7F$.
- $K_1 = EFC796D47FDFFFE9AB7DF3DFFF3CE7ABDEFEFC6977757FC9DA69D93F4D76FC7F$(the 125th bit changed).
- $K_2 = EFC796D47FDFFFEBAB7DF3DFFF3CE7AFDEFEFC6977757FC9DA69D93F4D76FC7F$(the 62nd bit changed).
- $K_3 = EFC796D47FDFFFE1AB7DF3DFFF3CE7AFDEFEFC6977757FC9DA69D93F4D76FC7F$(the 60th bit changed).
- $K_4 = EFC796D47FDFFFE9AB7DF3DFFF38E7AFDEFEFC6977757FC9DA69D93F4D76FC7F$(the 109th bit changed).

Because the cipher images are random-like ones, in is demonstrated in [34] that the NPCR and UACI can be used to evaluate the difference between two cipher images encrypted with two secret keys. Figure 9 shows the key sensitivity in the image encryption phase, and Table 5 presents the NPCR and UACI values. The results well demonstrate that in the encryption phase, a slight change in secret key could account for enormous difference.

In the decryption phase, these keys are utilized to decrypt the cipher image, and only the correct secret key can recover the original plain image, while the other keys even with only one-bit difference is unable to recover any useful information of the plain image. Because the original image is not a random one, NPCR and UACI test are not suitable in this phase. Therefore, calculating the incorrectly decrypted pixels in decrypted images can evaluate the sensitivity of secret key in the decryption phase. Table 6 and Fig. 10 demonstrate the key sensitivity in decryption phase. The result indicates the high key sensitivity of the proposed encryption scheme in the decryption procedure.

6.4 Local Shannon entropy

The pixels in a cipher image should be random and uniformly distributed; thus, it can resist various attacks. Generally, Shannon entropy is adopted to evaluate the randomness of image pixels [39]. For an image information source S, its entropy is defined by Eq. (16), where L is the gray level and $P(s_i)$ represents the probability of symbol s_i .

$$H(S) = -\sum_{i=0}^{L} P(s_i)\log_2 P(s_i). \tag{16}$$

For an 8-bit gray cipher image, its ideal entropy is 8. Due to different secret keys, original plain images, the entropy of cipher image would vary in a certain range. Generally, if the entropy of cipher image is higher than 7.99, it can be considered secure enough to defeat statistical analysis. For example, in studies [31,32], their entropy test results are, respectively, higher than 7.99 and 7.999.

Wu et al. further indicate that the global entropy is not appropriate to describe the randomness in some cases [35]. Then they propose local Shannon entropy

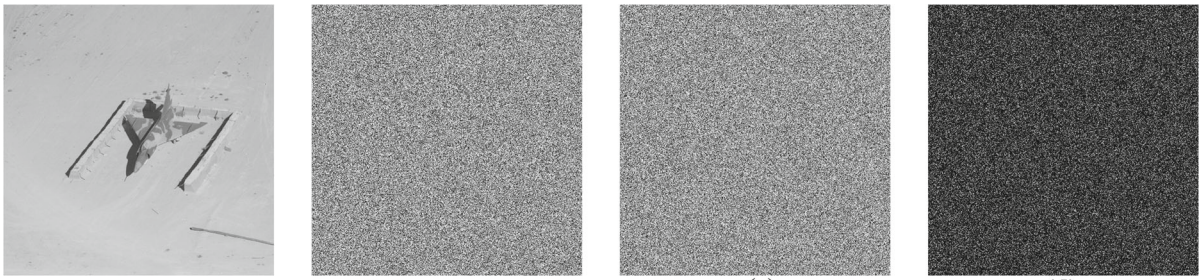

Fig. 9 Key sensitivity in encryption phase: **a** plain image Plane; **b** cipher image with K_0; **c** cipher image with K_1; **d** the difference between **b** and **c**

Table 5 Key sensitivity perform in the encryption phase

Key	NPCR (%)	UACI (%)
Key1	99.6098	33.4402
Key2	99.6147	33.4440
Key3	99.5907	33.4626
Key4	99.6227	33.4448

Table 6 Error rate comparing with Plane

Key	Error rate (%)
Key1	99.6140
Key2	99.5930
Key3	99.6159
Key4	99.6342

which can be described as

$$\overline{H_{k,T_B}} = \sum_{i=1}^{k} \frac{H(S_i)}{k},$$

where S_i is a selected block in test image, k is the number of the blocks and T_B represents the number of pixels in each block, respectively. Local Shannon entropy is an average result of these non-overlapping selected blocks.

Following Wu et al. [35], $k = 30$, $T_B = 1936$ and the significance level $\alpha = 0.05$, an 8-bit image is considered to pass the test when its local Shannon value falls into the interval $(h^*_{left}, h^*_{right}) = (7.901901, 7.903037)$. Table 7 lists the local Shannon test results of our proposal. It is clear that the cipher images have high global entropy, which are higher than 7.999, and can pass the local Shannon test. Therefore, the proposed scheme can encrypt image with sufficient randomness.

6.5 Plaintext attack analysis

The classic confusion–diffusion structure has inherent weakness, and the drawback has been utilized by many researchers to crack the whole encryption scheme. Chen et al. [36] proposed that diffusion phase adopting XOR or modular addition has intrinsic drawback, and the opponent can obtain the equivalent chaos sequence by plaintext analysis and further crack the whole system.

In this scheme, we adopt block encryption in block-wise and reversible cellular automaton within block, which has been demonstrated having complex chaotic behavior. Because the RCA is much more complex, the opponent cannot analyze the mapping relation between plaintext and ciphertext, even though the opponent construct some elaborately designed plain image with identical pixels. Therefore, the drawback of confusion–diffusion structure is significantly addressed.

Generally, the opponents select some special plain images to prove the insecurity of encryption schemes. Many encryption algorithm when dealing with such as full black and full white images may behave insecure. Figure 11 shows the results that our proposed algorithm encrypting full black and white image with size of 512×512. Table 8 further shows information entropy and adjacent correlations of cipher images of full black and white. It can be concluded that our proposed algorithm can defeat plaintext analysis.

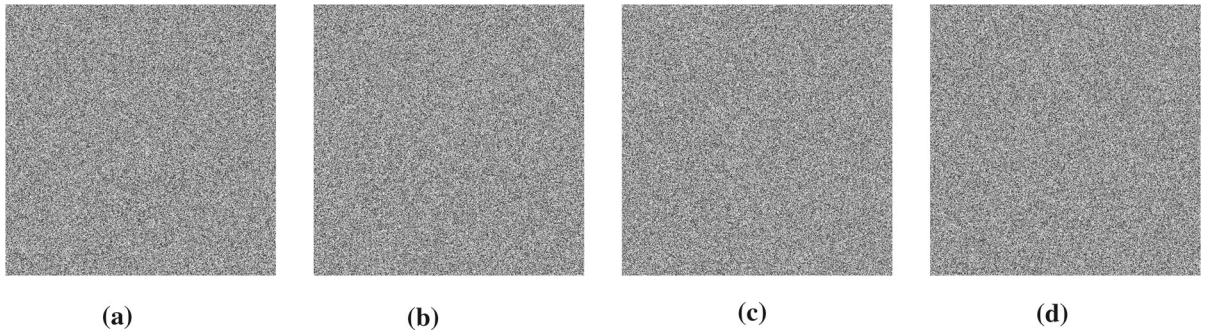

Fig. 10 Key sensitivity in decryption phase: **a** decipher image with K_1; **b** decipher image with K_2; **c** decipher image with K_3; **d** decipher image with K_4

Table 7 Entropy and local entropy test results

Image	Plain image entropy	Cipher image entropy	Cipher image local entropy
Aerial	7.185637955	7.999422409	7.901901730
Airfield	7.452672714	7.999331658	7.902547857
Baboon	7.357949076	7.999296012	7.902492905
Barb	7.466426194	7.999206143	7.902762629
Boat	7.072868435	7.999155394	7.902619948
Couple	7.201007960	7.999289489	7.902019703
House	7.477779634	7.999236791	7.902766305
Plane	4.004499445	7.999326725	7.902910780
Peppers	7.571477564	7.999262255	7.903009686
Tank	5.495739989	7.999317713	7.902033498

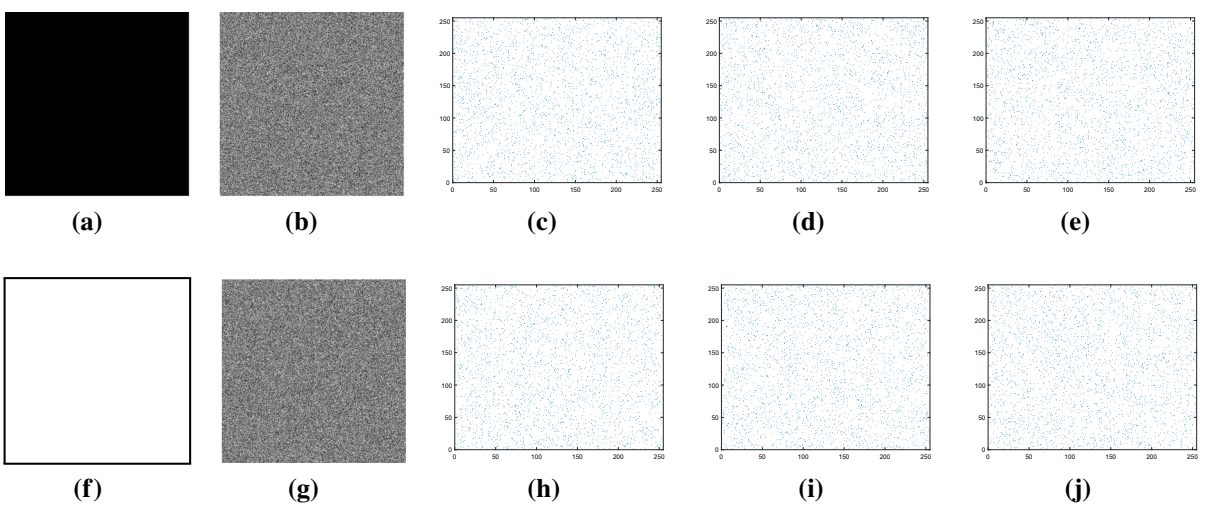

Fig. 11 Experimental result of plaintext analysis. **a**, **f** full black and white images; **b**, **g** corresponding cipher images of **a**, **f**; **c**, **h** adjacent pixel correlation in horizontal; **d**, **i** adjacent pixel correlation in vertical; **e**, **j** adjacent pixel correlation in diagonal

Table 8 Adjacent pixel correlation and information entropy of cipher images (full black and white)

Image	Pixel correlation			entropy
	Horizontal	Vertical	Diagonal	
Full black	0.000592	−0.002256	−0.000705	7.999341
Full white	−0.007721	−0.003100	−0.001899	7.999413

6.6 Robustness analysis

When the cipher images are transmitted in public networks, they may loss data or be blurred by noise. The ability of defeating noise and data loss is also significant for an encryption scheme. Because the scheme utilizes block encryption, the robustness of this scheme is greatly improved. When a pixel error occurs in a cipher image, this error will make the block containing this pixel fail to recover in decryption phase; in addition, it may affect the next block by diffusion property. However, the error could be limited within the block, and the rest pixels in the cipher image could be correctly decrypted.

To quantitatively demonstrate the ability of the encryption scheme in resisting data loss and random noise, the peak signal-to-noise ratio (PSNR) and mean square error (MSE) are adopted to test the difference between the original plain image and the decrypted image. The MSE and PSNR are, respectively, defined as Eqs. (17)–(18).

$$MSE(P_1, P_2) = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (P_1(i, j) - P_2(i, j))^2, \quad (17)$$

$$PSNR(P_1, P_2) = 10 \times \log \left(\frac{(L-1)^2}{MSE(P_1, P_2)} \right). \quad (18)$$

In both formulas, P_1 and P_2 are the original plain image and decrypted image, L is the gray level of image, and M, N represent the image size, respectively.

The image *Boat* is utilized as the example. We adopt salt and pepper noise to interfere cipher image, which means the pixel value influenced by noise would randomly be 0 or 255, and it is a common noise in image transmission. Table 9 lists the MSE and PSNR test results of the cipher image with different percentages of data loss and noise. Figure 12 presents the results of this scheme in defending data loss and noise. Figure 13 shows the PSNR performance of different data loss.

Table 9 Resistance of occlusion attack

Data loss and noise attack	MSE	PSNR
5% data loss	82.4408	28.9694
10% data loss	278.9599	23.6754
20% data loss	829.8883	18.9406
0.05% salt and pepper noise	946.1590	18.3712
0.1% salt and pepper noise	1843.6348	15.4741
0.15% salt and pepper noise	2645.7197	13.9054

The scatters represent original experimental results, and the red line is the fitted curve. It is clear that even though the cipher images loss some data or are blurred by noise, the decryption phase can still recover the original ones with high visual perception. In addition, it is obvious that the proposed algorithm have good PSNR performance when loss rate less than 25%, and the performance on data loss is better than random noise.

6.7 Comparison analysis

To further analyze the performance of our proposed scheme, we compare it with other advanced algorithms [37,38]. For conducting a fair comparison, we adopt the results reported in these references papers. Table 10 compares the proposed scheme and other encryption algorithms by NPCR and UACI tests. The numbers in underline are the results fail to pass the UPCR and UACI test. The bold font numbers are the results fail to pass the test; thus, our proposed scheme has more stable performance comparing with these advanced methods.

Table 11 compares the adjacent pixel correlation with some recent algorithms. The most closed to 0 result is marked in bold font, and our proposed method performed best on many tests. So this proposed algorithm have good ability to weaken the correlation of adjacent pixels among these advanced methods.

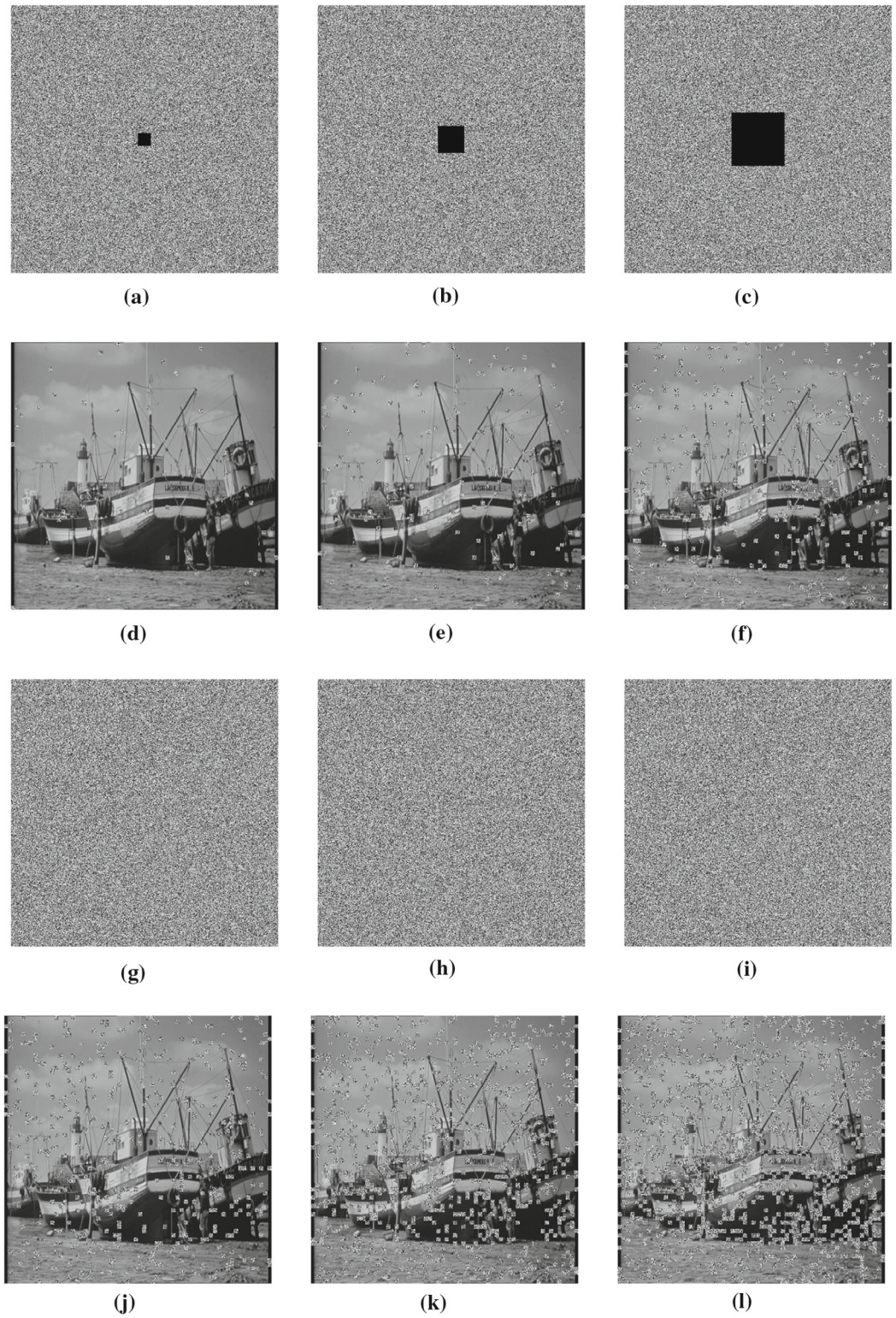

Fig. 12 Robustness performance: **a–c** cipher images with 5, 10 and 20% data loss; **d–f** corresponding decipher images of **a–c**; **g–i** cipher images with 0.05, 0.1 and 0.15% salt and pepper noise; **j–l** corresponding decipher image of **g–i**

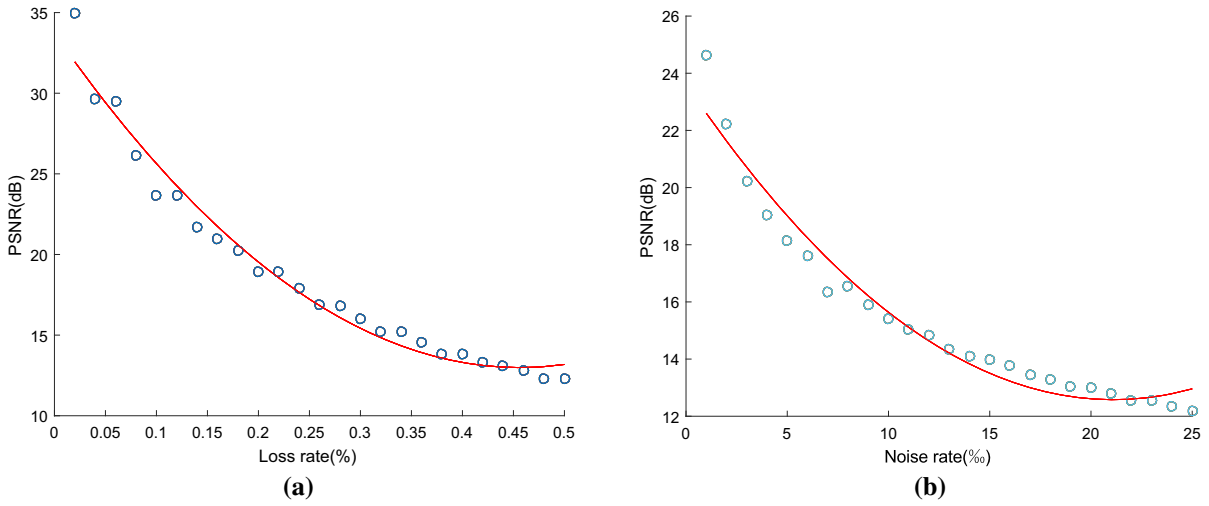

Fig. 13 PSNR performance on different degree attacks: **a** PSNR performance on data loss; **b** PSNR performance on salt and pepper noise

Table 10 Comparison with other methods of NPCR and UACI

Image	Peppers		Baboon		Boat	
	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)	NPCR (%)	UACI (%)
Proposed	99.6040	33.4170	99.6037	33.4308	99.6014	33.4308
Ref. [37]	99.787	33.621	99.881	33.415	99.625	33.671
Ref. [38]	99.5703	33.4706	99.6080	33.4955	99.5975	33.4602
Ref. [39]	99.7136	33.5413	99.6234	33.4156	99.6194	33.5562
Ref. [40]	99.6141	33.4812	–	–	99.6104	33.4363

Table 11 Comparison with other methods of adjacent pixel correlation

Image	Peppers			Baboon			Boat		
	H	V	D	H	V	D	H	V	D
Proposed	−0.0029	0.0038	**−0.0009**	**−0.0001**	0.0047	**0.0008**	−0.0069	−0.0054	**0.0010**
Ref. [38]	0.0034	**0.0014**	−0.0018	0.0012	0.0014	0.0039	0.0048	**−0.0022**	0.0013
Ref. [41]	0.0252	0.0248	−0.0072	0.0190	−0.0302	−0.0648	–	–	–
Ref. [42]	**0.0006**	0.0038	0.0010	0.0029	0.0033	0.0062	**0.0003**	0.0034	0.0011
Ref. [43]	0.0023	−0.0014	0.0046	0.0066	**−0.0001**	0.0105	0.0038	0.0116	−0.0118

7 Conclusion

In this paper, we propose an image encryption scheme based on block encryption, 2D-LSCM chaos map and reversible Life-liked CA. First, we find that the reversible Life-liked CA lack of robustness, which can make the encryption algorithm cannot defeat data loss or random noise. To overcome this loophole, the block encryption not only maintain the diffusion performance of Life-liked CA, but increase the robustness of whole system. In addition, we demonstrate that a proper designed permutation method can increase the efficiency of whole permutation–diffusion structure. The permutation method needs to move the last pixel to the first one. Thus, we propose an efficient permutation method, which can ensure sufficient diffusion perfor-

mance with only 2 permutation–diffusion rounds. The simulation results presents that the cipher images are noise-like with uniform distributed pixels. Chi-square test and histogram of cipher image indicate the uniform distribution of pixel value in cipher image. The security analysis demonstrates that the proposed encryption scheme can greatly weaken the pixel correlation, resist differential attack and plaintext analysis. In addition, the proposed scheme also have sufficient key sensitivity and pixel randomness. The results of comparison with other advanced algorithms show that our proposal has good secure performance among these algorithms. In particular, it has more stable performance in resisting differential attack. The robustness analysis demonstrates that the cryptosystem can efficiently resist data loss and random noise.

Acknowledgements This work is funded by the National Natural Science Foundation of China (No. 62171114), the Fundamental Research Funds for the Central Universities (No. N2224001-7) and the National Key R&D Program of China (No. 2021YFF0306405).

Funding This work is supported by the National Natural Science Foundation of China (No. 62171114), the Fundamental Research Funds for the Central Universities (No. N2224001-7) and the National Key R&D Program of China (No. 2021YFF0306405).

Data availability In this paper, the original plain images are from the USC-SIPI Image Database. We mainly use the images with size of 512 × 512. And the database is available in https://sipi.usc.edu/database/. The proposed encryption scheme and some related analysis are implemented in C++, and some other tests, like the simulation of RCA, are implemented in MATLAB 2016a. These source code are open and can be found in: https://github.com/NEUboy/encode_RCA.git.

Declarations

Conflict of interest The authors declare that they don't have conflict of interest.

References

1. May, R.M.: The Theory of Chaotic Attractors, pp. 85–93. Springer (2004)
2. Zhou, Y., Bao, L., Chen, C.P.: A new 1d chaotic system for image encryption. Signal Process. **97**, 172–182 (2014)
3. Li, C., Luo, G., Qin, K., Li, C.: An image encryption scheme based on chaotic tent map. Nonlinear Dyn. **87**(1), 127–133 (2017)
4. Li, C., Li, S., Asim, M., Nunez, J., Alvarez, G., Chen, G.: On the security defects of an image encryption scheme. Image Vis. Comput. **27**(9), 1371–1381 (2009)
5. Hua, Z., Jin, F., Xu, B., Huang, H.: 2D logistic-sine-coupling map for image encryption. Signal Process. **149**, 148–161 (2018)
6. Srivastava, A.N., Das, S.: Detection and prognostics on low-dimensional systems. IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.) **39**(1), 44–54 (2008)
7. Lin, L., Shen, M., So, H.-C., Chang, C.: Convergence analysis for initial condition estimation in coupled map lattice systems. IEEE Trans. Signal Process. **60**(8), 4426–4432 (2012)
8. Xie, E.Y., Li, C., Yu, S., Lü, J.: On the cryptanalysis of Fridrich's chaotic image encryption scheme. Signal Process. **132**, 150–154 (2017)
9. Li, C., Liu, Y., Xie, T., Chen, M.Z.: Breaking a novel image encryption scheme based on improved hyperchaotic sequences. Nonlinear Dyn. **73**(3), 2083–2089 (2013)
10. Wolfram, S.: Conference on the Theory and Application of Cryptographic Techniques, pp. 429–432. Springer (1985)
11. Tomassini, M., Perrenoud, M.: Cryptography with cellular automata. Appl. Soft Comput. **1**(2), 151–160 (2001)
12. Machicao, J., Marco, A.G., Bruno, O.M.: Chaotic encryption method based on life-like cellular automata. Expert Syst. Appl. **39**(16), 12626–12635 (2012)
13. Alexan, W., ElBeltagy, M., Aboshousha, A.: In: 2021 International Conference on Microelectronics (ICM), pp. 34–39. IEEE (2021)
14. Jin, J.: An image encryption based on elementary cellular automata. Opt. Lasers Eng. **50**(12), 1836–1843 (2012)
15. Abdo, A., Lian, S., Ismail, I.A., Amin, M., Diab, H.: A cryptosystem based on elementary cellular automata. Commun. Nonlinear Sci. Numer. Simul. **18**(1), 136–147 (2013)
16. Conway, J., et al.: The game of life. Sci. Am. **223**(4), 4 (1970)
17. Ping, P., Wu, J., Mao, Y., Xu, F., Fan, J.: Design of image cipher using life-like cellular automata and chaotic map. Signal Process. **150**, 233–247 (2018)
18. Abu Dalhoum, A.L., Mahafzah, B.A., Awwad, A.A., Aldhamari, I., Ortega, A., Alfonseca, M.: Digital image scrambling using 2D cellular automata. IEEE MultiMedia **19**(4), 28–36 (2012). https://doi.org/10.1109/MMUL.2011.54
19. Fridrich, J.: Symmetric ciphers based on two-dimensional chaotic maps. Int. J. Bifurc. Chaos **8**(06), 1259–1284 (1998)
20. Chen, J., Chen, L., Zhou, Y.: Universal chosen-ciphertext attack for a family of image encryption schemes. IEEE Trans. Multimedia **23**, 2372–2385 (2020)
21. Wang, Y., Wong, K.-W., Liao, X., Xiang, T., Chen, G.: A chaos-based image encryption algorithm with variable control parameters. Chaos, Solitons Fractals **41**(4), 1773–1783 (2009)
22. Souyah, A., Faraoun, K.M.: An image encryption scheme combining chaos-memory cellular automata and weighted histogram. Nonlinear Dyn. **86**(1), 639–653 (2016)
23. Song, X., Shi, M., Zhou, Y., Wang, E.: An block image encryption algorithm based on reversible cellular automata. In: 2021 IEEE 21st International Conference on Communication Technology (ICCT), pp. 1167–1172. IEEE (2021)
24. Briggs, K.: An improved method for estimating Liapunov exponents of chaotic time series. Phys. Lett. A **151**(1–2), 27–32 (1990)
25. Grassberger, P., Procaccia, I.: Estimation of the Kolmogorov entropy from a chaotic signal. Phys. Rev. A **28**(4), 2591 (1983)

26. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E.: A statistical test suite for random and pseudorandom number generators for cryptographic applications. Tech. rep., Booz-Allen and Hamilton Inc., McLean (2001)
27. Zhang, W., Wong, K.-W., Yu, H., Zhu, Z.-L.: A symmetric color image encryption algorithm using the intrinsic features of bit distributions. Commun. Nonlinear Sci. Numer. Simul. **18**(3), 584–600 (2013)
28. Wong, K.-W., Kwok, B.S.-H., Law, W.-S.: A fast image encryption scheme based on chaotic standard map. Phys. Lett. A **372**(15), 2645–2652 (2008)
29. Trujillo-Toledo, D.A., López-Bonilla, O.R., García-Guerrero, E.E., Tlelo-Cuautle, E., López-Mancilla, D., Guillén-Fernández, O., Inzunza-González, E.: Real-time RGB image encryption for IoT applications using enhanced sequences from chaotic maps. Chaos, Solitons Fractals **153**, 111506 (2021)
30. Kwok, H., Tang, W.K.: A fast image encryption system based on chaotic maps with finite precision representation. Chaos, Solitons Fractals **32**(4), 1518–1529 (2007)
31. Lin, C.H., Hu, G.H., Chen, J.S., Yan, J.J., Tang, K.H.: Novel design of cryptosystems for video/audio streaming via dynamic synchronized chaos-based random keys, . Multimedia Syst. 28, 1793–1808. https://doi.org/10.1007/s00530-022-00950-6
32. Guillén-Fernández, O., Tlelo-Cuautle, E., de la Fraga, Luis G., Sandoval-Ibarra, Y., Nuñez-Perez, J.-C.: An image encryption scheme synchronizing optimized chaotic systems implemented on raspberry pis. Mathematics 11(10), 1907 (2022)
33. Wu, Y., Noonan, J.P., Agaian, S., et al.: NPCR and UACI randomness tests for image encryption. Cyber J.: Multidiscipl. J. Sci. Technol., J. Sel. Areas Telecommun. (JSAT) **1**(2), 31–38 (2011)
34. Alvarez, G., Li, S.: Some basic cryptographic requirements for chaos-based cryptosystems. International Journal of Bifurcation and Chaos **16**(08), 2129–2151 (2006)
35. Wu, Y., Zhou, Y., Saveriades, G., Agaian, S., Noonan, J.P., Natarajan, P.: Local Shannon entropy measure with statistical tests for image randomness. Inf. Sci. **222**, 323–342 (2013)
36. Chen, J., Chen, L., Zhou, Y.: Cryptanalysis of image ciphers with permutation-substitution network and chaos. IEEE Trans. Circuits Syst. Video Technol. **31**(6), 2494–2508 (2020)
37. Pourasad, Y., Ranjbarzadeh, R., Mardani, A.: A new algorithm for digital image encryption based on chaos theory. Entropy **23**(3), 341 (2021)
38. Wang, X., Xue, W., An, J.: Image encryption algorithm based on tent-dynamics coupled map lattices and diffusion of household. Chaos, Solitons Fractals **141**, 110309 (2020)
39. Yasser, I., Khalifa, F., Mohamed, M.A., Samrah, A.S.: A new image encryption scheme based on hybrid chaotic maps. Multimedia Tools Appl. **80**(2), 2753–2772 (2021)
40. Wang, X., Su, Y.: Image encryption based on compressed sensing and DNA encoding. Signal Process. Image Commun. **95**, 116246 (2021)
41. Zhang, Y.: The fast image encryption algorithm based on lifting scheme and chaos. Inf. Sci. **520**, 177–194 (2020)
42. Wu, J., Liao, X., Yang, B.: Image encryption using 2D Hénon–Sine map and DNA approach. Signal Process. **153**, 11–23 (2018)
43. Shahna, K., Mohamed, A.: A novel image encryption scheme using both pixel level and bit level permutation with chaotic map. Appl. Soft Comput. **90**, 106162 (2020)