Signal Processing: Image Communication 52 (2017) 6–19

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

An image encryption algorithm based on the memristive hyperchaotic system, cellular automata and DNA sequence operations

Xiuli Chai[a,b,*], Zhihua Gan[c], Kang Yang[a], Yiran Chen[b], Xianxing Liu[a]

[a] Institute of Image Processing and Pattern Recognition, Henan University, Kaifeng 475004, China
[b] Department of Electrical and Computer Engineering, Duke University, Durham, NC 27708, United States
[c] School of Software, Henan University, Kaifeng 475004, China

ARTICLE INFO

Keywords:
Image encryption
Memristive hyperchaotic system
Cellular automata (CA)
DNA sequence
SHA 256

ABSTRACT

A novel image encryption scheme employing the memristive hyperchaotic system, cellular automata (CA) and DNA sequence operations is presented, which consists of diffusion process. SHA 256 hash function is used to give the secret key and compute the initial values of the chaotic system. Moreover, a dynamic DNA encoding scheme is introduced. Two DNA rule matrices for encoding the plain image and two-dimensional (2D) CA are generated from chaotic sequences, and they are controlled by the plain image, so that there are different DNA encoding rules for different original image. Besides, we manipulate block diffusion encryption method to the plain image in order to save time. The previous diffused block image and 2D CA are combined to affect the encryption effect of the current block image. Among them, 2D CA is updated by the local rule computed from the previous diffused sub image, and its initial configuration is determined by the chaotic sequences. Simulation results and security analyses both confirm that the proposed image encryption scheme not only demonstrates extraordinary encryption performance, but also resists various attacks. It can be applied in secure image and video communication fields.

1. Introduction

Along with the rapid development of network and computer technology, an increasing number of images are transmitting and storing over internet. In view of bulky data capacity, high redundancy and strong correlations among adjacent pixels of the digital images, the typical encryption schemes, such as DES, AES, RSA are not suitable, and new encryption algorithms based on optical transform [1], DNA sequence operations [2], wave motion [3], Brownian motion [4], cellular automata [5], compressive sensing [6] and chaotic systems [7] are introduced. Among them, chaos-based image encryption method is receiving more and more interests, for chaotic system has many excellent features, for example, sensitive dependence on initial conditions, pseudo-randomness, ergodicity and reproduction. Therefore, many image encryption algorithms based on chaotic systems have been put forward in recent years [7–12].

In 1998, Fridrich [13] proposed the first general architecture for chaos-based image cipher, and it was composed of permutation and diffusion. Permutation is used to break the correlation between adjacent pixels of the digital images by shuffling the positions of the image pixels, and the histograms are not changed. Diffusion alters the pixel values, which is an indispensable process and permutation can be omitted. In 2000, Yen et al. [14] gave an efficient hierarchical chaotic image encryption (HCIE) algorithm, the plain image was firstly divided into the same size blocks, and then position permutation on intra-block and inter-block level were operated. Recently, Li et al. [15] cracked this algorithm, and found that the security of HCIE against ciphertext-only attack was much lower, and they also cryptanalyzed Fridrich's chaotic image encryption scheme [16]. Benyamin Norouzi et al. [17] introduced a fast image encryption algorithm for color images, two hyperchaotic systems are employed to produce diffusion sequences for encrypting the *R*, *G*, *B* components of the image, the diffusion process can modify the pixel values and remove the strong correlations between adjacent pixels of an image at the same time. Without the permutation, it can run fast with good encryption effect. Besides, they [18] presented a novel image encryption scheme based on hash function and two-round diffusion process, in the first round, an plain image is divided horizontally to an array which is composed of 1024 sections sized 8×8, in the second round, the image is partitioned vertically to the transpose of the obtained array, and the algorithm used the average of the image pixels for encryption. In the paper, the encryption architecture of only diffusion is employed in order to

* Corresponding author at: Institute of Image Processing and Pattern Recognition, Henan University, Kaifeng 475004, China.
E-mail address: chaixiuli@henu.edu.cn (X. Chai).

http://dx.doi.org/10.1016/j.image.2016.12.007
Received 11 August 2016; Received in revised form 14 November 2016; Accepted 22 December 2016
Available online 23 December 2016
0923-5965/

improve the encryption efficiency.

Due to the features of vast parallelism and extraordinary information density, DNA computing has entered in the domain of image encryption. DNA computing consists of some biological operations and algebra operations on DNA sequence, such as DNA addition operations, DNA subtraction operations and DNA XOR operations. Encryption algorithms based on DNA techniques use DNA encoding and DNA computing to encrypt images [19]. For example, in Ref. [20], a novel image encryption algorithm by combining 1D and 2D Logistic maps and DNA addition operation was proposed. Wang [21] et al. presented a novel image encryption scheme using DNA sequence operations, and the spatiotemporal chaos system was used. Recently, Zhang et al. [22] used a new mixed linear-nonlinear coupled map lattices (MLNCML) along with DNA computing to encrypt the images. Guesmi et al. [23] proposed a novel chaos-based image encryption, DNA sequence operation, SHA 256 hash function and Lorenz chaotic system were utilized to strengthen the cryptosystem. In the DNA-based image encryption scheme, the plain image and key image are encoded by DNA encoding rules, and afterwards some DNA operations are conducted on the encoded plain image and key image. At most times, permutation is added in order to improve the security level. There are 8 DNA encoding rules, and 8 DNA addition, subtraction, XOR operations. In many encryption scheme, the encoding rules of the plain image and key image are determined [20,23–26], for example, in Ref. [25], the third encoding rule is adopted and the fourth decoding rule is chosen; sometimes they can be taken as a secret key to enlarge the key space [21,22,27–29], and to sum up, there is the same one encoding rule for all elements of the original image and key image, and encoding rules are independent of the plain image, which reduce the ability of encryption methods to resist statistical attacks, known-plaintext and chosen-plaintext attacks.

Cellular automata (CA), presented by Ulam and John Von Neumann, can produce complex and random patterns out of simple rules. Since Wolfram proposed a CA-based stream cipher [30], CA has been applied in encryption field, and some encryption algorithms based on CA have been reported [32–35]. In image encryption scheme, CA can not only be used for diffusing the image, but also confusing it. Chen et al. [31] introduced an image encryption scheme, the SCAN methodology was employed to manipulate position permutation, and 2D CA was employed to generate pseudo-random number sequences for pixel value substitution. Ping et al. [32] utilized a 2D reversible CA with nonlinear balanced rules to encrypt color images, the plain image was firstly divided into four blocks, the block images was used as the initial configurations of the 2D CA, the confusion operation was performed by the nonlinear rule of CA, and the diffusion one is manipulated by the local interactions among cells. Subsequently, they [33] presented a novel image encryption scheme using reversible and irreversible CA, the second-order reversible CA was responsible for confusion and diffusion processes, and the irreversible CA was to generate pseudo-random key stream sequence. Ref. [34] gave a novel image encryption scheme based on chaos and Langton's ant cellular automaton, CA was used for scrambling the image and PWLCM chaotic map was utilized for diffusion. Recently, a novel image encryption scheme was introduced based on DNA computing, 2D Tinkerbell chaotic map and 1D CA in Ref. [35], 2D chaotic map was utilized to determine the rule numbers of plain images and CA, however, it has some drawbacks, such as the pixel-by-pixel diffusion operation was implemented, and it may cost much time, and the 1D CA has small key space and weak security, which both restricts the security level of the encryption algorithm. In this paper, in order to present a fast and highly secure image encryption algorithm, 2D CA is adopted, the local rule of the 2D CA is determined by the previous diffused block image, and the proposed algorithm can withstand the chosen-plaintext attack; besides, the diffusion process is manipulated by block, and it can improve the encryption speed.

Based on the above analyses, a new chaos-based image encryption combined 2D cellular automata and DNA sequence operations is introduced. Our proposed algorithm has some merits. First, the DNA encoding schemes for the plain image and 2D CA are dynamical, and two DNA rule matrices generated from the plain image and the memristive chaotic system are designed. DNA rule is kept fixed or as a key (varying from 1 to 8) in the past encryption methods, but in our algorithm, every element has its own DNA rule which changes with the plain image, thus our method has high sensitivity with the original image. Moreover, diffusion encryption by block without permutation phase is adopted, in the diffusion process of the current sub-block image, the previous diffused sub-block image is employed to set the local rule of the 2D CA, and then the updated 2D CA and the previous diffused image are used to encrypt the current image. In addition, in order to improve the ability of resisting the known-plaintext and chosen-plaintext attacks, we employ the SHA 256 hash function of the plain image to generate the secret key, and based on it, initial values of the memristive chaotic system are calculated.

This paper is organized as follows. In the next section, we give the preliminary works. Section 3 introduces the encryption scheme. Simulation results are presented in Section 4. Security analyses are drawn in Section 5 and our conclusions are left to the final Section.

2. Preliminary works

2.1. The memristive hyperchaotic system

In 1983, Chua designed the famous third-order autonomous system–Chua's circuit, which can generate rich chaotic phenomena and bifurcations under different parameters. As the simplest electronic circuit, it is comprised of one inductor, one active resistor, one nonlinear resistor and two capacitors. The memristor is the fourth basic circuit element in addition to resistor, capacitor and inductor, which is originally presented by Chua in 1971 [36] and firstly invented by Williams et al. [37] at HP laboratory in 2008. Due to its excellent features, such as non-volatility, nano-size, low power consumption, the memristor has many promising applications in electronics, neural network [38], and it is very suitable for designing chaotic circuits as a nonlinear circuit elements [39].

Employing a magnetic-controlled memristor to substitute the normal resistance of the Chua chaotic circuit, a new four-dimensional memristive chaotic system may be obtained [40]. It can be described by the following equation.

$$\begin{cases} \dot{x} = a[W(w)(y - x) - h(x)] \\ \dot{y} = W(w)(x - y) + z \\ \dot{z} = -by - cz \\ \dot{w} = y - x \end{cases} \tag{1}$$

where,

$$h(x) = m_1 x + 0.5(m_0 - m_1) \times (|x + 1| - |x - 1|) \tag{2}$$

$$W(w) = \frac{d}{2e}||(w + b)| - |(w - b)|| \tag{3}$$

Here, x, y, z and w make up the system states and a、b、c、d、e、m_0、m_1 are system parameters. When $a=30$, $b=36$, $d=2.5$, $e=3.5$, $m_0=-0.5$, $m_1=-0.1$ and $c\in[0.073,0.162]$, the system is hyperchaotic. The attractor is shown in Fig. 1 with $c=0.1$ and Fig. 2 illustrates the change diagram of the state variables within 20,000 iterations. When the initial values of the system is $(x(0), y(0), z(0), w(0))^T=(0, 0.01, 0.01, 0)^T$, and Lyapunov exponents are 0.317, 0.063, 0.001 and −11.90. From them, we can see that the system has good hyperchaotic dynamical characteristics and can be used for image encryption.

2.2. Cellular automata

Cellular automata (CA) is an abstract dynamical system where state, space and time are all discrete. CA has many features, such as simple

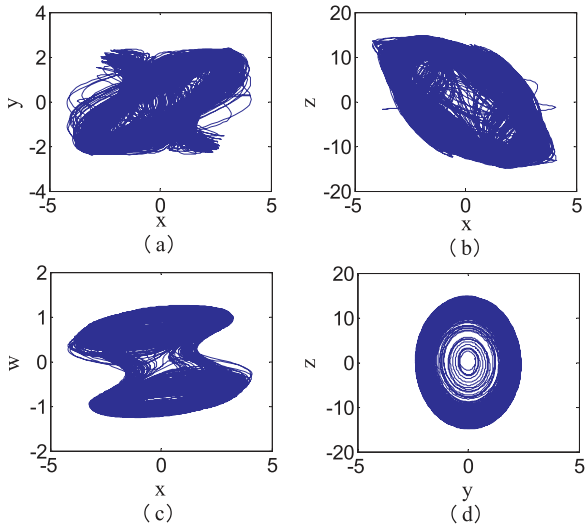


Fig. 1. Hyperchaotic attractor. (a) (*x*–y) plane, (b) (*x*–z) plane, (c) (*x*–w) plane, (d) (*y*–z) plane.

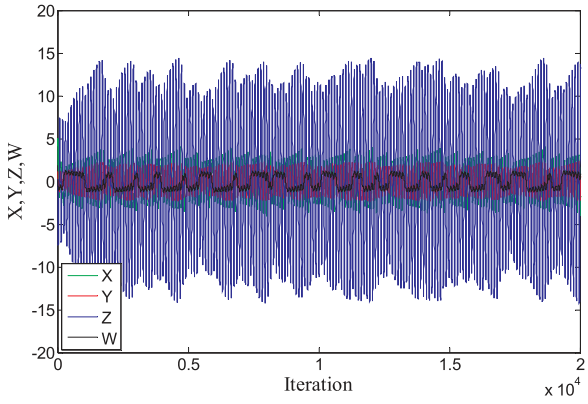


Fig. 2. Chaotic tracks of the memristive chaotic system.

regular structure, local interaction, random-like behavior and massive parallelism, and it has been applied in image encryption. A 2D CA is defined as two-dimensional lattice of cells, and every cell may take a finite number of discrete states, update in discrete time steps according to a certain local rule. In the paper, a 2D CA with von-Neumann neighborhood is used, and each cell having two states 0 or 1. The local rule with neighbor radius equal to 1 is as follows:

$$S_{i,j}^{t+1} = f(S_{i,j}^{t}, S_{i-1,j}^{t}, S_{i,j-1}^{t}, S_{i+1,j}^{t}, S_{i,j+1}^{t}) \tag{4}$$

where, $S_{i,j}^{t}$ is the state of cell (i, j) at time step t, and f denotes a Boolean function that gives the new state of a cell based on the current states of all its neighborhood cells.

From Eq. (4), we know that it has 5 state variables, every variable has two states, thus there are $2^{2^5} = 2^{32} = 4294967296$ rules. The local rule increases at the exponential law with the increasing of the cell numbers, which can upgrade the security level of encryption algorithms to some extent. However, the algorithm complexity may increase enormously. In order to reduce the computation complexity and improve encryption speed, Eq. (4) is converted to the following equation.

$$S_{i,j}^{t+1} = L \oplus (VM \times S_{i,j}^{t}) \oplus (VT \times S_{i-1,j}^{t}) \oplus (VL \times S_{i,j-1}^{t}) \oplus (VD \times S_{i+1,j}^{t}) \oplus (VR \times S_{i,j+1}^{t}) \tag{5}$$

where, $S_{i,j}^{t}$ denotes the state of cell (i, j) at time step t, and L, VM, VT, VL, VD, VR are variables with the state 0 or 1.

When $L=1$, the local rule is nonlinear, or else it is linear. The combination of VM, VT, VL, VD, VR is employed to decide which cells contribute to the state update of the central cell. The different combinations mean different local rules. For example, when (L, VM, VT, VL, VD, VR) = $(001110)_2 = (14)_{10}$, then the local rule may be called as rule 14, Eq. (5) is changed to Eq. (6).

$$S_{i,j}^{t+1} = S_{i-1,j}^{t} \oplus S_{i,j-1}^{t} \oplus S_{i+1,j}^{t} \tag{6}$$

Eq. (6) has three variables, and there are $2^6 = 64$ rules. Therefore, using the local rule shown in Eq. (6) can improve encryption efficiency enormously.

Table 1
DNA encoding rules.

Rule	1	2	3	4	5	6	7	8
A	00	00	01	01	10	10	11	11
T	11	11	10	10	01	01	00	00
C	01	10	00	11	00	11	01	10
G	10	01	11	00	11	00	10	01

2.3. DNA sequence operations

2.3.1. DNA encoding and decoding rules

A DNA sequence contains four nucleic acid bases, namely A (Adenine), G (Guanine), C (Cytosine) and T (Thymine), where A and T are complementary, and G and C are complementary. These relationships are known as Watson-Crick base pairing rules [41]. In the binary, 0 and 1 are complementary, so 00 and 11 are complementary, 10 and 01 are also complementary. Table 1 illustrates encoding and decoding rules of the DNA sequence, in order to satisfy the Watson-Crick base pairing rule. DNA decoding rules are the reverse operation of DNA encoding rules.

For example, the grayscale value of a pixel is "126", the corresponding binary number is "01111110", a DNA sequence "GTTC" is gotten by using DNA encoding rule 2. Inversely, if the DNA sequence is "TGCA", the binary number can be obtained by the rule 6 (the decoding rule is 6), that is "00011011", the decimal number is "78", and this is the decoding process of the DNA sequence.

2.3.2. DNA XOR algebraic operation

DNA XOR operation is manipulated according to traditional XOR in the binary. For there exist eight kinds of DNA encoding rules, correspondingly, eight kinds of DNA XOR rules appear. In this paper, XOR operation is used in the diffusion process. One type of XOR operation is shown in Table 2. An example of DNA XOR operation is given. Using Table 2, the XOR result of DNA sequece "AGCT" and "TGAC" is "TACG".

3. Encryption scheme

3.1. Calculating the initial values of the memristive hyperchaotic system

In the paper, SHA 256 hash function of the plain image is used to produce the initial values of the chaotic system. If there is one bit difference between two original images, their hash values will be completely different. Before encrypting the plain image, its hash value

Table 2
DNA XOR operation.

XOR	A	G	C	T
A	A	G	C	T
G	G	A	T	C
C	C	T	A	G
T	T	C	G	A

is calculated as the secret key K, and it can be shown as

$$K = k_1, k_2, \ldots, k_{32}, \text{ subject to: } k_i = \{k_{i,0}, k_{i,1}, \ldots, k_{i,7}\} \tag{7}$$

where, in $k_{i,j}$, i is the character number and j denotes the bit number.

The initial values for the memristive chaotic system can be derived as

$$\begin{cases} x_0 = \frac{1}{256}(k_1 \oplus k_2 \oplus k_3 \oplus k_4 + k_5 \oplus k_6 \oplus k_7 \oplus k_8) \\ y_0 = \frac{1}{256}(k_9 \oplus k_{10} \oplus k_{11} \oplus k_{12} + k_{13} \oplus k_{14} \oplus k_{15} \oplus k_{16}) \\ z_0 = \frac{1}{256}(k_{17} \oplus k_{18} \oplus k_{19} \oplus k_{20} + k_{21} \oplus k_{22} \oplus k_{23} \oplus k_{24}) \\ w_0 = \frac{1}{32} \times \frac{sum(k_{21}, k_{22}, \ldots, k_{32})}{max(k_{21}, k_{22}, \ldots, k_{32})} \end{cases} \tag{8}$$

where, x_0, y_0, z_0 and w_0 are initial values, $sum(k_{21}, k_{22}, \ldots, k_{32})$ denotes the sum of $k_{21}, k_{22}, \ldots, k_{32}$, $max(k_{21}, k_{22}, \ldots, k_{32})$ means the maximum value of $k_{21}, k_{22}, \ldots, k_{32}$, and $x \oplus y$ is the XOR operation of x and y.

3.2. Image encryption

In the proposed image encryption scheme, SHA 256 hash function of the original image is used to calculate the initial values of the memristive chaotic system, so that the algorithm depends on the plain image; besides, based on the first pixel of the original image and chaotic sequences from the chaotic system, two DNA encoding rule matrices for the plain image and the 2D CA are obtained, and so the DNA encoding processes are dynamical; additionally, the plain image is diffused by block, the local rule of the CA is obtained by the diffused sub-block image, the encryption result of the current sub-block image has relationship with the previous diffused sub-block image and the 2D CA.

Suppose the size of the plain grayscale image P is $M \times N$, the encryption steps are as follows and an illustration is shown in Fig. 3.

Step 1: Calculate the external key K and initial values x_0, y_0, z_0 and w_0 of the chaotic system as described in Section 3.1.
Step 2: Iterate the chaotic system N_0 times with x_0, y_0, z_0 and w_0 to avoid the transient effect, and $N_0 = 500 + floor(\frac{k_1 + k_2 + \ldots + k_{32}}{32})$.
Step 3: Continue to iterate the chaotic system $4MN$ times to get four sequences X, Y, Z, W, $X = [x_1, x_2, \ldots, x_{4MN}]$, $Y = [y_1, y_2, \ldots, y_{4MN}]$, $Z = [z_1, z_2, \ldots, z_{4MN}]$, $W = [w_1, w_2, \ldots, w_{4MN}]$.
Step 4: By implementing the following operations to every element of X, Y, Z, W as described by Eqs. (9)-(12), four vectors $X1$, $Y1$, $Z1$, $W1$ may be obtained. And then, transform $X1$, $Y1$, $Z1$, $W1$ to four $M \times 4N$ matrices X_1, Y_1, Z_1, W_1, respectively.

$$X1 = \text{mod}((abs(x_h) - floor(x_h)) \times 10^{14}, 8) + 1 \tag{9}$$

$$Y1 = \text{mod}((abs(y_h) - floor(y_h)) \times 10^{14}), 8) + 1 \tag{10}$$

$$Z1 = \text{mod}((abs(z_h) - floor(z_h)) \times 10^{14}, 8) + 1 \tag{11}$$

$$W1 = \text{mod}((abs(w_h) - floor(w_h)) \times 10^{14}, 2) \tag{12}$$

Here, x_h, y_h, z_h, w_h denote the hth element of X, Y, Z, W, $h \in [1, M \times 4N]$, $abs(x)$ is the absolute value of x, $floor(x)$ gives the largest integer less than or equal to x, and $mod(a, b)$ returns the remainder of x divided by y.

Step 5: Group by X_1, Y_1, Z_1, then $A1 = [X\_1, Y\_1]$, $A2 = [X\_1, Z\_1]$ and $A3 = [Y\_1, Z\_1]$ are gotten. Then, by using the first pixel of the plain image, the DNA rule matrices $R1$ and $R2$ for DNA encoding operations may be obtained, and $R1 = Ai(1)$, $R2 = Ai(2)$, $i = 1, 2, 3$. The choosing rule is as follows:

Firstly, a variable *index* is defined as Eq. (13), and we use it to calculate $R1$ and $R2$.

$$index = mod(P(1), 3) + 1 \tag{13}$$

Here, $P(1)$ is the first pixel value of the original image.

When $index = 1$, then $i = 1$, $Ai = A1$, then we get $R1 = X\_1$ and $R2 = Y\_1$,

When $index = 2$, then $i = 2$, $Ai = A2$, then we have $R1 = X\_1$ and $R2 = Z\_1$,

When $index = 3$, then $i = 3$, $Ai = A3$, then we obtain $R1 = Y\_1$ and $R2 = Z\_1$.

Step 6: Transform every pixel of the original image P to its binary form, matrix U with size of $M \times 8N$ appears, and then take every two elements as a group, we may obtain matrix D, $D(i, j) = [U(i, 2*j-1), U(i, 2*j)]$, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, 4N$.
Step 7: Decompose D, $R1$, $R2$ into $p \times q$ sub-blocks. The size of each block image is $m \times n$, and $m \times p = M$ and $n \times q = 4N$. Then, each sub-block denoted as D_1, R_1, R_2 is converted to a vector sized $1 \times pq$, respectively; $D\_1[k]$, $R\_1[k]$, $R\_2[k]$ are the kth sub-block of the corresponding matrix D, $R1$, $R2$, $k = 1, 2, \ldots, p \times q$.
Step 8: Pick up $2mn$ elements from matrix W_1, a new $m \times 2n$ matrix may be gotten, and it is used as the initial configuration C° of the 2D CA.
Step 9: Manipulate transition to the configuration C°, C^k denotes the configuration after k time transition, and the local rule is decided by the diffused DNA image. In detail, the diffused DNA image is employed to determine the value of L, VM, VT, VL, VD, VR, the rule number for 2D CA can be gotten as follows:

$$\begin{aligned} rule(k) &= L \times 2^5 + VM \times 2^4 + VT \times 2^3 + VL \times 2^2 + VD \times 2^1 + VR \times 2^0 \\ &= \begin{cases} mod(sum(\text{DNA}(E(k-1))), 64); & k \neq 1 \\ mod(sum(E(0)), 64); & k = 1 \end{cases} \end{aligned} \tag{14}$$

where, $rule(k)$ is the rule number for kth transition of the 2D CA, $\text{DNA}(E(k-1))$ is the diffused DNA image matrix after $(k-1)$th transition, $sum(\text{DNA}(E(k)))$ denotes the sum of ASCII code values of all the elements of diffused DNA image matrix $\text{DNA}(E(k))$, and when $k = 1$, then $E(0) = C^\circ$, $k = 1, 2, \ldots, p \times q$.

Next, transform $rule(k)$ to its binary form, and get L, VM, VT, VL, VD, VR. Then, the corresponding local rule of the CA can be obtained by renewing Eq. (5).

Step 10: Encode the kth sub-block matrix $D\_1[k]$ of D_1 according to the kth sub-block matrix $R\_1[k]$ of $R1$, and $\text{DNA}(D\_1[k])$ is gotten. At the same time, encode the cell configuration C^k according to kth sub-block matrix $R\_2[k]$ of $R2$, and $\text{DNA}(C^k)$ is obtained.
Step 11: Apply the diffusion operation to the plain sub-block image as Eq. (15).

$$\begin{cases} \text{DNA}(E[1]) = \text{DNA}(C^1) \oplus \text{DNA}(D\_1[1]) \oplus \text{DNA}(D\_1[p \times q]), & k = 1 \\ \text{DNA}(E[k]) = \text{DNA}(C^k) \oplus \text{DNA}(D\_1[k]) \oplus \text{DNA}(E[k-1]), & k \neq 1 \end{cases} \tag{15}$$

where, $\text{DNA}(E[k])$ is the encrypted DNA matrix of kth sub-block image, $\text{DNA}(E[k-1])$ is the encrypted DNA matrix of $(k-1)$th sub-block image, $\text{DNA}(C^k)$ denotes the DNA matrix of cell configuration C^k, and $\text{DNA}(D\_1[k])$ is the DNA matrix of the kth plain sub-block image D_1, $k = 1, 2, \ldots, p \times q$.

Step 12: Set $k = k + 1$, do Steps 9–12 in a loop, until all the sub-blocks of the plain image have been encrypted.
Step 13: Transform the encrypted sub-blocks of the plain image to a $M \times 4N$ matrix, then decode it by the rule matrix $R1$, we can obtain the cipher image.

The decryption process is the reverse one of the encryption

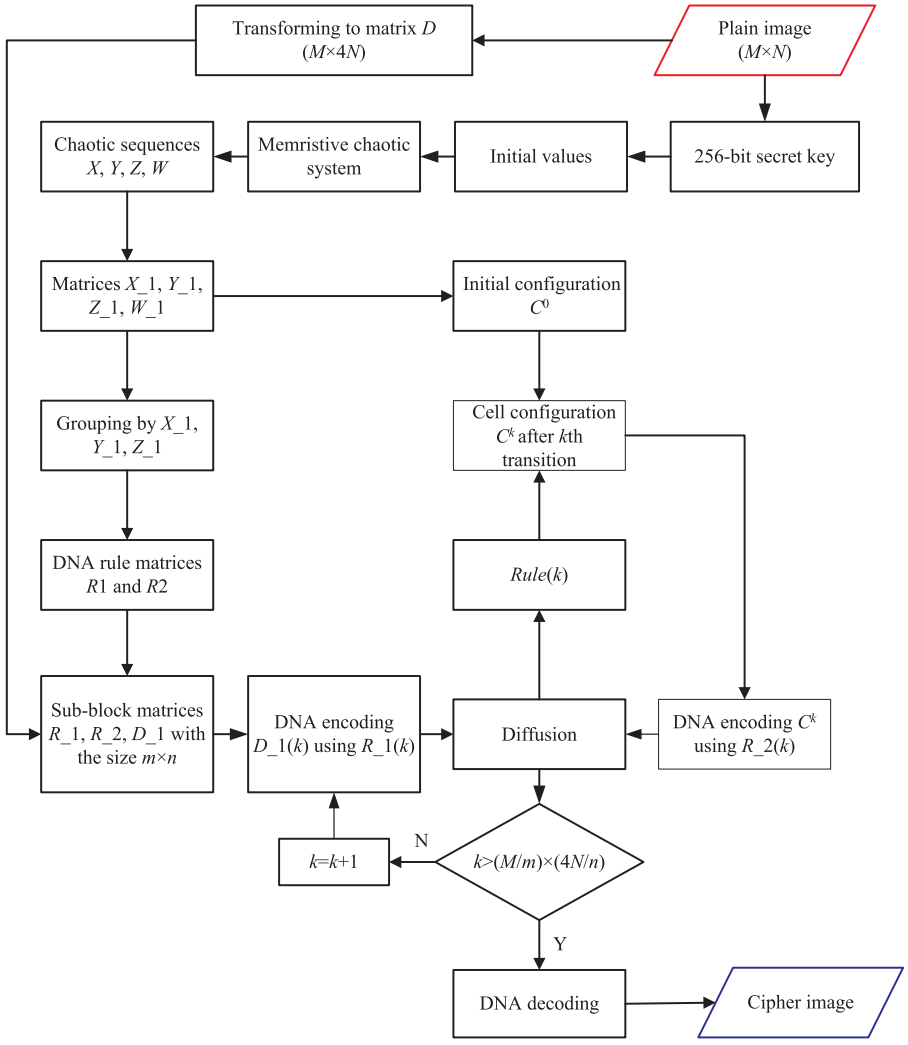


Fig. 3. The diagram of the proposed encryption algorithm.

algorithm. Before the receiver decrypts the cipher image, the secret keys must be transmitted to it through a secure channel, and keys consist of the 256-bit hash value K produced through the SHA 256 hash function of the plain image, the first pixel $P(1)$ of the original image, and the parameter c of the memristive chaotic system. In decryption phase, the reverse operation of Eq. (15) is as follows:

$$\begin{cases} \mathrm{DNA}(D[1]) = \mathrm{DNA}(C^1) \oplus \mathrm{DNA}(E[1]) \oplus \mathrm{DNA}(D[p \times q]), \ k = 1 \\ \mathrm{DNA}(D[k]) = \mathrm{DNA}(C^k) \oplus \mathrm{DNA}(E[k]) \oplus \mathrm{DNA}(E[k-1]), \ k \neq 1 \end{cases} \quad (16)$$

Here, DNA($D[k]$) denotes the decrypted DNA matrix of kth sub-block image, DNA($E[k]$) is the encrypted DNA matrix of the kth plain sub-block image, DNA(C^k) denotes the DNA matrix of cell configuration C^k, DNA($E[k\text{-}1]$) is the encrypted DNA matrix of the (k-1)th plain sub-block image, $k = p \times q, p \times q - 1, \ldots, 1$.

3.3. Discussion

In the encryption steps, the local rule of the CA is calculated by the diffused DNA image, the cell configuration is updated according to the local rule. We give an example to make the whole process easy to understand.

First, we set $C^0 = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$, when $k=1$, $rule(1) = mod(sum(C^0), 64) = 11$, convert $(11)_{10}$ to its binary form $(001011)_2$, then we can get $L=0$, $VM=0$, $VT=1$, $VL=0$, $VD=1$, $VR=1$, the local rule is $S_{i,j}^{t+1} = S_{i-1,j}^t \oplus S_{i+1,j}^t \oplus S_{i,j+1}^t$. According to the local rule, we may obtain C^1, that is $C^1 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$.

We assume the diffused DNA matrix is as follows:

$$\mathrm{DNA}(E(1)) = \begin{bmatrix} \mathrm{T} & \mathrm{A} & \mathrm{T} \\ \mathrm{C} & \mathrm{T} & \mathrm{C} \\ \mathrm{C} & \mathrm{C} & \mathrm{A} \\ \mathrm{G} & \mathrm{G} & \mathrm{C} \end{bmatrix}.$$

Then, we can get,

$$\begin{aligned} rule(2) &= \mathrm{mod}(sum(\mathrm{DNA}(E(1))), \ 64). \\ &= mod((84 + 65 + 84 + 67 + 84 + 67 + 67 + 67 + 65 + 71 + 71 \\ &\quad + 67), \ 64). \\ &= 27. \end{aligned} \quad (17)$$

Next, convert $(27)_{10}$ to $(011011)_2$, we may obtain $L=0$, $VM=1$, $VT=1$, $VL=0$, $VD=1$, $VR=1$, the corresponding local rule is $S_{i,j}^{t+1} = S_{i,j}^t \oplus S_{i-1,j}^t \oplus S_{i+1,j}^t \oplus S_{i,j+1}^t$, the configuration of the CA after the second transition is $C^2 = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}$. And so on, all the

configurations of the cell can be obtained.

Our contributions are as follows:

Firstly, a novel image encryption algorithm combined the memristive chaotic system, 2D CA and DNA sequence operations is proposed, diffusion is adopted without permutation, good encryption effect is gotten and much time can be saved at the same time.

Secondly, a dynamic DNA encoding scheme is adopted. When we encode the plain image and 2D CA, the plain image and the memristive chaotic system are united to produce the DNA rule matrices, there are different DNA rule matrices for different original images, and different elements in the plain image and CA have different DNA encoding rules. As a consequence, our method has high resisting ability to statistical attack and known-plaintext attack.

Thirdly, image encryption scheme by block is presented. The diffusion result of the current sub-block image depends on the previous diffused sub-image and the current configuration of 2D CA. The cell transition law of 2D CA is determined by the previous diffused sub-block image, thus the law can be gotten dynamically, and therefore the proposed encryption method has high security level against chosen-plaintext attack.

Lastly, the SHA 256 hash function of the original image is utilized to generate the secret keys, so our algorithm has large key space. Besides, the initial values of the chaotic system are generated from the secret keys, and the initial configuration of the 2D CA has relationship with the plain image. Thus, our encryption algorithm depends on the plain image, and resists against known-plaintext and chosen-plaintext attacks effectively.

4. Simulation results

In the paper, the standard 256×256 image of Lena shown in Fig. 4(a) is employed as the test image. All experiments are manipulated by Matlab 2014a to run the encryption and decryption process in computer with 3.3 GHz CPU, 4 GB memory and Windows 7 operating system. The experiment parameters are presented in Table 3, the cipher image is shown in Fig. 4(b), and the decrypted image is illustrated in Fig. 4(c).

From the figures, we can see that the cipher image is a noise-like one, there are no relationship between the original image and the cipher image, and the decrypted image is just like the original image from the view point. These show that our algorithm has good encryption and decryption effect.

5. Security analyses

A good encryption scheme should have large key space to resist brute force attack, be sensitive to the secret key, and resist all kinds of known attacks such as differential attack, statistical attack, known-plaintext and chosen-plaintext attacks, and others. In this section, we will assess the security performances of the proposed encryption algorithms.

Table 3
Experiment parameters.

Items	Parameter values
Parameters of the memristive chaotic system	$a=30$, $b=36$, $c=0.1$, $d=2.5$, $e=3.5$, $m_0=-0.5$, $m_1=-0.1$
256-bit secret key (in hexadecimal form)	2 6 2 8 8 F E 1 3 E E E E 0 C 6 B A A F 8 D A 1 C 8 C 2 5 F 5 A C 3 6 9 A 0 9 9 4 6 5 6 D 5 9 E 4 6 0 5 E 3 D 0 4 7 6 7 2 3 E 9
The first pixel of the plain image	$P(1)=162$

5.1. Key space

The key space of a good encryption algorithm should be large enough, so that it can resist all kinds of brute-force attacks from information eavesdroppers. In the proposed algorithm, the secret keys include:

(1) the 256-bit hash value K generated by the hash function of the plain image,
(2) the first pixel $P(1)$ of the original image, and
(3) the parameter c of the memristive chaotic system.

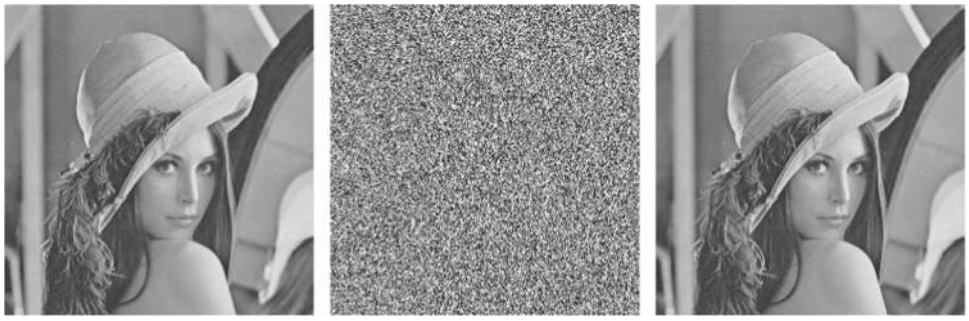
It is known that the key space of SHA 256 hash function with complexity of the best attack is 2^{128} larger than 2^{100} [42], and this means our algorithm is enough to prevent the exhaustive searching and any brute force attack.

5.2. Key sensitivity

A good encryption method must have high sensitivity to the secret key in encryption process and decryption process, which means that a completely different cipher image may be obtained when the secret key has a little change in the encryption process, and in the decryption process, the plain image cannot be recovered by a decryption key with a slight difference with the encryption key.

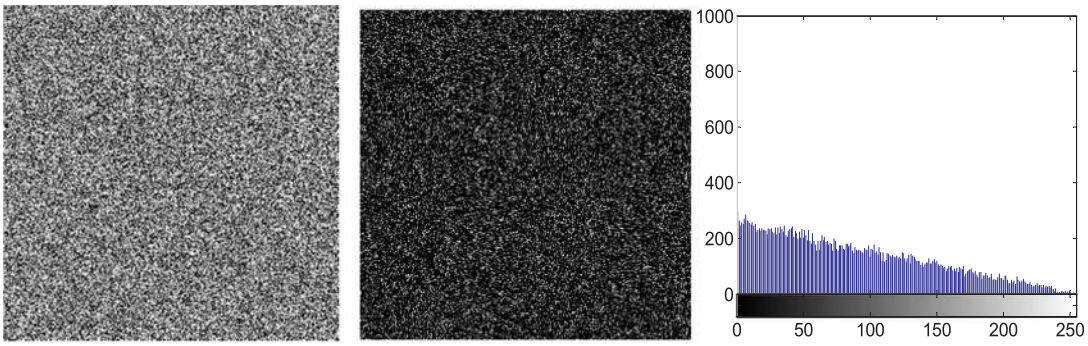
Firstly, we test the key sensitivity in the encryption process. Lena (Fig. 4(a)) is the test image, and its cipher image is Fig. 4(b). We perform a key sensitivity test using a key that is just a bit different from the original encryption key to encrypt Lena, for example, set $c'=0.10000000001$ and keep other keys constant. The corresponding cipher image is shown in Fig. 5(a), Fig. 5(b) is the subtraction of Fig. 4(b) and Fig. 5(a), and there are 49.75% pixel changed. The histogram of Fig. 5(b) is illustrated in Fig. 5(c). Next, we change the 256-bit hash value to evaluate the key sensitivity. The original hash value K and the changed $K1$ and $K2$ are as follows:

$K=$ 38 40 143 225 62 238 224 198 186 175 141 161 200 194 95 90 195 105 160 153 70 86 213 158 70 5 227 208 71 103 35 233

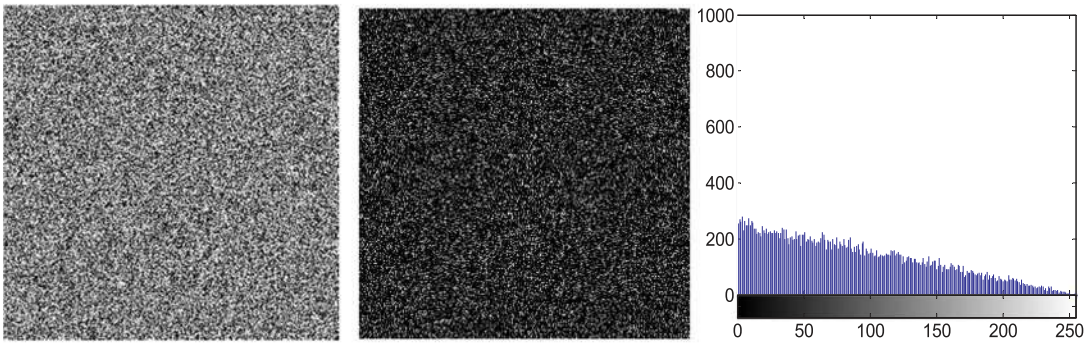


(a) plain image of Lena (b) corresponding cipher image (c) decrypted image

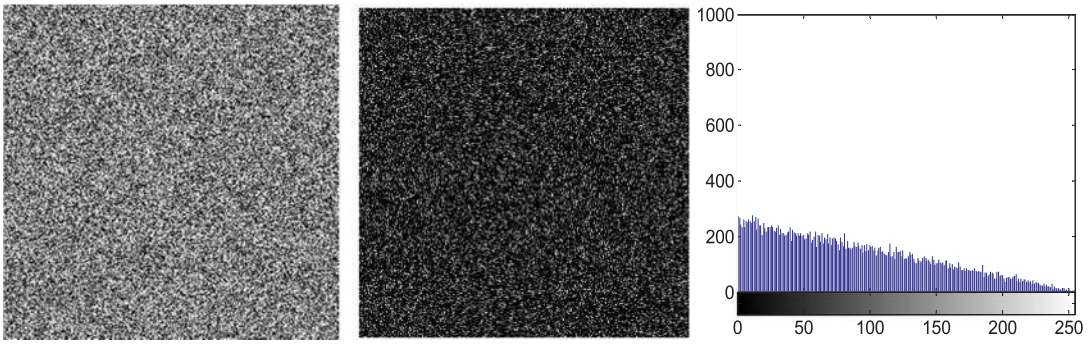
Fig. 4. Simulation results. (a) plain image of Lena (b) corresponding cipher image (c) decrypted image.



(a) the cipher image with c' (b) the subtraction of two cipher images (c) the histogram of Fig. 5(b)



(d) the cipher image with $K1$ (e) the subtraction of two cipher images (f) the histogram of Fig. 5(e)



(g) the cipher image with $K2$ (h) the subtraction of two cipher images (i) the histogram of Fig. 5(h)

Fig. 5. Key sensitivity results in encryption process. (a) the cipher image with c' (b) the subtraction of two cipher images (c) the histogram of Fig. 5(b). (d) the cipher image with $K1$ (e) the subtraction of two cipher images (f) the histogram of Fig. 5(e). (g) the cipher image with $K2$ (h) the subtraction of two cipher images (i) the histogram of Fig. 5(h).

$K1 =$ **39** 40 143 225 62 238 224 198 186 175 141 161 200 194 95 90 195 105 160 153 70 86 213 158 70 5 227 208 71 103 35 233
$K2 =$ 38 40 143 225 62 238 224 198 186 175 141 161 200 194 95 90 195 105 160 153 70 86 213 158 70 5 227 208 71 103 35 **234**

The cipher image is illustrated in Fig. 5(d) with $K1$, the subtraction of Fig. 4(b) and Fig. 5(d) is Fig. 5(e), (f) is the histogram of Fig. 5(e), and there are 49.75% pixel modified. Fig. 5(g) is the cipher image with $K2$, Fig. 5(h) is its subtraction with Fig. 4(b), and Fig. 5(i) is its histogram, and the pixel change ratio is 49.90%.

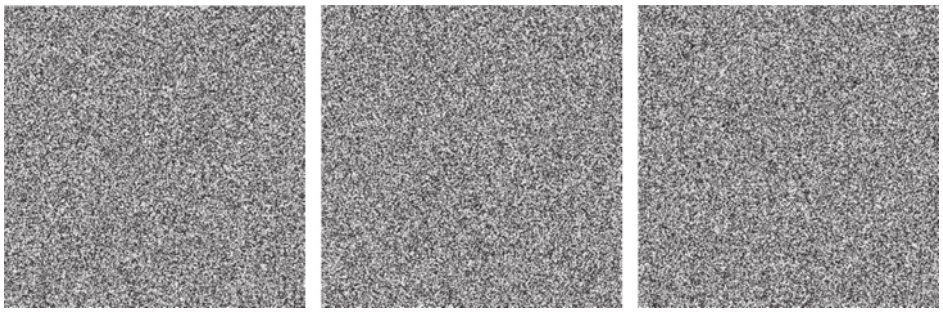
Secondly, the key sensitivity in decryption process is evaluated. We use $c' = 0.10000000001$ to decrypt the cipher image Fig. 4(b), the recovered image is illustrated in Fig. 6(a). Next, $K1$ and $K2$ are employed separately to decrypt Fig. 4(b), and the recovered images are Figs. 6(b) and (c). Figs. 6(a), (b) and (c) are something like noise, we cannot get any visible information from them, this means our encryption algorithm has high sensitivity to the secret key, and the plain image cannot be recovered with a slight changed decryption key. Pixel change ratio between the recovered images and the plain image (Fig. 4(a)) are calculated and given in Table 4. From the table, we watch that the pixel change ratio is over 49% and near the ideal value 50%, which means that the proposed algorithm is highly sensitive to the change of the key.

5.3. Statistical attack

5.3.1. Histogram analysis

We can get the distribution of pixel values of an image from its histogram. If it is not flat enough, some amount of information can be leaked by the statistical attack. Thus, a uniform and flat distribution is desirable for a good encryption scheme. The histograms of plain image Lena and its cipher image by the proposed algorithm are shown in Fig. 7. It is clear that histogram of the cipher image is uniform and significantly different from that of the plain image. So our algorithm can make the statistical attack invalid.

Moreover, variances of histograms are mostly used to quantitatively test the uniformity of the images. When the variance is lower, the



(a) the recovered image with c' (b) the recovered image with $K1$ (c) the recovered image with $K2$

Fig. 6. Key sensitivity results in decryption process. (a) the recovered image with c' (b) the recovered image with $K1$ (c) the recovered image with $K2$.

Table 4
Quantitative experimental results.

Recovered image	Decryption key								Pixel change ratio
	a	b	c	d	e	m_0	m_1	256-bit secret key	
Fig. 4(c)	30	36	0.1	2.5	3.5	−0.5	−0.1	K	0
Fig. 6(a)	30	36	0.1 $+10^{-12}$	2.5	3.5	−0.5	−0.1	K	49.64%
Fig. 6(b)	30	36	0.1	2.5	3.5	−0.5	−0.1	$K1$	49.77%
Fig. 6(c)	30	36	0.1	2.5	3.5	−0.5	−0.1	$K2$	49.75%

uniformity of the image is higher and the security level of the encryption algorithm is higher. The variance of histograms may be obtained by the following equation [43],

$$\mathrm{var}(X) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}(x_i - x_j)^2 \tag{18}$$

Here $X = \{x_1, x_2, \ldots, x_{256}\}$ denotes the vector of the histogram values, and x_i and x_j are the numbers of pixels which gray values are equal to i and j, respectively. For Lena (256×256), the variances of the histograms of the plain image and cipher image are given in Table 5. From the results, it can be concluded that the variance of the cipher image is lower, it is 260.7188, and the corresponding value of the plain image is 39868. Compared with Ref. [44] and Ref. [45], for the same Lena image, variance of our algorithm is less than that of Ref. [45] and higher than that of Ref. [44], and thus the proposed algorithm is more efficient and secure.

5.3.2. Correlation coefficient analysis

Randomly select 10000 pairs of pixels in horizontal, vertical and diagonal direction from the plain and cipher images and calculate the correlation coefficients according to the following equations.

Table 5
Variances of the histograms of the Lena (256×256).

Algorithms	Ours		Ref. [44]	Ref. [45]
Images	Plain image	Cipher image	Cipher image	Cipher image
Variances	39868	260.7188	244.3125	276.3906

$$r_{x,y} = \frac{E((x - E(x))(y - E(y)))}{\sqrt{D(x)D(y)}} \tag{19}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{20}$$

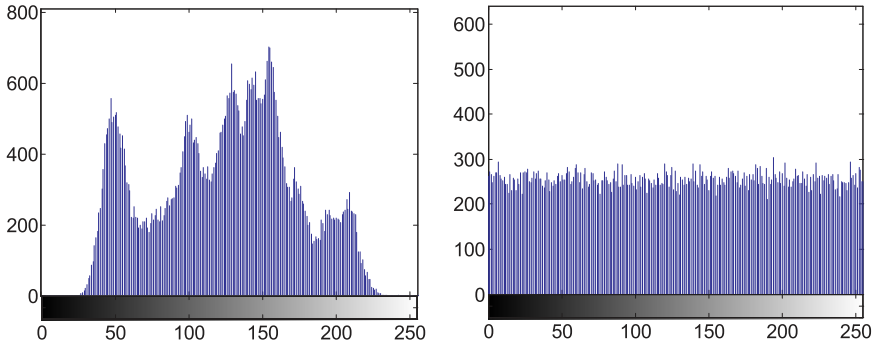
$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2 \tag{21}$$

where x, y are gray level values of two adjacent pixels of the image, N is the total number of pixels selected from the image, and $E(x)$ and $D(x)$ denote the expectation and variance of variable x, respectively.

Fig. 8 plots the correlation of two adjacent pixels of the plain image Lena and its cipher image in the horizontal, vertical and diagonal direction. Table 6 illustrates the correlation coefficients of two adjacent pixels in the plain images (shown in Fig. 9) and their corresponding cipher images. From the results, it is clear that the correlations between adjacent pixels in the original images are strong, the correlation coefficients are all near 1, however, those of the cipher images are all smaller than 0.03, which indicates that the correlation is greatly reduced in the cipher images and the opponents cannot get any useful information from the cipher images by statistical attack.

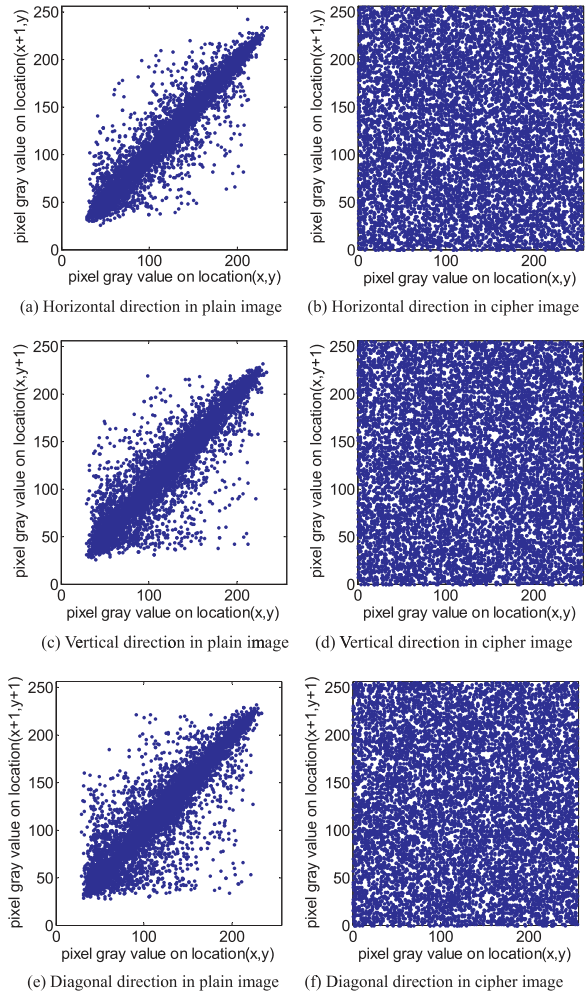
5.4. Differential attack

We employ number of pixels change rate (NPCR) and unified average changing intensity (UACI) to measure the resisting differential attack performance of the encryption algorithm. The following for-



(a)Histogram of Lena (b)Histogram of cipher image

Fig. 7. Histogram analysis of the plain and cipher images. (a) Histogram of Lena (b) Histogram of cipher image.



(a) Horizontal direction in plain image (b) Horizontal direction in cipher image
(c) Vertical direction in plain image (d) Vertical direction in cipher image
(e) Diagonal direction in plain image (f) Diagonal direction in cipher image

Fig. 8. Correlation of two adjacent pixels of the plain image Lena (256×256) and its cipher image. (a) Horizontal direction in plain image (b) Horizontal direction in cipher image. (c) Vertical direction in plain image (d) Vertical direction in cipher image. (e) Diagonal direction in plain image (f) Diagonal direction in cipher image.

Table 6
Correlation coefficients of two adjacent pixels in the plain and cipher images.

Images		Correlation coefficients		
		Horizontal	Vertical	Diagonal
Lena (256×256)	Plain image	0.9588	0.9260	0.9025
	Cipher image	−0.0016	−0.0033	0.0130
Brain (512×512)	Plain image	0.9851	0.9813	0.9690
	Cipher image	0.0012	0.0253	0.0207
Pepper (256×256)	Plain image	0.9459	0.9474	0.8994
	Cipher image	−0.0021	−0.0008	−0.0077
Baboon (512×512)	Plain image	0.7508	0.8562	0.7153
	Cipher image	−0.0061	0.0130	0.0017
City (512×512)	Plain image	0.8874	0.8351	0.7693
	Cipher image	0.0005	−0.0081	−0.0053

mulas can be used to calculate NPCR and UACI.

$$\mathrm{NPCR} = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \tag{22}$$

$$\mathrm{UACI} = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\% \tag{23}$$

where $D(i, j)$ is defined as

$$D(i,j) = \begin{cases} 1, & C_1(i,j) \neq C_2(i,j) \\ 0, & otherwise \end{cases} \tag{24}$$

Here, C_1 and C_2 are the cipher images before and after one pixel of the plain image is modified, and W and H are the width and height of the image, respectively.

NPCR and UACI for different images are presented in Table 7. Table 8 is the results for Lena image when the pixel values at different positions have changed. From the two tables, we can watch that UACI is more than 0.33, NPCR is over 0.99, which mean that the encryption scheme is highly sensitive to the change in the plain images, and two completely different cipher images may be gotten even if there has a little change in the original images, thus our algorithm can resist the differential attack effectively.

5.5. Information entropy

Let m be the information source, then the formula for computing information entropy is as follows,

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

where $p(m_i)$ denotes the probability of symbol m_i. For a random image with 256 gray levels, the ideal entropy value is 8 [46]. The more the entropy is close to 8, the more random the image is, and the less possible the encryption scheme divulges information. Information entropies for the plain images and cipher images are listed in Table 9. After encryption, the entropies of the cipher images are close to the ideal value 8, which means that the cipher images gotten by our encryption algorithm are highly random and could hardly leak information.

Recently, by computing the sample mean of conventional information entropy over a number of non-overlapping and randomly selected image blocks, Wu et at [47] proposed the local Shannon entropy to measure the image randomness, and it may overcome some known weaknesses of the conventional information entropy. The local Shannon entropy has some advantages, first, it can capture local image block randomness that may not be correctly gotten by information entropy; second, it is able to assess the image randomness by the same set of parameter regardless of the various sizes of test images, and finally, it only needs a portion of the pixel information to measure the image, and it has higher efficiency. Next, the local Shannon entropy is used to measure the randomness of our encryption algorithm.

The (k, T_B)-local Shannon entropy with respect to local image blocks may be computed by the following steps: firstly, non-overlapping image blocks $S_1, S_2,\ldots, S_k$ with T_B pixels for a test image S are randomly selected, and then information entropy $H(S_i)$ for all image blocks via Eq. (25) may be obtained, finally, the local Shannon entropy over these k image blocks is computed by the following equation [47],

$$\overline{H}_{k,T_B}(m) = \sum_{i=1}^{k} \frac{H(S_i)}{k} \tag{26}$$

In the experiment, for three test images, we select $k = 32$ and $T_B = 1936$, and the results are demonstrated in Table 10. From the table, we can see that the local Shannon entropies of the cipher images are greater than 7.90, the local image blocks are chosen randomly in the local Shannon entropy measure, thus the cipher images generated by the proposed encryption algorithm have good local randomness, and our algorithm is enough secure to resist the entropy attack.

5.6. Noise attack

In real communication atmosphere, there are all kinds of noises, such as salt-and-pepper noise (SPN), gaussian noise (GN) and speckle noise (SN). When the cipher images are transmitted across the channel,

Fig. 9. Test images. (a) 512×512 Brain (b) 256×256 Pepper (c) 512×512 Baboon (d) 512×512 City.

Table 7
NPCR and UACI for different images.

Images	Lena	Brain	Pepper	Baboon	City
UACI	0.3342	0.3352	0.3340	0.3349	0.3350
NPCR	0.9961	0.9961	0.9965	0.9960	0.9960

Table 8
The Lena image for different positions.

Positions	(1, 1)	(20, 35)	(114, 197)	(256, 256)
UACI	0.3347	0.3367	0.3356	0.3345
NPCR	0.9963	0.9959	0.9961	0.9962

Table 9
Information entropies for the plain images and cipher images.

Images		Lena	Brain	Pepper	Baboon	City
Information Entropies	Plain image	7.4428	5.9398	7.5326	7.3683	6.7468
	Cipher image	7.9971	7.9993	7.9973	7.9992	7.9993

Table 10
Local entropies for the plain images and cipher images.

Images		Brain	Baboon	City
Size		512×512	512×512	512×512
Local entropies	Plain image	3.9364	6.9919	6.9810
	Cipher image	7.9020	7.9013	7.9048

they will be inevitably disturbed by these noises, which give much difficulty in recovering the plain images. Thus, the robustness against noise is an important index to test the performance of the encryption scheme.

We employ the Peak Signal-to-Noise Ratio (PSNR) to compute the quality of the recovered image after attack. For a grayscale image, the PSNR may be computed by the following equations.

$$\mathrm{PSNR} = 10 \times \log_{10}\left(\frac{255 \times 255}{\mathrm{MSE}}\right)(\mathrm{dB}) \tag{27}$$

$$\mathrm{MSE} = \frac{1}{mn}\sum_{i=1}^{m}\sum_{i=1}^{n}\|I_1(i,j) - I_2(i,j)\|^2 \tag{28}$$

where MSE is the mean square error between the recovered image $I_2(i, j)$ and the plain image $I_1(i, j)$, and m and n are the width and height of the image, respectively.

The cipher image Lena (shown in Fig. 4(b)) is used as the test image. Fig. 10 illustrates the cipher and recovered images under SPN, GN and SN with different variances. The MSE, PSNR, NPCR and UACI of the noisy decrypted and the plain Lena image are listed in Table 11. From the results, it is clear that first, the proposed algorithm has the strongest resisting ability to SPN, with the PSNR above 40 dB; second, when the variance changes from 0.00001 to 0.0001, MSEs, PSNRs, NPCRs and UACIs of the recovered images suffered from GN are almost the same with PSNRs about 30 dB, PSNRs of those from SN are decreasing from 32.2799 dB to 30.0043 dB. All in all, our encryption algorithm has a robustness property on resisting noise attack to some degree.

5.7. Occlusion attack

When the cipher images are transmitting through the communication channel, they may lose some data, which also makes decrypting the original image harder. Occlusion attack is used to test the capacity of recovering plain images from cipher images. We employ PSNR to evaluate the resisting occlusion performance of the encryption algorithm.

Fig. 11(a)-(c) show the cipher images of Fig. 4(b) with 3/4, 1/2, 1/4 occlusion respectively, and Fig. 11(d)-(f) give the corresponding recovered images. The quantitative results of resisting occlusion attack of the proposed algorithm are presented in Table 12. From the figure and the table, we can see that firstly, when the cipher image has 75% data lost, the PSNR is 28.4033 dB and the decrypted image is also recognizable; secondly, when the cipher image has a quarter and a half data cut, the PSNRs are above 30 dB and they are more than those in Ref. [48]. Therefore, our scheme is more effective in view of resisting occlusion attack.

5.8. Known-plaintext and chosen-plaintext attacks

In the encryption process, some skills are manipulated to improve the ability to resist known-plaintext and chosen-plaintext attacks. Firstly, we use the SHA 256 hash function of the original image to compute the initial values of the memristive chaotic system, thus our algorithm has high sensitivity to the change of the plain image. Moreover, the DNA encoding schemes of the plain image and the 2D CA cell configuration are all dynamic, two DNA rule matrices are obtained based on the first pixel of the original image and chaotic sequences from the chaotic system, therefore, the DNA encoding process has close relationship with the plain image. Besides, in the diffusion process, the encryption result of the current sub-block image is influenced by the previous diffused sub-block image, and the local rule of the CA is also calculated by the diffused sub-block image. Therefore, our algorithm could withstand known-plaintext and chosen-plaintext attacks.

Some attackers always use all white and all black to make the permutation process of encryption methods invalid, and then try to get some useful information. Our algorithm has no permutation step, so it has good performance for these attacks. All white and all black images

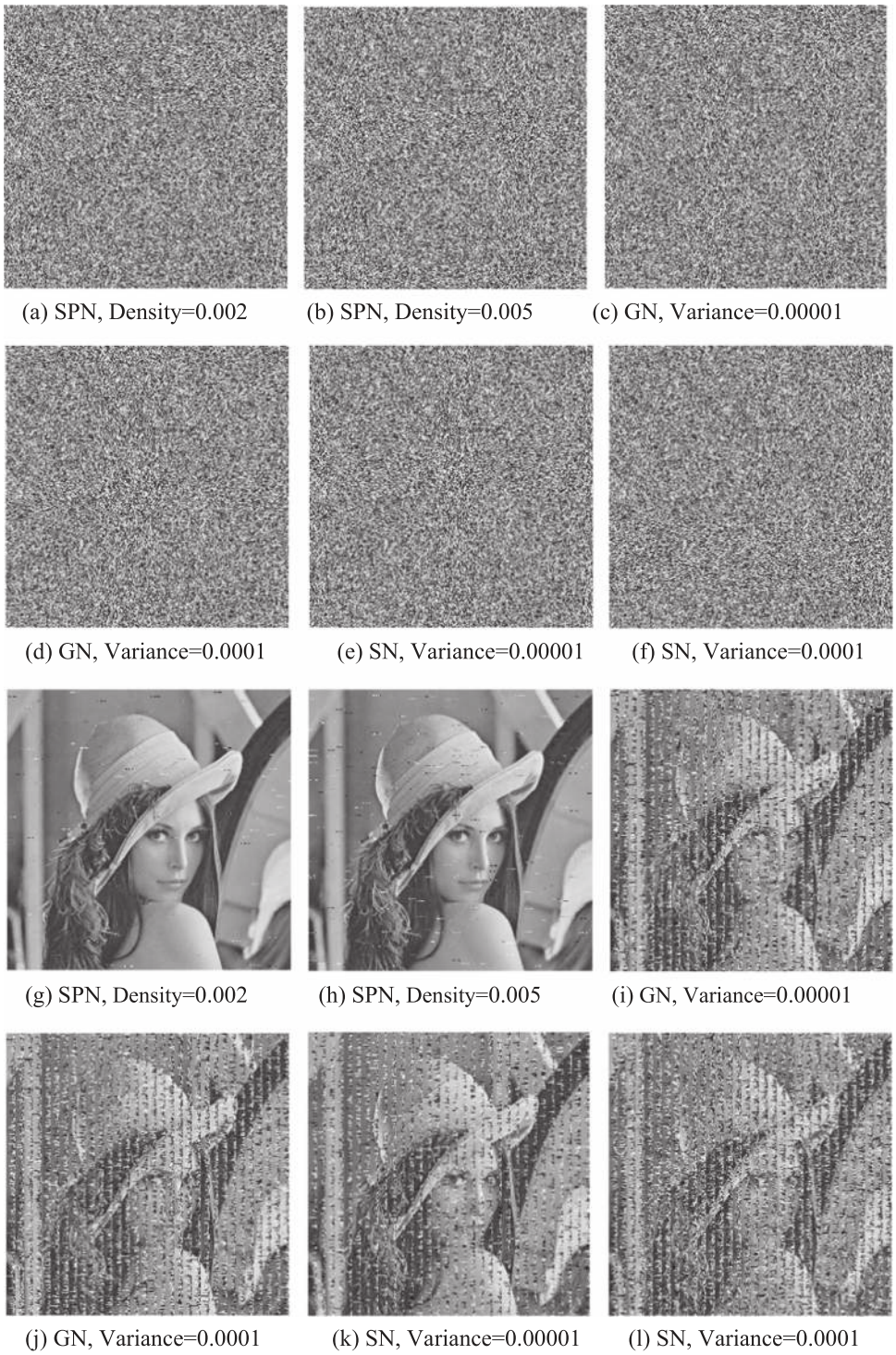


Fig. 10. Cipher and recovered images under SPN, GN and SN with different level of noise. (a) SPN, Density = 0.002 (b) SPN, Density = 0.005 (c) GN, Variance = 0.00001. (d) GN, Variance = 0.0001 (e) SN, Variance = 0.00001 (f) SN, Variance = 0.0001. (g) SPN, Density = 0.002 (h) SPN, Density = 0.005 (i) GN, Variance = 0.00001. (j) GN, Variance = 0.0001 (k) SN, Variance = 0.00001 (l) SN, Variance = 0.0001.

with the size of 256×256 are used as test images, their cipher images and histograms of cipher images are illustrated in Fig. 12, and Table 13 gives their entropies and correlation coefficients. From the results, we can watch that firstly, the cipher images are noisy, different from the original images, and their histograms distribute randomly, nothing useful information can be obtained from analyzing the cipher images; besides, entropies of the cipher images are more than 7.99, and correlation coefficients in three directions are near 0. These all mean that our algorithm has good encryption effect for all white and all black images and high security level.

5.9. Speed analyses and performance comparison

Running speed is an important characteristic parameter for encryption algorithms, when the security level may meet the requirements. Chaos-based image encryption scheme is mostly composed of permutation process and diffusion process. Thus, the round number of permutation, diffusion and image-scanning operations directly influ-

Table 11
Quantitative results of resisting noise attacks.

Noise	Density or variance	Ours			
		MSE	PSNR	NPCR	UACI
Salt-and-pepper noise (SPN)	0.002	1.3164	46.9371	0.0159	0.0022
	0.005	2.4054	44.3189	0.0393	0.0046
Gaussian noise (GN)	0.00001	53.5938	30.8397	0.5554	0.1238
	0.0001	53.5665	30.8419	0.5506	0.1230
Speckle noise (SN)	0.00001	38.4676	32.2799	0.4209	0.0841
	0.0001	64.9603	30.0043	0.6439	0.1501

ence the running time. Table 14 is the performance comparison of different methods to meet a satisfactory security level. In the table, our algorithm needs 1 round image-scanning and 1 round diffusion to achieve NPCR > 0.996 and UACI > 0.333, whereas Refs. [51,52] needs 6 rounds image-scanning, 3 rounds permutation and 3 rounds diffusion. In this respect, our encryption algorithm has higher running efficiency than Refs. [44,49–52].

In Table 15, Cameraman (256×256) is as the test image, correlation coefficient, NPCR, UACI and entropy of the cipher images generated from the proposed method and other latest methods are calculated and listed. In order to easily compare, we modify the first pixel of the original image to 250 to compute the NPCR and UACI. From the results, it is clear that firstly, horizontal correlation coefficient produced by our algorithm is the least, vertical correlation coefficient is less than that in Ref. [3] and larger than those in Refs. [44,45,53], diagonal correlation coefficient is the largest, which means that our algorithm has limited ability to remove the correlation compared with the four existing algorithm, but the correlation coefficients are less than 0.03 and the correlation between adjacent pixels are nearly eliminated; secondly, NPCR values generated by our method is more than 99%, UACI value is more than 33%, but NPCR value is the minimum, UACI is just larger than Ref. [45]; additionally, as for entropy, our result is larger than those in Refs. [44,53], less than that in Ref. [3] and the same with that in Ref. [45]. In the paper, only diffusion process is employed, combining confusion process in the following work may improve the encryption performance.

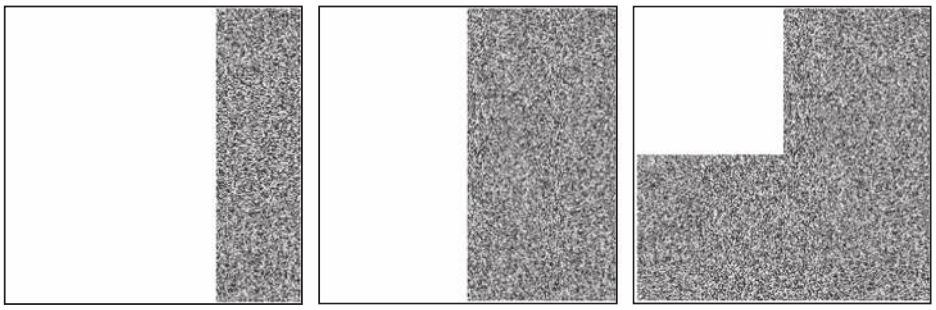
Table 12
Quantitative results of resisting occlusion attack.

Occlusion	Ours				Ref. [48]
	MSE	PSNR	NPCR	UACI	PSNR
3/4	93.9174	28.4033	0.9877	0.1255	–
1/2	38.1644	32.3142	0.9517	0.0315	30.40
1/4	18.9023	35.3657	0.4756	0.0161	33.37

6. Conclusions

In this paper, we propose a new image encryption algorithm using a memristive chaotic system, 2D cellular automata and DNA sequence operations. Our scheme differs from others in four ways.

First, the plain image is divided into sub-block images, and image diffusion encryption scheme by block is employed to save encryption time. Second, in the past encryption algorithms, DNA encoding rules are constant or taken as a key (changing from 1 to 8), in our algorithm, we use the plain image and the memristive chaotic system to produce two DNA rule matrices for encoding the plain image and 2D CA, each element has its own rule and the dynamical DNA encoding process is achieved. Third, the encryption effect of the current sub-block image has relationship with the previous diffused sub-image and the current configuration of 2D CA, 2D CA is updated according to the previous diffused sub-image, and its initial configuration is determined by the chaotic sequences. At the same time, in order to reduce the computation complexity, a simplified local rule of 2D CA is presented, and it can be employed in other fields. Lastly, the SHA 256 hash function is used to get the secret key and initial values of the memristive chaotic system, thus the proposed encryption algorithm is highly sensitive to the original image.



(a) Cipher image(3/4 occlusion) (b) Cipher image(1/2 occlusion) (c) Cipher image(1/4 occlusion)

(d)Recovered image(3/4 occlusion) (e)Recovered image(1/2 occlusion) (f)Recovered image(1/4 occlusion)

Fig. 11. Occlusion attack analysis results. (a) Cipher image(3/4 occlusion) (b) Cipher image(1/2 occlusion) (c) Cipher image(1/4 occlusion). (d) Recovered image(3/4 occlusion) (e) Recovered image(1/2 occlusion) (f) Recovered image(1/4 occlusion).

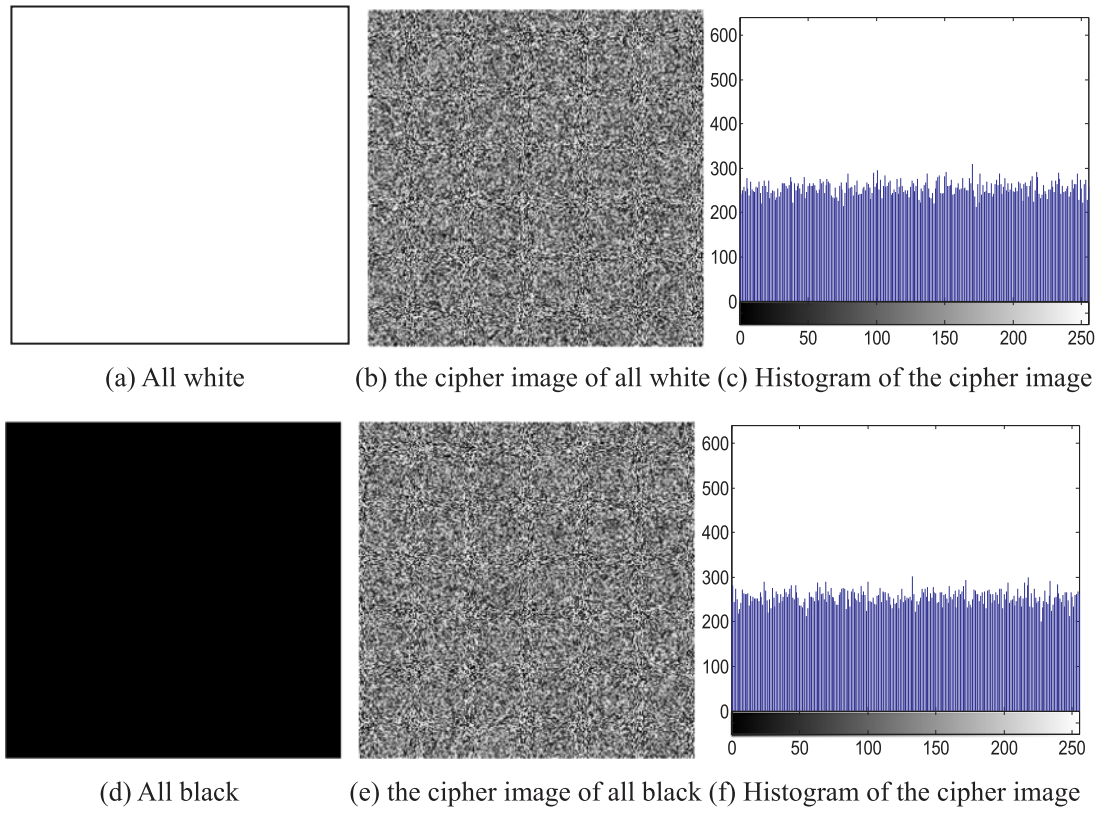


Fig. 12. Experimental results of all white and black images. (a) All white (b) the cipher image of all white (c) Histogram of the cipher image. (d) All black (e) the cipher image of all black (f) Histogram of the cipher image.

Table 13
The entropies and correlation coefficients of the plain, cipher images of all white and all black images.

Images (256 × 256)	Entropies	Correlation coefficients		
		Horizontal	Vertical	Diagonal
All white	0	–	–	–
Cipher image of white	7.9971	0.0071	0.0144	−0.0068
All black	0	–	–	–
Cipher image of black	7.9970	−0.0101	0.0135	−0.0010

Table 14
Performance comparison of different methods to achieve a satisfactory security level.

Algorithms	NPCR	UACI	The round number of		
			Image-scanning	Permutation	Diffusion
Ours	> 0.996	> 0.333	1	0	1
Ref. [49]	> 0.996	> 0.333	2	2	2
Ref. [44]	> 0.996	> 0.333	4	2	2
Ref. [50]	> 0.996	> 0.333	4	4	2
Ref. [51]	> 0.996	> 0.333	6	3	3
Ref. [52]	> 0.996	> 0.333	6	3	3

Simulation results and security analyses demonstrate that the proposed algorithm has large key space, high sensitivity to the secret key and can resist well-known attacks, such as, statistical attack, differential attack, noise attack, occlusion attack, known-plaintext and chosen-plaintext attacks. All these features illustrate that our algorithm is very suitable for image and video encryption and it can be applied in the secure communication of image and video files. But the correlation coefficient computed by our algorithm is a little higher compared with some existing studies, and in the future, we will improve this encryption scheme and design other algorithms to attain the combination of high security and quick speed.

Table 15
Performance of the proposed scheme and other methods for Cameraman in size 256 × 256.

	Correlation coefficient			NPCR	UACI	Entropy
	Horizontal	Vertical	Diagonal			
Plain image	0.9592	0.9319	0.9000	–	–	7.0097
Ours	0.0010	−0.0158	0.0273	0.9957	0.3342	7.9970
Ref. [3]	0.0062	−0.0210	−0.0084	0.9962	0.3348	7.9971
Ref. [44]	0.0129	0.0118	0.0088	0.9960	0.3344	7.9969
Ref. [45]	0.0108	−0.0026	0.0160	0.9966	0.3329	7.9970
Ref. [53]	0.0108	0.0086	−0.0056	0.9962	0.3349	7.9969

Acknowledgements

All the authors are deeply grateful to the editors for careful and fast handling of the manuscript. The authors would also like to thank the anonymous referees for their valuable suggestions to improve the quality of this paper. This work is supported by the National Natural Science Foundation of China (Grant No. 41571417 and 61305042), National Science Foundation of the United States (Grant No. CNS-1253424 and ECCS-1202225), Science and Technology Foundation of Henan Province of China (Grant No. 152102210048), Foundation and Frontier Project of Henan Province of China (Grant No. 142300410126 and 162300410196), China Postdoctoral Science Foundation (Grant No. 2016M602235) and Henan Postdoctoral Scientific Program (Grant No. 2013029), Natural Science Foundation of Educational Committee of Henan Province of China (Grant No. 14A413015) and the Research Foundation of Henan University (Grant No. xxjc20140006).

References

[1] E. Galizzi Gustavo, Cuadrado-Laborde Christian, Joint transform correlator optical encryption system: Extensions of the recorded encrypted signal and its inverse fourier transform, Opt. Commun. 353 (2015) 76–82.
[2] Y. Liu, J. Tang, T. Xie, Cryptanalyzing a RGB image encryption algorithm based on DNA encoding and chaos map, Opt. Lasers Eng. 60 (2014) 111–115.
[3] G.D. Ye, A block image encryption algorithm based on wave transmission and chaotic systems, Nonlinear Dyn. 75 (2014) 417–427.
[4] X.L. Chai, An image encryption algorithm based on bit level Brownian motion and new chaotic systems, Multimed. Tools Appl. (2015). http://dx.doi.org/10.1007/s11042-015-3088-1.
[5] Amina Souyah, Kamel Mohamed Faraoun, Fast and efficient randomized encryption scheme for digital images based on Quadtree decomposition and reversible memory cellular automata, Nonlinear Dyn. 84 (2016) 715–732.
[6] H. Liu, D. Xiao, R. Zhang, Y.S. Zhang, S. Bai, Robust and hierarchical watermarking of encrypted images based on compressive sensing, Signal Process.: Image 45 (2016) 41–51.
[7] Z.Y. Hua, Y.C. Zhou, Image encryption using 2D logistic-adjusted-sine map, Inf. Sci. 339 (2016) 237–253.
[8] Safwan E.I. Assad, Mousa Farajallah, A new chaos-based image encryption system, Signal Process.: Image 41 (2016) 144–157.
[9] Adrian-Viorel Diaconu, Circular inter-intra pixels bit-level permutation and chaos-based image encryption, Inf. Sci. 3 (2015) 1–14.
[10] W. Zhang, H. Yu, Y.L. Zhao, Z.L. Zhu, Image encryption based on three-dimensional bit matrix permutation, Signal Process. 118 (2016) 36–50.
[11] N.R. Zhou, S.M. Pan, S. Cheng, Z.H. Zhou, Image compression-encryption based on hyper-chaotic system and 2D compressive sensing, Opt. Laser Technol. 82 (2016) 121–133.
[12] G.D. Ye, X.L. Huang, A secure image encryption algorithm based on chaotic maps and SHA-3, Secur. Commun. Netw. 9 (2016) 2015–2023.
[13] J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps, Int. J. Bifurcat. Chaos 8 (1998) 1259–1284.
[14] J.C. Yen, J.I. Guo, Efficient hierarchical chaotic image encryption algorithm and its VLSI realization, IEEE Proc. Vis. Image Signal Process. 147 (2000) 167–175.
[15] C.Q. Li, Cracking a hierarchical chaotic image encryption algorithm based on permutation, Signal Process. 118 (2016) 203–210.
[16] Eric Y. Xie, C.Q. Li, S.M. Yu, J.H. Lü, On the cryptanalysis of Fridrich's chaotic image encryption scheme, Signal Process. 132 (2017) 150–154.
[17] Benyamin Norouzi, Sattar Mirzakuchaki, A fast color image encryption algorithm based on hyper-chaotic systems, Nonlinear Dyn. 78 (2014) 995–1015.
[18] Benyamin Norouzi, Seyed Mohammad Seyedzadeh, Sattar Mirzakuchaki, Mohammad Reza Mosavi, A novel image encryption based on hash function with only two-round diffusion process, Multimed. Syst. 20 (2014) 45–64.
[19] Y.S. Zhang, D. Xiao, On the security of symmetric ciphers based on DNA coding, Inf. Sci. 28 (2014) 254–261.
[20] Q. Zhang, L. Guo, X.P. Wei, Image encryption using DNA addition combining with chaotic maps, Math. Comput. Model. 52 (2010) 2028–2035.
[21] X.Y. Wang, Y.Q. Zhang, X.M. Bao, A novel chaotic image encryption scheme using DNA sequence operations, Opt. Lasers Eng. 73 (2015) 53–61.
[22] Y.Q. Zhang, X.Y. Wang, J. Liu, Z.L. Chi, An image encryption scheme based on the MLNCML system using DNA sequences, Opt. Lasers Eng. 82 (2016) 95–103.
[23] R. Guesmi, M.A.B. Farah, A. Kachouri, M. Samet, A novel chaos-based image encryption using DNA sequence operation and secure hash algorithm SHA-2, Nonlinear Dyn. 83 (2016) 1123–1136.
[24] Majid Babaei, A novel text and image encryption method based on chaos theory and DNA computing, Nat. Comput. 12 (2013) 101–107.
[25] Q. Zhang, L. Guo, X. Wei, A novel image fusion encryption algorithm based on DNA sequence operation and hyper-chaotic system, Optik 124 (2013) 3596–3600.
[26] X. Huang, G. Ye, An image encryption algorithm based on hyper-chaos and DNA sequence, Multimed. Tools Appl. 72 (2014) 57–70.
[27] Q. Zhang, L. Liu, X. Wei, Improved algorithm for image encryption based on DNA encoding and multi-chaotic maps, AEU-Int, J. Electron. C 68 (2014) 186–192.
[28] L. Liu, Q. Zhang, X. Wei, A RGB image encryption algorithm based on DNA encoding and chaos map, Comput. Electr. Eng. 38 (2012) 1240–1248.
[29] H. Liu, X. Wang, Kadir Abdurahman, Image encryption using DNA complementary rule and chaotic maps, Appl. Soft Comput. 12 (2012) 1457–1466.
[30] S. Wolfram, Random sequence generation by cellular automata, Adv. Appl. Math. 7 (1986) 123–169.
[31] R.J. Chen, S.J. Horng, Novel SCAN-CA-based image security system using SCAN and 2-D von Neumann cellular automata, Signal Process.: Image 25 (2010) 413–426.
[32] P. Ping, F. Xu, Z.J. Wang, Color image encryption based on two-dimensional cellular automata, Int. J. Mod. Phys. C. 24 (2013) 1350071.
[33] P. Ping, F. Xu, Z.J. Wang, Image encryption based on non-affine and balanced cellular automata, Signal Process. 105 (2014) 419–429.
[34] X.Y. Wang, D.H. Xu, A novel image encryption scheme using chaos and Langton's ant cellular automaton, Nonlinear Dyn. 79 (2015) 2449–2456.
[35] Rasul Enayatifar, Hossein Javedani Sadaei, Abdul Hanan Abdullah, Malrey Lee, Ismail Fauzi Isnin, A novel chaotic based image encryption using a hybrid model of deoxyribonucleic acid and cellular automata, Opt. Lasers Eng. 71 (2015) 33–41.
[36] L.O. Chua, Memristors-the missing circuit element, IEEE Trans. Circuit Theory 18 (1971) 507–519.
[37] J.M. Tour, T. He, Electronics: the fourth element, Nature 453 (2008) 42–43.
[38] S.P. Adhikari, C. Yang, H. Kim, L.O. Chua, Memristor bridge synapse-based neural network and its learning, IEEE Trans. Neural Netw. Learn. Syst. 23 (2012) 1426–1435.
[39] L.D. Wang, E. Drakakis, S.K. Duan, P.F. He, X.F. Liao, Memristor model and its application for chaos generation, Int. J. Bifurc. Chaos 22 (2012) 1250205.
[40] F.Y. Yang, J.L. Leng, Q.D. Li, The 4-dimensional hyperchaotic memristive circuit based on Chua's circuit, Acta Phys. Sin. 63 (2014) 080502.
[41] I.I. Cisse, H. Kim, T. Ha, A rule of seven in Watson-Crick base-pairing of mismatched sequences, Nat. Struct. Mol. Biol. 19 (2012) 623–627.
[42] G. Àlvarez, S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, Int. J. Bifur. Chaos 16 (2006) 2129–2151.
[43] Y.Q. Zhang, X.Y. Wang, A symmetric image encryption algorithm based on mixed linear- nonlinear coupled map lattice, Inf. Sci. 273 (2014) 329–351.
[44] X.L. Chai, K. Yang, Z.H. Gan, A new chaos-based image encryption algorithm with dynamic key selection mechanisms, Multimed. Tools Appl. (2016). http://dx.doi.org/10.1007/s11042-016-3585-x.
[45] X.Y. Wang, D.H. Xu, A novel image encryption scheme based on Brownian motion and PWLCM chaotic system, Nonlinear Dyn. 75 (2014) 345–353.
[46] O. Mirzaei, M. Yaghoobi, H. Irani, A new image encryption method: parallel sub-image encryption with hyper chaos, Nonlinear Dyn. 67 (2012) 557–566.
[47] Y. Wu, Y.C. Zhou, Saveriades George, Agaian Sos, P. Noonan Joseph, Natarajan Premkumar, Local Shannon entropy measure with statistical tests for image randomness, Inf. Sci. 222 (2013) 323–342.
[48] Hung-I. Hsiao, Lee Junghsi, Color image encryption using chaotic nonlinear adaptive filter, Signal Process. 117 (2015) 281–309.
[49] Y. Wang, K.W. Wong, X. Liao, G. Chen, A new chaos-based fast image encryption algorithm, J. Appl. Soft Comput. 11 (2011) 514–522.
[50] K.W. Wong, B.S.H. Kwok, W.S. Law, A fast image encryption scheme based on chaotic standard map, Phys. Lett. A 372 (2008) 2645–2652.
[51] D. Xiao, X.W. Liao, Analysis and improvement of a chaos-based image encryption algorithm, Chaos Solitons Fract. 40 (2009) 2191–2199.
[52] Y.S. Zhang, D. Xiao, An image encryption scheme based on rotation matrix bit-level permutation and block diffusion, Commun. Nonlinear Sci. Numer. Simul. 19 (2014) 74–82.
[53] Y.C. Zhou, W.J. Cao, C.L. Philip Chen, Image encryption using binary bitplane, Signal Process. 100 (2014) 197–207.