An image encryption scheme based on three-dimensional Brownian motion and chaotic system*

Xiu-Li Chai(柴秀丽)[1,2,†], Zhi-Hua Gan(甘志华)[3], Ke Yuan(袁科)[1], Yang Lu(路杨)[4], and Yi-Ran Chen(陈怡然)[2]

[1] School of Computer and Information Engineering, Institute of Image Processing and Pattern Recognition, Henan University, Kaifeng 475004, China

[2] Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA 15261, USA

[3] School of Software, Henan University, Kaifeng 475004, China

[4] Research Department, Henan University, Kaifeng 475004, China

(Received 3 August 2016; revised manuscript received 26 November 2016; published online 10 January 2017)

At present, many chaos-based image encryption algorithms have proved to be unsafe, few encryption schemes permute the plain images as three-dimensional (3D) bit matrices, and thus bits cannot move to any position, the movement range of bits are limited, and based on them, in this paper we present a novel image encryption algorithm based on 3D Brownian motion and chaotic systems. The architecture of confusion and diffusion is adopted. Firstly, the plain image is converted into a 3D bit matrix and split into sub blocks. Secondly, block confusion based on 3D Brownian motion (BCB3DBM) is proposed to permute the position of the bits within the sub blocks, and the direction of particle movement is generated by logistic-tent system (LTS). Furthermore, block confusion based on position sequence group (BCBPSG) is introduced, a four-order memristive chaotic system is utilized to give random chaotic sequences, and the chaotic sequences are sorted and a position sequence group is chosen based on the plain image, then the sub blocks are confused. The proposed confusion strategy can change the positions of the bits and modify their weights, and effectively improve the statistical performance of the algorithm. Finally, a pixel level confusion is employed to enhance the encryption effect. The initial values and parameters of chaotic systems are produced by the SHA 256 hash function of the plain image. Simulation results and security analyses illustrate that our algorithm has excellent encryption performance in terms of security and speed.

Keywords: image encryption, logistic-tent system (LTS), memristive chaotic system, three-dimensional (3D) Brownian motion

PACS: 05.45.Gg, 05.45.–a, 05.45.Vx, 05.40.Jc **DOI:** 10.1088/1674-1056/26/2/020504

1. Introduction

With the wide development of internet technology and social networking services (SNS), more and more digital images, videos are transmitted and stored online, therefore the security of multimedium data has received much attention. Image encryption is an effective method to secure the digital images. Traditional encryption schemes, for example, data encryption standard (DES), advanced encryption standard (AES) and Ron Rives Adi Shamir Leonard Adleman (RSA) are not employed to effectively encrypt images because of the intrinsic features of the images, such as high correlation between adjacent pixels, bulk data capacity and high redundancy. Chaotic systems are characterized by sensitivity to initial conditions and parameters, ergodicity, pseudo randomness and topological transitivity, thus they are very suitable for image encryption.[1–3]

Since Matthews presented the first chaos-based encryption algorithm in 1989,[4] many chaos-based image encryption methods have been introduced.[4–10] For example, a fast image encryption algorithm based on only blocks in cipher text was given in Ref. [5], and Logistic map and skew tent map were used for constructing chaotic sequences. Zhou *et al.*[6] introduced a general chaotic framework named as cascade chaotic system (CCS), generated many new chaotic maps by using two one-dimensional (1D) chaotic maps as seed maps based on this framework, these newly generated chaotic maps have more unpredictable and better chaotic performance, and they have been applied to image encryption. Ye[7] presented a block-based image encryption algorithm by using wave function and chaotic system, and the random sequence produced by Chen chaotic system was employed to obtain the source point in the wave and give a diffusion matrix for modular operation. Recently, a novel chaos-based image encryption scheme based on compressive sensing was introduced,[8] and one-dimensional skew tent map was used to generate the measurement matrix. In view of the secure problems of low-dimensional chaos in image encryption and cycle degradation caused by computer

*Project supported by the National Natural Science Foundation of China (Grant Nos. 41571417 and 61305042), the National Science Foundation of the United States (Grant Nos. CNS-1253424 and ECCS-1202225), the Science and Technology Foundation of Henan Province, China (Grant No. 152102210048), the Foundation and Frontier Project of Henan Province, China (Grant No. 162300410196), China Postdoctoral Science Foundation (Grant No. 2016M602235), the Natural Science Foundation of Educational Committee of Henan Province, China (Grant No. 14A413015), and the Research Foundation of Henan University, China (Grant No. xxjc20140006).

†Corresponding author. E-mail: chaixiuli@henu.edu.cn

© 2017 Chinese Physical Society and IOP Publishing Ltd

finite precision, Zhang and Wang[9] put forward a novel image encryption algorithm by using a mixed linear-nonlinear coupled map lattice, and Tong *et al.*[10] designed a perturbed high-dimensional chaos according to Devaney and topological conjugate definition, and employed it for image encryption as pseudo-random sequence generators.

However, a number of encryption algorithms have illustrated security problems and proved to be unsafe.[11–25] In 2000, an efficient hierarchical chaotic image encryption (HCIE) algorithm was presented,[11] and it divided the plain image into blocks with the same size, and then operated position permutation at intra-block and inter-block levels. Recently, this algorithm has been cracked, and the security of HCIE against ciphertext-only attack is much lower, and it should be combined with confusion to provide high security.[12] Li *et al.*[13] gave a novel double-image encryption scheme by using chaos-based local pixel scrambling technique in gyrator transform (GT) domains, and Chen *et al.*[14] found that there is serious cross-talk disturbance in the decrypted phase-based image when the cipher image undergoes noise perturbation or occlusion attack. Liu *et al.*[15] evaluated the security of the encryption algorithm in Ref. [16], and found that it can be broken with only two pairs of known plain image and corresponding cipher images, pointing out that it has some security issues, such as insensitivity to the change of the plain image and secret keys. Zhang and Liu[17] proposed a novel image encryption method by using the permutation-diffusion architecture and skew tent chaotic map. Wang and He[18] found that this method had two flaws including that the key was fixed regardless of the plain image, and the permutation vector was constant for different original images, and it was vulnerable for chosen plaintext attack. Subsequently Eslami and Bakhshandeh[19] gave an improved method based on Ref. [17], and recently Akhavan *et al.*[20] cryptanalyzed this improved algorithm by using differential attack and found that it was not sensitive enough to the plain image. Recently, Wang *et al.*[21] analyzed a parallel sub-image encryption scheme combined with high-dimensional chaotic system in Ref. [22], and found that the produced keystream was unchanged for different original images, and the plain image may be recovered through chosen-plaintext attack. Li *et al.*[23] reevaluated the security of a novel image encryption algorithm in Ref. [24] and found that the scheme can be broken with only two known plain images, and they also cryptanalyzed the Fridrich's chaotic image encryption scheme.[25] By analyzing these papers, the main flaw is that these encryption schemes have not enough sensitivity to the plain image, "one original image, one secret key" cannot be achieved, and they have less resistance to chosen-plaintext and known-plaintext attacks. Therefore, more secure and efficient image encryption methods are needed to be introduced and employed in secure communication field.

The classical encryption process consisting of confusion and diffusion was first introduced by Shannon,[26] and afterwards many image encryption schemes were designed based on it.[1,2,4–8,27–41] There are two methods employed in the confusion process. One is pixel level permutation[27–29] and the other is bit level confusion.[30–32] The former can change the positions of image pixels while the values of pixels are unchanged, and the latter can not only permute the positions of pixels, but also modify the values of the image pixels. Thus, bit level confusion is considered to be a perfect permutation method for the better encryption efficiency,[33] and a great many of image encryption schemes using bit level permutation have been introduced. For example, a new permutation method to confuse and diffuse the grayscale image at the bit level was proposed in Ref. [34], Arnold cat map was used to permute the bits, and logistic map was adopted for diffusing the permuted image. In view of the limits of Arnold cat map, it requires that the image must be square. Besides, logistic map has a dense set of periodic windows for a certain range of parameters, which restricts its application in image encryption. To alleviate the problem, Liu and Wang[35] presented a color image encryption method using bit-level permutation and high-dimensional chaotic system. Firstly, a color image ($M \times N$) was converted into a grayscale image ($M \times 3N$) through R, G, B splitting, and next a binary matrix was obtained by bit plane decomposition; secondly chaotic sequence generated from piecewise linear chaotic map (PWLCM) was employed to permute the two-dimensional (2D) bit matrix; thirdly, Chen's chaotic system was used for diffusion. Recently, Martin *et al.*[36] gave a color image encryption algorithm based on three-dimensional (3D) cellular automata and chaotic maps, and 24 2D bit planes could be obtained from the bit plane decomposition of the color image, the 2D Cat map was used to permute the positions in each bit plane, and 3D cellular automata was employed for diffusion. In the bit level permutation process, after bit plane expansion of the original image, the 2D image is converted into a 3D bit matrix (the width, height and bit length), and each bit plane has its own weight. A majority of bit level confusion algorithms were manipulated on the 2D bit planes,[37–40] they decomposed the plain image into many bit planes. The confusion process of the plain image was transformed into that of the bit plane, more bit planes arose and more confusion steps were done. Moreover, some scholars have presented heterogeneous bit level confusion methods in view of the different importance of bit planes.[41] But in the final analysis, it was the confusion scheme of the 2D matrix or 1D vector from the 2D matrix. Few image encryption schemes confuse the original image as a 3D bit matrix, and thus bits cannot move to any position and any bit plane and the movement range of bits are limited. Thus, we may consider the original image as a 3D bit matrix, and then

study the method to make the bit elements randomly move to any position in the matrix, thereby enhancing the permutation effect and upgrading the security level of the encryption scheme.

Based on the above analyses, a new image encryption algorithm using 3D Brownian motion and chaotic systems is introduced in the paper. Brownian motion is a random movement form of particles, generally existing in the liquid and gas, particles move fast in the 3D space, and the movement traces distribute in the whole space. Our contributions are as follows: first, we employ the 3D Brownian motion to confuse the bits in the 3D bit matrix, where the bits are regarded as Brownian motion particles, and use the Monte Carlo method to simulate Brownian motion. Due to the increase of the image size, the 3D bit matrix is split into many sub blocks to save the execution time, a new block confusion based on 3D Brownian motion (BCB3DBM) is given to confuse the positions of the bits within the blocks, and logistic-tent system (LTS) is employed to generate the direction of particle movement. Besides, block confusion based on position sequence group (BCBPSG) is introduced to confuse the positions of the sub blocks, random chaotic sequences are produced by a four-order memristive chaotic system and sorted, a proper position sequence group is picked out according to the plain image and utilized to change the position of the sub block. Our confusion method can fully change the positions of the bits and their weights, the bits in one bit plane can appear at any position and in any bit plane theoretically, and it greatly increases the encryption performance. Furthermore, diffusion is adopted to increase the security of the algorithm. In order to improve the algorithm sensitivity to the plain image and the ability to resist the known-plaintext and chosen-plaintext attacks, the initial values and parameters of chaotic systems are obtained by the SHA 256 hash value of the original image.

The rest of the present paper is organized as follows. In Section 2, preliminary studies including a four-order memristive chaotic system, LTS and 3D Brownian motion are described. Section 3 gives our encryption scheme in detail. Section 4 presents the simulation results, and security performances of the proposed algorithm are analyzed in Section 5. Conclusions are drawn in the last section.

2. Preliminary studies

2.1. A four-order memristive chaotic system

The concept of memristor was initially presented by Chua in 1971,[42] it is believed to be the fourth passive circuit element, and other three elements are resistor, capacitor and inductor. In recent years, the studies of memristor-based dynamical systems have been receiving more and more attention because of their potential applications in signal processing, neural networks, image encryption and other fields.[43–45] The memristive chaotic systems have complex dynamical characteristics and close dependence on the initial values and system parameters, and they are desirable for image encryption.

Chua's diode with Chua's circuit was replaced with an active memristive circuit, and it was composed of a negative conductance and a flux-controlled memrisor, therefore a novel four-order memristive system was presented,[46] and it can be derived as follows:

$$\begin{cases} \dot{x}(t) = \alpha(y(t) - x(t) - Q(w(t))x(t) + \gamma x(t)), \\ \dot{y}(t) = x(t) - y(t) - z(t) + py(t), \\ \dot{z}(t) = \beta y(t) - kz(t), \\ \dot{w}(t) = x(t), \end{cases} \tag{1}$$

where

$$Q(w(t)) = \begin{cases} a - 2bw(t) + 3cw(t)^2, & w(t) < 0, \\ a + 2bw(t) + 3cw(t)^2, & w(t) > 0. \end{cases} \tag{2}$$

And when $\alpha = 9$, $\beta = 13$, $\gamma = 2$, $p = 0.003$, $a = 0.8$, $b = -0.03$, $c = 0.01$, and $k = 0.05$, the system is chaotic with a positive Lyapunov exponent $LE = 0.185$, and its attractor diagram is shown in Fig. 1. From it, we can see that the memristive chaotic system has a good chaotic dynamic behavior.

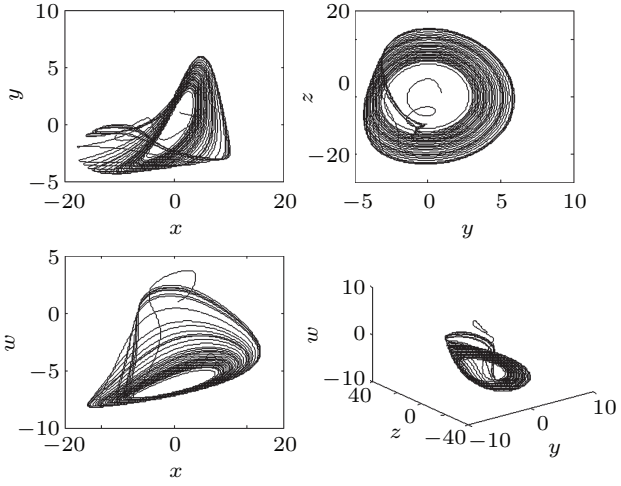


Fig. 1. Attractor diagram of Chua's system.

2.2. LTS

Utilizing the logistic map and tent map as seed maps, LTS is presented in the following equation[47]

$$\begin{aligned} & u_{n+1} \\ &= \mathrm{LTS}(r, u_n) \\ &= \begin{cases} (ru_n(1-u_n) + (4-r)u_n/2) \bmod 1, & u_n < 0.5, \\ (ru_n(1-u_n) + (4-r)(1-u_n)/2) \bmod 1, & u_n \geq 0.5, \end{cases} \end{aligned} \tag{3}$$

where system parameter $r \in (0, 4]$. Figure 2 illustrates the bifurcation diagram of LTS. From Fig. 2, it can be found that LTS has a wider chaotic range than logistic and tent maps, and the output sequences of LTS spread out in an entire chaotic range between 0 and 1.

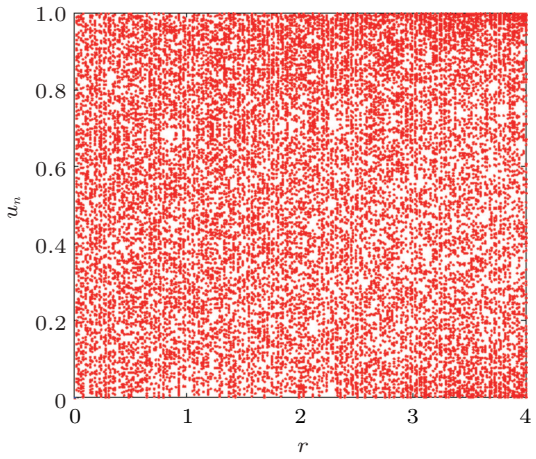


Fig. 2. (color online) Bifurcation diagram of LTS.

2.3. 3D Brownian motion

Brownian motion is a random motion of particles suspended in a fluid (a liquid or a gas) resulting from their collisions with the fast-moving atoms or molecules in the gas or liquid. A point in the space may be shown in Fig. 3 by the following equation in the spherical coordinates:

$$X = r\sin a\cos b,\ \ Y = r\sin a\sin b,\ \ Z = r\cos a, \tag{4}$$

where $0 \le r \le +\infty$, $0 \le b \le 2\pi$, and $0 \le a \le \pi$.

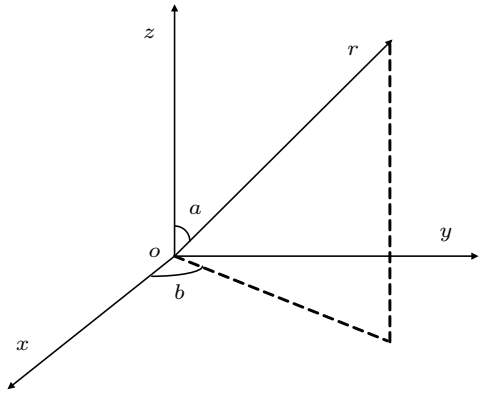


Fig. 3. An point in the three-dimensional space.

One point in the space is regarded as one Brownian particle. When the direction of movement and step length are given, the position of Brownian particle can be determined. Here, the step length is chosen as $r = 2$, the random number generator function is employed to calculate the direction of movement, then the x, y, z components of the position of each Brownian particle can be obtained. Brownian motion can be simulated through Monte Carlo method. We simulate the Brownian motion of 200 particles in the 3D space, a matrix of 200×3 is utilized to store the positions of Brownian particles, and the x, y, z components are limited in a region of $[-20, 20]$. Figure 4(a) shows the positions of the 200 particles at some time, and the movement trace of one particle is illustrated in Fig. 4(b). Figure 4(a) displays that the particles have evenly distributed in the space, and Fig. 4(b) shows that the movement of the particle has strong randomness at different times.

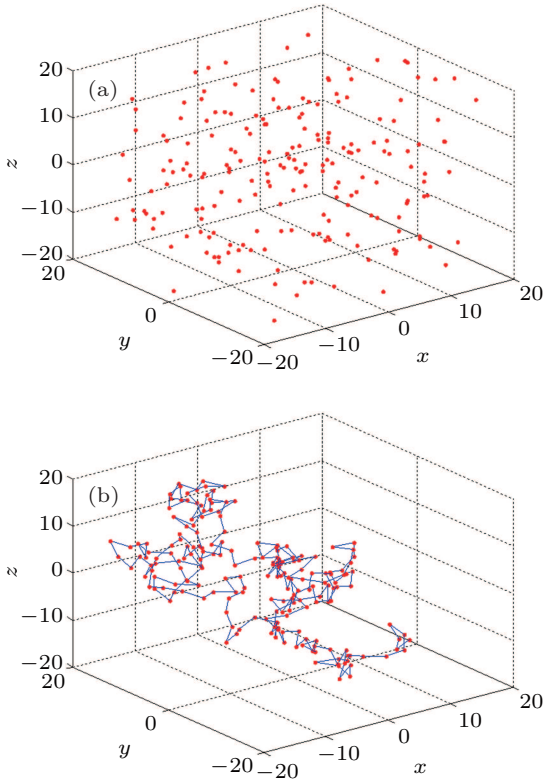


Fig. 4. (color online) Simulation results of 3D Brownian motion, showing (a) the positions of the 200 particles at some time, and (b) the movement trace of one particle.

3. Encryption scheme

3.1. Generation of the initial values and parameters of chaotic systems

In the proposed algorithm, SHA 256 hash function is used to generate the initial values and control parameters of the chaotic systems. Before encrypting the plain image, its SHA 256 hash value is computed as the 256-bit secret key K, and it can be denoted by

$$K = k_1, k_2, \ldots, k_{32},\ \ \text{subject to}:\ \ k_i = \{k_{i,0}, k_{i,1}, \ldots, k_{i,7}\}, \tag{5}$$

where in $k_{i,j}$, i refers to the character number and j denotes the bit number in k_i, and $1 \le i \le 32$, $0 \le j \le 7$.

In the encryption process, two LTSs are employed to determine the direction of Brownian particle movement, and another two LTSs are required in the confusion process to modify the pixel values of the plain image. Thus, four initial values and four parameters for LTSs are derived as follows:

$$t_i = \left(\frac{k_{2i-1} + (2i-1)}{k_{2i} + 2i} \times \sum_{j=1}^{16} \frac{k_j \times 2^{8(j-1)}}{2^{128}}\right) \bmod 1,$$
$$i = 1, 2, \ldots, 8, \tag{6}$$

where u_{10}, r_{10}, u_{20}, r_{20}, u_{30}, r_{30}, u_{40}, and r_{40} are equal to t_1, t_2, t_3, t_4, t_5, t_6, t_7, and t_8, respectively.

In addition, the initial values for the memristive chaotic system can be computed by

$$x_0 = (k_{17} \oplus k_{18} \oplus k_{19} \oplus k_{20}) \bmod 2^8/255, \quad (7)$$

$$y_0 = (k_{21} \oplus k_{22} \oplus k_{23} \oplus k_{24}) \bmod 2^8/255, \quad (8)$$

$$z_0 = (k_{25} \oplus k_{26} \oplus k_{27} \oplus k_{28}) \bmod 2^8/255, \quad (9)$$

$$w_0 = (k_{29} \oplus k_{30} \oplus k_{31} \oplus k_{32}) \bmod 2^8/255, \quad (10)$$

where $x \oplus y$ denotes the XOR operation.

According to Eqs. (5)–(10), the initial values and parameters of chaotic systems depend on the hash value of the original image. When there is a slightly change in the plain image, a completely different hash value will appear, thus, the initial values and parameters of chaotic systems will change greatly. So, our algorithm is greatly sensitive to the secret key and it can resist any brute force attack with a total complexity of 2^{128}.

3.2. Confusion

In order to improve the encryption speed, the confusion process of the plain image is manipulated by block. Our confusion strategy consists of block confusion based on 3D Brownian motion (BCB3DBM) and block confusion based on position sequence group (BCBPSG).

Firstly, by splitting the bit planes, the plain image P with a size of $M \times N$ is converted into a 3D bit matrix P_1 with the size $M \times N \times 8$, and the width r_1, the height r_2 and the bit length r_3 are M, N, and 8, respectively. Secondly, transforming the bit matrix P_1 into a cubic matrix P_2, the sizes are all r, and $r^3 = M \times N \times 8$. For example, when $M = N = 512$, then the size of the cubic matrix is $128 \times 128 \times 128$. The transformation of a 3D bit matrix into a cubic matrix may reduce the iterative times of memristive chaotic systems and save the permutation time. Secondly, splitting the cubic matrix P_2 into many 3D sub blocks, the size of every sub block is $m \times m \times m$, and there are n^3 blocks ($n = r/m$). Then, we use Brownian motion to permute the positions of the bits for each sub block.

If the 3D bit matrix of the original image cannot be transformed directly into a cubic matrix, a cubic matrix may be obtained through adding more pixels to the plain image.

3.2.1. Block confusion based on 3D Brownian motion

Block confusion based on 3D Brownian motion (BCB3DBM) is employed to confuse the positions of bits within each sub block. The detailed steps are as follows.

Step 1 Iterating the LTS a finite number of times with initial values u_{10} and parameter r_{10}, the chaotic sequence U_1 is obtained; then iterating the LTS with initial values u_{20} and parameter r_{20}, and the chaotic sequence U_2 is obtained. The two sequences can be shown as follows:

$$U_1 = \{u_{11}, u_{12}, \ldots, u_{1m^3}\}, \ U_2 = \{u_{21}, u_{22}, \ldots, u_{2m^3}\}. \quad (11)$$

Step 2 Set the step length r of the particle movement to be 2, the direction of movement is generated by two LTSs, and the two angles in polar coordinates for any particle i can be acquired by the following equation

$$b = u_{1i} \times 2 \times \pi, \quad a = u_{2i} \times \pi, \ \ i = 1, 2, \ldots, m^3. \quad (12)$$

Step 3 According to Eqs. (4), (11), and (12), x, y, z components of the current moving distance for each particle can be obtained. Repeating the movement for R rounds, we can obtain the final movement distance of every particle.

Step 4 Regard the bit of a 3D bit matrix as a particle, and establish a one-to-one mapping between the position of the particle and that of the bit. For example, after R round Brownian motion, the x, y, z components of the particle located at (i, j, k) are $l_i(1)$, $l_j(2)$, $l_k(3)$, respectively; then change the position of the bit located at (i, j, k) as $x = l_i(1)$, $y = l_j(2)$, $z = l_k(3)$.

When they move to the same location, in other words, the final locations of the Brownian motion particles are the same, the former particle (named particle 1) in the 3D matrix moves to this place, the latter particle (named particle 2) is picked out, and when the movements of other particles have finished, the particle 2 moves to the empty position in 3D bit matrix. When particle 2 represents more particles than one, they are shifted to the empty positions according to the order of sequence in the 3D matrix.

3.2.2. Block confusion based on position sequence group

After BCB3DBM is applied to each sub block, the sub block image is regarded as a unit, its position is permutated based on position sequence group. The detailed steps of block confusion based on position sequence group (BCBPSG) are shown below.

Step 1 Iterate the memristive chaotic system $(n+l)$ times using the initial values x_0, y_0, z_0, and w_0, then discard the former l ($l \geq 500$) numbers to avoid the transient effect, state variables x_i, y_i, z_i, and w_i can be obtained, and then the four integer sequences X, Y, Z, W are produced as follows:

$$\begin{cases} X = \{x_1, x_2, \ldots, x_n\}, \\ Y = \{y_1, y_2, \ldots, y_n\}, \\ Z = \{z_1, z_2, \ldots, z_n\}, \\ W = \{w_1, w_2, \ldots, w_n\}, \end{cases} \quad (13)$$

where

$$\begin{cases} x_i = \lfloor (abs(x_i) - \lfloor x_i \rfloor) \times 10^{14} \rfloor \bmod 2^{10}, \\ y_i = \lfloor (abs(y_i) - \lfloor y_i \rfloor) \times 10^{14} \rfloor \bmod 2^{10}, \\ z_i = \lfloor (abs(z_i) - \lfloor z_i \rfloor) \times 10^{14} \rfloor \bmod 2^{10}, \\ w_i = \lfloor (abs(w_i) - \lfloor w_i \rfloor) \times 10^{14} \rfloor \bmod 2^{10}, \end{cases} \quad (14)$$

and $\lfloor x \rfloor$ returns to the nearest integer for x.

Step 2 Sort the sequences X, Y, Z, W in ascending order, then obtain four position index sequences $\bar{X}$, $\bar{Y}$, $\bar{Z}$, $\bar{W}$, $\bar{X} = \{k_1^x, k_2^x, \ldots, k_n^x\}$, $\bar{Y} = \{k_1^y, k_2^y, \ldots, k_n^y\}$, $\bar{Z} = \{k_1^z, k_2^z, \ldots, k_n^z\}$, $\bar{W} = \{k_1^w, k_2^w, \ldots, k_n^w\}$.

Step 3 Divide the position sequences $\bar{X}$, $\bar{Y}$, $\bar{Z}$, $\bar{W}$ into groups $A_i (i = 1, 2, \ldots, 24)$, and they can be denoted as $A_1 = [\bar{X}, \bar{Y}, \bar{Z}]$, $A_2 = [\bar{X}, \bar{Y}, \bar{W}]$, $A_3 = [\bar{X}, \bar{Z}, \bar{Y}]$, $A_4 = [\bar{X}, \bar{Z}, \bar{W}]$, $A_5 = [\bar{X}, \bar{W}, \bar{Y}]$, $A_6 = [\bar{X}, \bar{W}, \bar{Z}]$; $A_7 = [\bar{Y}, \bar{X}, \bar{W}]$, $A_8 = [\bar{Y}, \bar{X}, \bar{Z}]$, $A_9 = [\bar{Y}, \bar{Z}, \bar{X}]$, $A_{10} = [\bar{Y}, \bar{Z}, \bar{W}]$, $A_{11} = [\bar{Y}, \bar{W}, \bar{X}]$, $A_{12} = [\bar{Y}, \bar{W}, \bar{Z}]$; $A_{13} = [\bar{Z}, \bar{X}, \bar{Y}]$, $A_{14} = [\bar{Z}, \bar{X}, \bar{W}]$, $A_{15} = [\bar{Z}, \bar{Y}, \bar{X}]$, $A_{16} = [\bar{Z}, \bar{Y}, \bar{W}]$, $A_{17} = [\bar{Z}, \bar{W}, \bar{X}]$, $A_{18} = [\bar{Z}, \bar{W}, \bar{Y}]$; $A_{19} = [\bar{W}, \bar{X}, \bar{Y}]$, $A_{20} = [\bar{W}, \bar{X}, \bar{Z}]$, $A_{21} = [\bar{W}, \bar{Y}, \bar{X}]$, $A_{22} = [\bar{W}, \bar{Y}, \bar{Z}]$, $A_{23} = [\bar{W}, \bar{Z}, \bar{X}]$, and $A_{24} = [\bar{W}, \bar{Z}, \bar{Y}]$.

Step 4 Select the group A_i according to the first pixel value $P(1)$ of the plain image. Set index = $((P(1) \times 1000) \bmod 24) + 1$, when index = i, then group A_i is chosen to confuse the sub block image.

Step 5 Establish the mapping rule between the group A_i and the block images. Assume that the current block image is located at ($b1$, $b2$, $b3$), A_1 is picked according to Step 4 with index = 1, then the mapping rule is ($b1$, $b2$, $b3$) $\rightarrow (k_{b1}^x, k_{b2}^y, k_{b3}^z)$; when A_2 is chosen, the mapping rule is ($b1$, $b2$, $b3$) $\rightarrow (k_{b1}^x, k_{b2}^y, k_{b3}^w)$.

As is well known, there are different bit weights at different bit planes. Through BCB3DBM and BCBPSG, each bit is moved to other bit planes and its weight is also changed. That is to say, the position and weight of any bit are changed simultaneously, and therefore our confusion scheme has a good diffusion effect. Besides, in BCBPSG, 24 different position sequence groups are produced, a proper group depending on the plain image is chosen to confuse the sub blocks. Therefore, the confusion process has strong sensitivity to the plain image, and our algorithm has higher security level.

3.3. Diffusion

Diffusion is performed at the pixel level. Firstly, the confused bit matrix is transformed into a 2D pixel matrix, which is subsequently resized to a 1D pixel array $P_3(1 \times MN)$. Secondly, iterate the LTS using the initial values and parameters u_{30}, r_{30}, u_{40}, and r_{40}, sequences U_3, U_4, and U are obtained by the following equation:

$$\begin{cases} U_3 = \{u_3(1), u_3(2), \ldots, u_3(MN)\}, \\ U_4 = \{u_4(1), u_4(2), \ldots, u_4(MN)\}, \\ U = \{u(1), u(2), \ldots, u(MN)\}, \end{cases} \tag{15}$$

where

$$\begin{cases} u_3(i) = \lfloor (abs(u_3(i)) - \lfloor u_3(i) \rfloor) \times 10^{14} \rfloor \bmod 2^{10}, \\ u_4(i) = \lfloor (abs(u_4(i)) - \lfloor u_4(i) \rfloor) \times 10^{14} \rfloor \bmod 2^{10}, \\ u(i) = (u_3(i) + u_4(i)) \bmod 2^8, i = 1, 2, \ldots, MN. \end{cases} \tag{16}$$

Next, perform diffusion by Eqs. (17) and (18), and the cipher image C is obtained as follows:

$$C(i) = P_3(i) \oplus u(i) \oplus C(i-1), \quad i = 1, 2, \ldots, MN, \tag{17}$$

$$C(0) = P_3(MN) \oplus u(MN), \tag{18}$$

where, $P_3(MN)$ and $u(MN)$ denote the MN-th elements of arrays P_3 and U, respectively; $P_3(i)$ indicates the i-th element of array P_3; $C(i)$ and $C(i-1)$ are the i-th and $(i-1)$-th elements of cipher image C, respectively.

3.4. Complete encryption steps

As shown in Fig. 5, the whole encryption steps are described as follows.

Step 1 Obtain the external key K through the SHA 256 hash function of the plain image, and then compute the initial values and parameters of the chaotic systems by using Eqs. (5)–(10).

Step 2 Transform the plain image P $(M \times N)$ into a cubic matrix $P_2(r \times r \times r)$, then split P_2 into many 3D sub blocks, the size of each block is $m \times m \times m$, and there are n^3 sub blocks $(n = r/m)$.

Step 3 Perform BCB3DBM by using two LTSs within each sub block as described by Subsubsection 3.2.1.

Step 4 Manipulate BCBPSG by using the four-order memristive chaotic system to all the sub blocks as shown in Subsubsection 3.2.2.

Step 5 Transform the confused image into a 1D array $P_3(1 \times MN)$, and perform diffusion by using other two LTSs as illustrated in Subsection 3.3, and the cipher image C is obtained.

3.5. Discussion

The proposed encryption scheme has the following merits.

Firstly, SHA 256 hash function of the plain image is employed to generate the secret key, a four-order memristive chaotic system and LTS are used to produce the chaotic sequences for confusion and diffusion, initial values and parameters of the chaotic systems are obtained from the secret key, and therefore the encryption algorithm is highly sensitive to the plain image and it can withstand chosen-plaintext and known-plaintext attacks.

Secondly, we adopt the encryption architecture of confusion and diffusion. The confusion strategy is composed of block confusion based on 3D Brownian motion (BCB3DBM) and block confusion based on position sequence group (BCBPSG). Before encryption, the plain image is transformed

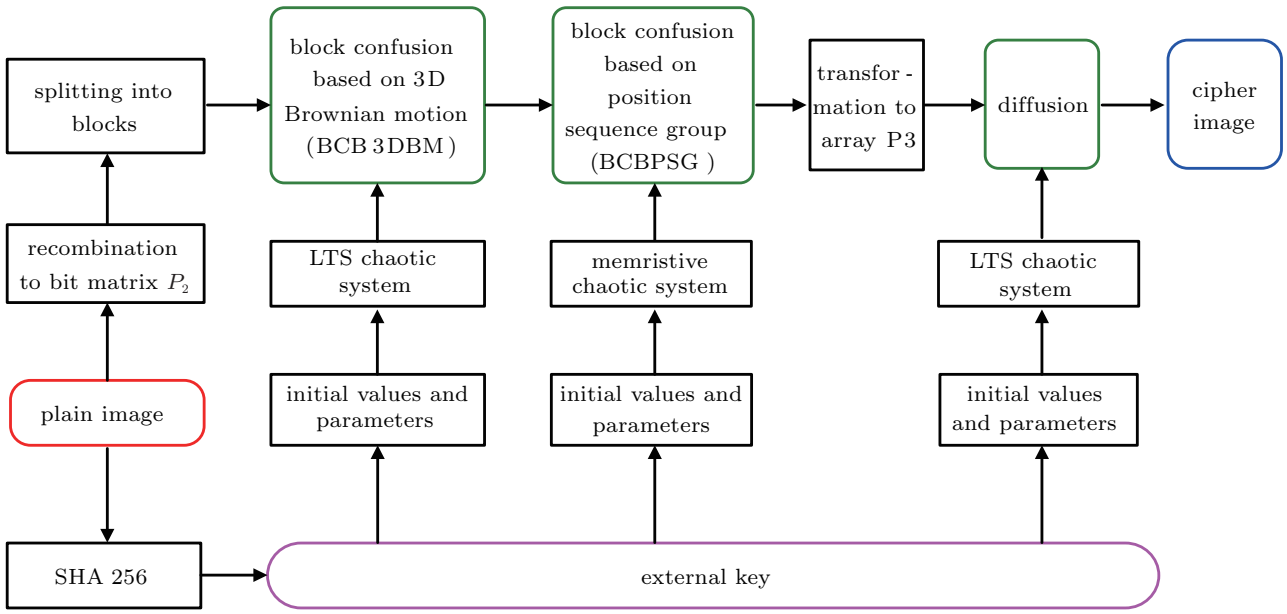


Fig. 5. Diagram of the encryption algorithm.

into a 3D bit matrix through bit plane decomposition. In order to improve encryption speed, the 3D bit matrix is divided into many non-overlapping sub blocks, strongly random 3D Brownian motion is used to confuse the position of the bits for the first time, Monte Carlo method is used to simulate Brownian motion, and bit elements are regarded as Brownian motion particles.

Thirdly, the subsequent BCBPSG is used to confuse the position of the sub blocks based on position sequence groups generated by a four-order memristive chaotic system, and the confusion process is controlled by the plain image. Thus, the proposed confusion method can attain the permutation of inter and intra sub blocks, which makes the bits in one bit plane shift to any position and any bit plane in theory, so it has good confusion effect.

Finally, in order to upgrade the encryption effect, the diffusion step at pixel level is performed on the confused image, the LTS is used to produce the key sequences, and we obtain the cipher through manipulating the XOR operation on the current image pixel, previous cipher pixel and key sequence element.

4. Simulation results

In this section, several experiments are conducted on multiple images, Lena, Baboon and Peppers of size 512×512 are chosen as the original images, and all the experiments are performed on a personal computer with the following hardware environment: 2.5-GHz CPU, 4-GB memory, and Windows 7 operating system, and the compiler software is Matlab 2014a. Some keys we use here are as follows: $\alpha = 9$, $\beta = 13$, $\gamma = 2$, $p = 0.003$, $a = 0.8$, $b = -0.03$, $c = 0.01$, and $k = 0.05$, the Brownian movement rounds $R = 100$, the size of every sub block image of $8 \times 8 \times 8$ and the iterating parameter of the memristive chaotic system, $l = 500$. Figures 6(a), 6(d), and 6(g) are the plain images, and after BCB3DBM and BCBPSG, we obtain the confused images Figs. 6(b), 6(e), and 6(h), and figures 6(c), 6(f), and 6(i) are their corresponding cipher images.

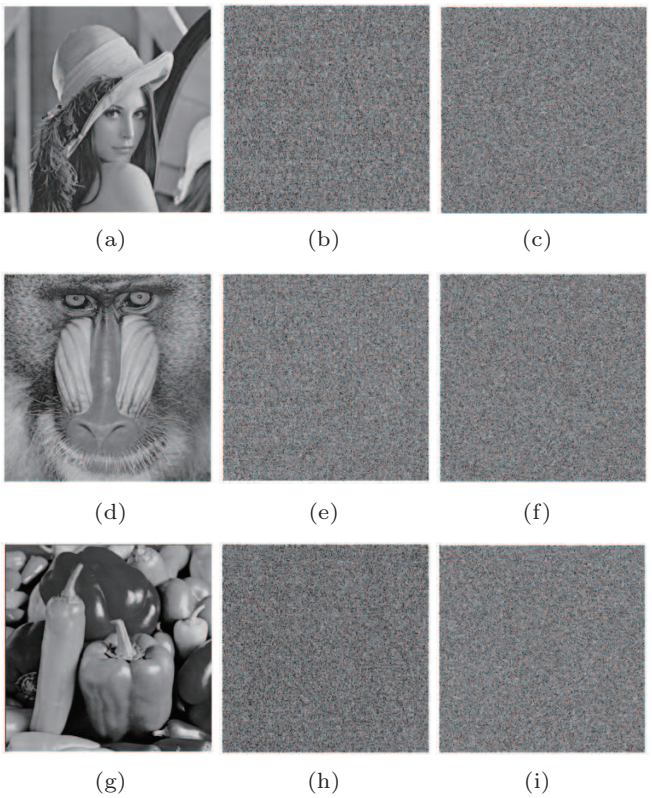


Fig. 6. (color online) Encryption results. showing (a) original image of Lena, corresponding (b) confused image, and (c) cipher image; (d) original image of Baboon, corresponding (e) confused image, and (f) cipher image; (g) original image of Peppers, corresponding (h) confused image, and (i) cipher image.

From the results, we can conclude that the confused images and cipher images are similar to noise images, any useful information cannot be obtained from them, and thus the proposed encryption algorithm has a good encryption effect and it can be applied to the secure transmission of private information.

5. Security analyses

5.1. Key space analysis

A good encryption algorithm should have a large enough key space to make the brute force attacks infeasible. In our algorithms, the secure key includes (i) the 256-bit long SHA 256 hash value; (ii) the Brownian movement rounds R; (iii) iterating parameter l of the memristive chaotic system; (iv) the first pixel value $P(1)$ of the plain image. It is known that the key space of SHA 256 with complexity of the best attack is 2^{128} larger than 2^{100},[48] which means that our algorithm is enough to resist any brute force attack.

5.2. Histogram analysis

The histogram is used to reflect the distribution of the pixels. The uniform distribution can hide the information about the plain image. Figure 7 illustrates the histograms of the plain images (Lena, Baboon, and Peppers), their corresponding confused images and cipher images, respectively. It can be found that the histograms of the plain images are statistically significant, those of the confused images are more uniform, and those of the cipher images are the most uniformly distributed.

Besides, variances of histograms are used to quantitatively evaluate the uniformity of the images. When the variance is lower, and the uniformity of the image is higher. The variance of histograms can be obtained by the following equation:[9]

$$\mathrm{var}(Z) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{2}(z_i - z_j)^2. \tag{19}$$

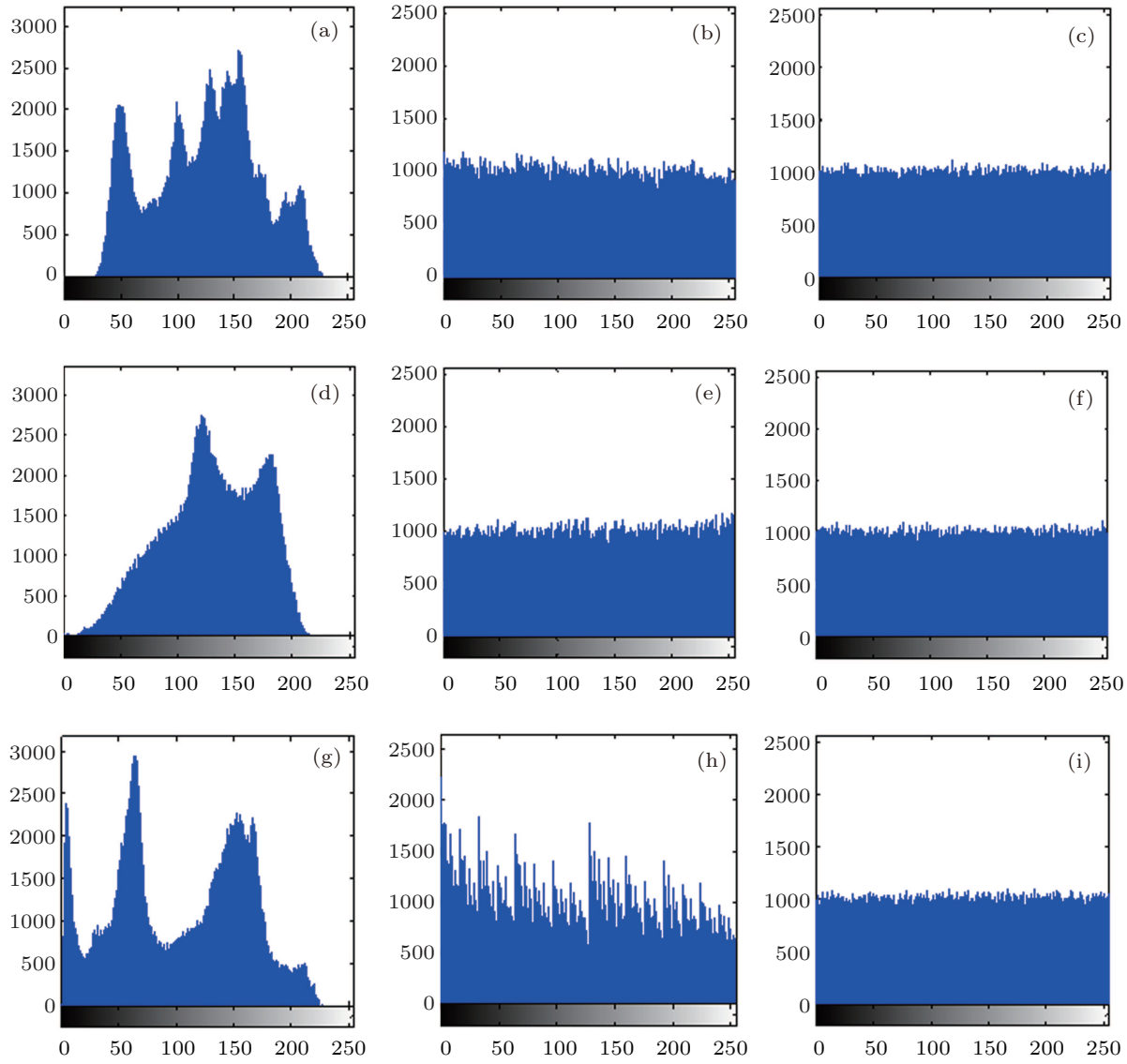


Fig. 7. (color online) Analyses of histograms of (a) Lena plain, corresponding (b) confused image, and (c) cipher image; histograms of (e) Baboon plain, corresponding (e) confused image, and (f) cipher image; histograms of (g) Peppers plain, (h) confused image, and (i) cipher image.

Here, $Z = \{z_1, z_2, \ldots, z_{256}\}$ is the vector of the histogram value, and z_i and z_j are the numbers of pixels of which the gray values are equal to i and j, respectively. For Lena (512×512), the variances of the histograms of the plain image, confused image and cipher image are shown in Table 1. From the results, we can see that the variance of the cipher image is the lowest, it is 970.11, the second lowerest one is that of confused image (that is, 4251.7), the corresponding value of the plain image is 632100. Moreover, variance of cipher image of Lena in our algorithm is much less than that of Ref. [34], and thus the proposed algorithm is more efficient and secure.

Table 1. Variances of the histograms of the Lena (512×512).

Algorithm	Ours			Ref. [34]
Images	Plain image	Confused image	Cipher image	Cipher image
Variances	632100	4251.7	970.11	5554.82

5.3. Information entropy analysis

Information entropy can be calculated from

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i) \log \frac{1}{p(m_i)}, \tag{20}$$

where $p(m_i)$ denotes the probability of symbol m_i. For a random image with 256 gray levels, the entropy should ideally be 8.[49] The closer to 8 the entropy, the more confused the image is. Table 2 gives the information entropies of some images. In Table 2, the confused and cipher images are obtained by our algorithm, and the entropies are shown in third and fourth columns, respectively. We can see that the entropies of the confused images are more than 7.9, those of cipher images are larger than 7.999, which implies that our algorithm is better than that in Ref. [32] and comparable to those in Refs. [7] and [50].

Table 2. Information entropies of the plain and cipher images.

Name	Plain image	Confused image	Cipher image	Ref. [7]	Ref. [50]	Ref. [32]
Lena (512×512)	7.4456	7.9971	7.9993	7.9993	7.9993	7.9984
Peppers (512×512)	7.5715	7.9544	7.9993	7.9998	7.9997	7.9986
Baboon (512×512)	7.3579	7.9979	7.9993	7.9992	7.9992	7.9984
Barbara (512×512)	7.4664	7.9927	7.9993	7.9991	7.9992	7.9983
Goldhill (512×512)	7.4778	7.9980	7.9993	7.9993	7.9992	7.9981
Aerial (512×512)	6.9940	7.9522	7.9993	7.9991	7.9992	7.9979

5.4. Correlations between two adjacent pixels

Adjacent pixels of the plain images have a strong correlation, which makes the leakage of the information possible. Thus, an effective image encryption algorithm should remove the correlation. 5000 pairs of pixels in horizontal, vertical and diagonal direction are randomly chosen from the plain, confused and cipher images and the coefficients are calculated as follows:

$$r_{x,y} = \frac{E((x - E(x))(y - E(y)))}{\sqrt{D(x)D(y)}}, \tag{21}$$

$$E(x) = \frac{1}{N} \sum_{i=1}^{N} x_i, \tag{22}$$

$$D(x) = \frac{1}{N} \sum_{i=1}^{N} (x_i - E(x))^2, \tag{23}$$

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

Figure 8 shows the plots of the horizontal, vertical and diagonal correlations between two adjacent pixels of the plain, confused and cipher image of Lena. And the correlation coefficients between two adjacent pixels in three directions for several test images with a size of 512×512 are shown in Table 3. From the figure and the table, it can be displayed that the coefficients of the original image are close to 1, those of the confused and cipher images have been almost cut down to 0.

5.5. Resistance to differential attacks

Number of pixels change rate (NPCR) and unified average changing intensity (UACI) are employed to assess the resisting differential attack performance of the proposed encryption algorithm. For two gray cipher images, the NPCR and UACI can be calculated from

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\% \,, \tag{24}$$

$$\text{UACI} = \frac{1}{W \times H} \left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255} \right] \times 100\%, \tag{25}$$

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

$$D(i,j) = \begin{cases} 1, & C_1(i,j) \neq C_2(i,j), \\ 0, & \text{otherwise}, \end{cases} \tag{26}$$

where W and H denote the width and height of the image, C_1 and C_2 represent the cipher images in which their plain images have only one pixel changed. The expected NPCR and UACI values for a 256-gray-scale image are 99.6094% and 33.4635%,[24] respectively.

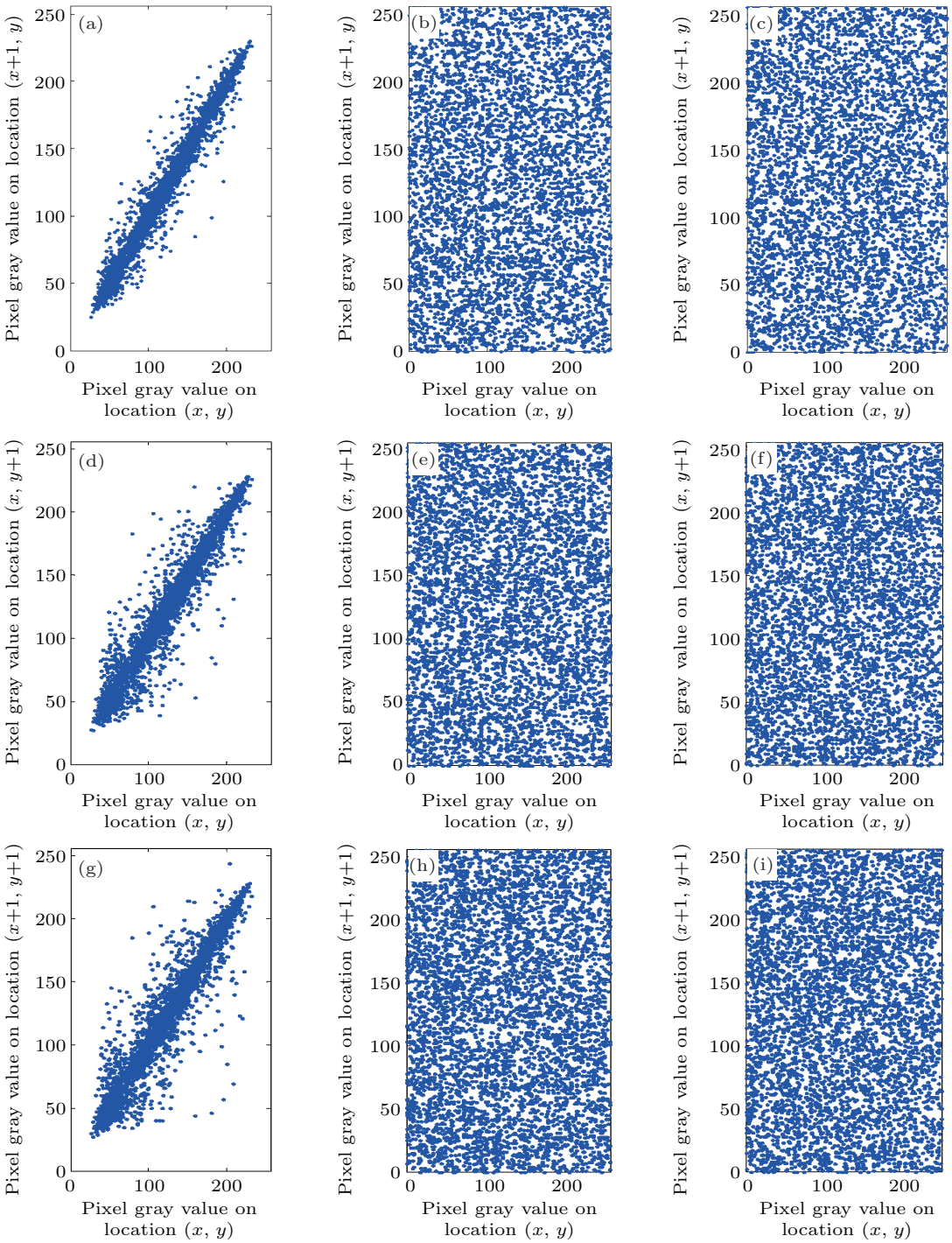


Fig. 8. (color online) Correlation of two adjacent pixels of horizontal direction in (a) plain image, corresponding (b) confused image, and (c) cipher image; vertical direction in (d) plain image, corresponding (e) confused image, and (f) cipher image; diagonal direction in (g) plain image, corresponding (h) confused image, and (i) cipher image.

Table 3. Correlation coefficients of two adjacent pixels in the plain, confused and cipher images.

Image	Direction	Lena	Peppers	Baboon	Barbara	Goldhill	Aerial
Plain image	Horizontal	0.9755	0.9813	0.7185	0.9672	0.9716	0.8413
	Vertical	0.9582	0.9690	0.8334	0.9332	0.9845	0.8791
	Diagonal	0.9383	0.9523	0.7006	0.9012	0.9646	0.7819
Confused image	Horizontal	−0.0193	0.0395	−0.0035	0.0291	0.0167	0.0797
	Vertical	0.0213	0.0209	0.0232	0.0151	0.0115	0.0347
	Diagonal	0.0290	0.0201	−0.0345	0.0160	0.0095	0.0301
Cipher image	Horizontal	0.0137	−0.0261	−0.0057	−0.0151	−0.0232	−0.0021
	Vertical	2.178×10^{-4}	0.0187	5.048×10^{-5}	-5.199×10^{-4}	−0.0251	−0.0061
	Diagonal	5.370×10^{-4}	−0.0025	0.0024	−0.0086	-3.537×10^{-4}	9.808×10^{-4}

We use Lena as the plain image, choose a location (i, j), change its pixel value and obtain another plain image. The two plain images are encrypted by our algorithm, the corresponding cipher images are utilized to compute NPCR and UACI, and the results are listed in Table 4. The average values of NPCR and UACI are 99.61% and 33.46%, respectively, which are close to their expected values.

Next, six test images are selected, and a pixel from the plain image is randomly chosen and the pixel value is added by 1, then a new plain image appears. The new cipher image and the old cipher image are employed to calculate the NPCR and UACI values. The 100 random pixels are selected for each image. The average, minimum, maximum NPCR and UACI values are shown in Table 5. From the results, it is evident that the proposed method is highly sensitive to tiny changes between two plain images. Two different cipher images are obtained, even if there is only one bit difference between the plain images.

Table 4. Lena images for different positions.

	Position			
	(1, 1)	(50, 60)	(200, 130)	(360, 400)
NPCR	99.60%	99.60%	99.61%	99.61%
UACI	33.47%	33.45%	33.46%	33.46%

Table 5. NPCRs and UACIs between ciphers while plain images only have one different pixel.

Image	Average		Minimum		Maximum	
	NPCR	UACI	NPCR	UACI	NPCR	UACI
Lena (512 × 512)	99.61%	33.46%	99.55%	33.39%	99.69%	33.54%
Peppers (512 × 512)	99.60%	33.45%	99.54%	33.33%	99.72%	33.57%
Baboon (512 × 512)	99.62%	33.44%	99.53%	33.35%	99.74%	33.58%
Barbara (512 × 512)	99.61%	33.46%	99.50%	33.31%	99.71%	33.56%
Goldhill (512 × 512)	99.62%	33.46%	99.56%	33.36%	99.74%	33.57%
Aerial (512 × 512)	99.61%	33.48%	99.51%	33.40%	99.75%	33.59%

5.6. Key sensitivity analysis

Key sensitivity of an image encryption algorithm can be assessed in two ways: (i) when the plain image is the same and the key has a slight change, a completely different cipher image should be obtained; (ii) the plain images cannot be recovered even though there is a slight difference between the encryption key and decryption key. Next, we will test the performance of our algorithm from these two aspects.

Firstly, Lena (Fig. 6(a)) is regarded as the plain image, and its cipher image is shown in Fig. 6(c). The SHA 256 hash value K is "b32171537de519d7c1c83ff25989a559-532c6b269a303 ac19d3d0a906bb5e250", then we modify K into K_1 = a32171537de519d7c1c83ff25989a559532c6b269a-303ac19 d3d0a906bb5e250" with a bit change, and the other parameters are the same. The cipher image is shown in Fig. 9(a), and figure 9(b) is the subtraction of Fig. 6(c) from Fig. 9(a), the NPCR is 99.61%, the histogram of Fig. 9(b) is illustrated in Fig. 9(c).

Secondly, we use a different decryption key to decrypt the cipher image shown in Fig. 6(c). When K is employed, the recovered image is shown in Fig. 10(a), and it is the same as the plain image. When we use K_1 to decrypt, we obtain the result presented in Fig. 10(b) and it is something like noise, we cannot obtain any information from it. The subtraction of Fig. 10(b) from Fig. 6(c) is illustrated in Fig. 10(c), and the NPCR is 99.60%.

From the above analyses, it is clear that the proposed algorithm is highly sensitive to the secret key and the plain image, and it is secure to withstand statistical attack and brute force attack.

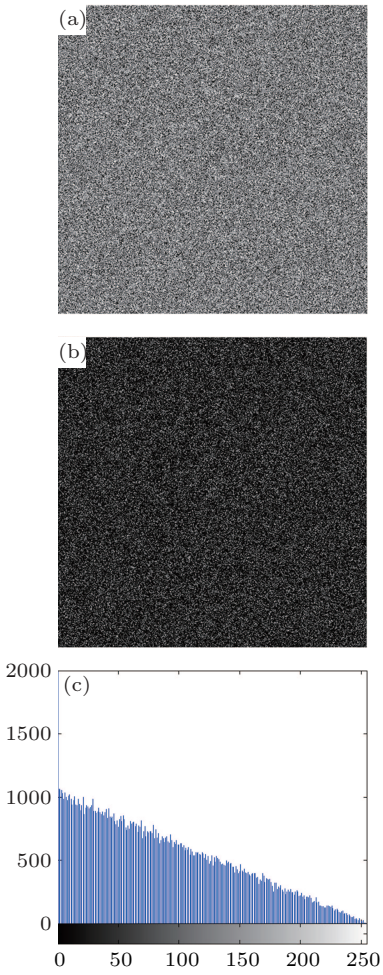


Fig. 9. (color online) Encryption results with the key changed, showing (a) cipher image with K_1, (b) difference between two cipher images, and (c) the histogram of Fig. 9(b).

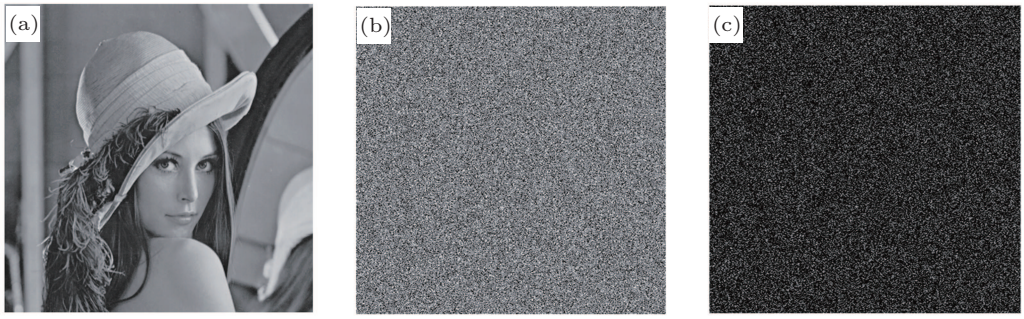


Fig. 10. Decryption results with different keys, showing (a) recovered image with K, (b) recovered image with K_1, and (c) subtraction of Fig. 10(b) from Fig. 6(c).

5.7. Resisting occlusion attack analysis

Occlusion attack means that the cipher image is damaged in the transmission process, and the more data it loses, the more distortion the recovered image has. The peak signal-to-noise ratio (PSNR) is employed to calculate the quality of the recovered image after being attacked. For a grayscale image, the PSNR may be expressed as

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

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

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

The results are shown in Fig. 11 and Table 6. Figures 11(a)–11(d) give the cipher images of Fig. 6(c) with 1/16, 1/8, 1/4, and 1/2 occlusion respectively, and figures 11(e)–11(h) show the corresponding recovered images whose PSNRs are 37.9059, 34.8570, 31.8821, and 30.5522, and there are 11.45%, 22.56%, 45.12%, and 49.81% difference between the recovered images and the plain image (shown in Fig. 6(a)). In Fig. 11(h), it is evident that the recovered image can be easily recognized though the cipher image lost 50% information. Therefore, the proposed encryption algorithm has a superb robustness to the occlusion attack and is comparable to that in Ref. [51].

5.8. Resisting noise attack analysis

In the image transmission, channel noise is inevitable. Gaussian white noises with zero mean and three different variances are added to the cipher image (Fig. 6(c)), the respective images are illustrated in Figs. 12(a)–12(c), the recovered images are shown in Figs. 12(d)–12(f), with the PSNRs of 28.5242, 28.2274, and 28.1247, and other quantitative results of resisting noise attack are given in Table 7. With the increase of the variance of noise, the MSE values increase, PSNR values decrease, and the correlations of adjacent pixels in the recovered images are very low. The NPCR and UACI values are less than those in Ref. [52], which means that our algorithm is better. When the variance of Gaussian white noise is 0.0005, the recovered image (Fig. 12(f)) has the worst image quality, and it can be roughly recognized visually, which means that our algorithm can withstand the noise attack to a certain extent.

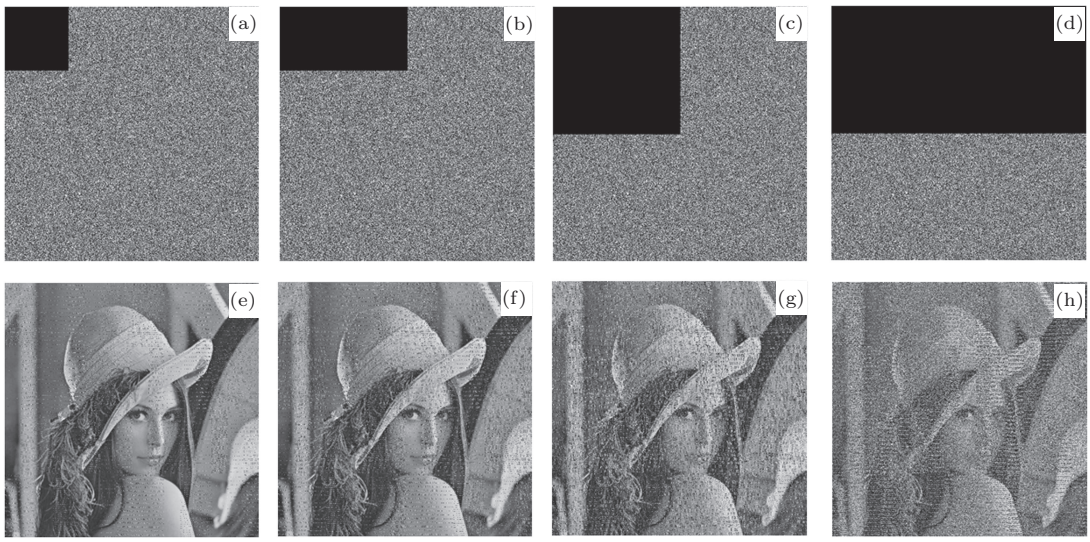


Fig. 11. (color online) Occlusion attack analysis results of the cipher images with (a) 1/16, (b) 1/8, (c) 1/4, and (d) 1/2 occlusion, and recovered images with (e) 1/16, (f) 1/8, (g) 1/4, and (h) 1/2 occlusion.

Table 6. Quantitative results of resisting occlusion attack.

Occlusion	Ours					Ref. [51]
	MSE	PSNR	Correlation	NPCR	UACI	PSNR
1/16	10.5316	37.9059	0.6132	0.1145	0.0232	39.23
1/8	21.2510	34.8570	0.4602	0.2256	0.0461	–
1/4	40.1568	32.8821	0.3151	0.4512	0.0922	33.37
1/2	57.2614	30.5522	0.0635	0.4981	0.1432	30.40

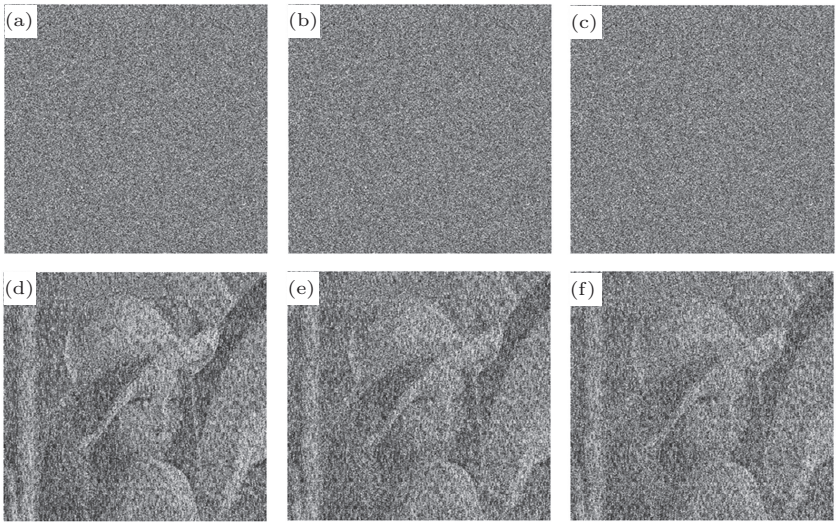


Fig. 12. Cipher and decrypted images under Gaussian noise with different level of noise: cipher images of (a) mean = 0, variance = 0.0001; (b) mean = 0, variance = 0.0003; (c) mean = 0, variance = 0.0005; decrypted images of (d) mean = 0, variance = 0.0001; (e) mean = 0, variance = 0.0003; (f) mean = 0, variance = 0.0005.

Table 7. Quantitative results of resisting noise attack.

Variance	Ours					Ref. [52]	
	MSE	PSNR	correlation	NPCR	UACI	NPCR	UACI
0.0001	91.3405	28.5242	0.1821	0.9269	0.2050	0.9920	0.2844
0.0003	97.8001	28.2274	0.1220	0.9561	0.2247	0.9961	0.2864
0.0005	100.1408	28.1247	0.1056	0.9656	0.2325	0.9974	0.2880

5.9. Resisting known-plaintext and chosen-plaintext attack analysis

Many encryption schemes are broken by known-plaintext and chosen-plaintext attacks. In the present paper, two methods are manipulated to upgrade the ability to resist known-plaintext and chosen-plaintext attacks. On the one hand, the chaotic sequences used in confusion and diffusion processes are generated from the four-order memristive chaotic system and LTS, the initial values and parameters of the chaotic systems are calculated by SHA 256 hash value of the plain image, and thus our algorithm depends on the original image. On the other hand, 24 different position sequence groups are obtained in BCBPSG, a certain group is picked out to confuse the sub blocks, and its choosing method has high relationship with the plain image. Therefore, in theory, our encryption algorithm has strong ability to withstand the known-plaintext and chosen-plaintext attacks.

The cryptanalysts often choose special images, such as all black and all white to attack the cryptosystem. They want to obtain the key by analyzing the cipher images. Figure 13 shows the encryption results of all white and black images by using our method. The entropies and correlation coefficients of the plain, cipher images of all white and all black are listed in Table 8. From the figure and the table, we can find that the histograms of the cipher images distribute evenly and uniformly, the correaltions of adjacent pixels decrease to nearly 0, and any infromation cannot be obtained from the ciphers. So our algorithm can resist the known-plaintext and chosen-plaintext attacks.

5.10. Speed analysis

Speed is an important issue for a cryptosystem. The proposed encryption scheme consists of block confusion based on 3D Brownian motion (BCB3DBM), block confusion based on position sequence group (BCBPSG) and diffusion. Before the confusion process, the original image is transformed into a 3D cubic matrix, the aim is to reduce the iterative times of memristive chaotic systems and improve the encryption speed. In addition, the cubic matrix is divided into many sub blocks. The permutation process is manipulated by blocks, and it can run in parallel, thereby saving time. The size of the sub block

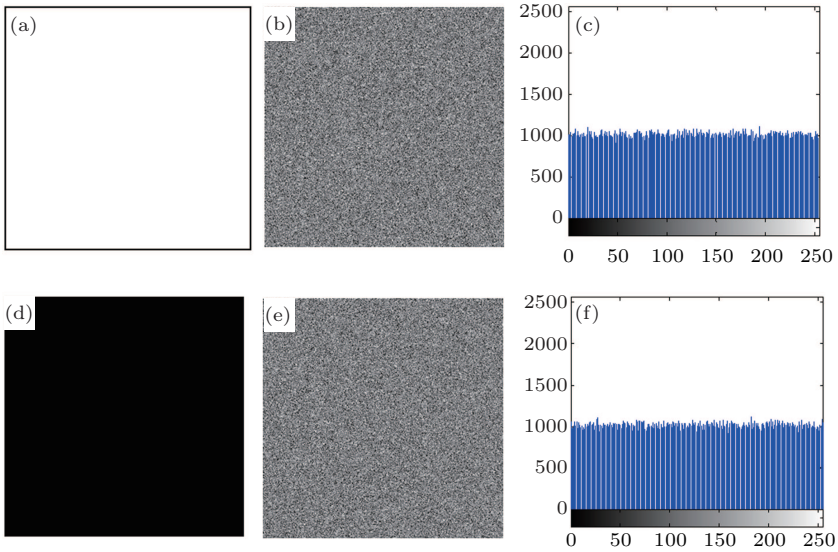


Fig. 13. (color online) Experimental results of (a) all white, (b) cipher image of all white, and (c) histogram of the cipher image; (d) all black, (e) cipher image of all black, and (f) histogram of the cipher image.

image and the test image can affect the encryption speed. For a 512×512 gray image, the encryption times for $4 \times 4 \times 4$, $8 \times 8 \times 8$, $16 \times 16 \times 16$, $32 \times 32 \times 32$, $64 \times 64 \times 64$ blocks are tested, and the relationship between execution time and the size m of the sub block image is shown in Fig. 14. It is evident that the encryption time is the shortest when $m = 8$, and it is 5.69 s. The proposed algorithm has shorter encryption time than Wang's[50] algorithm (15.14 s) and Zhou's[32] methods (11.42 s). Therefore, our encryption algorithm is more efficient and suitable for the secure communication.

Table 8. Entropies and correlation coefficients of the plain, cipher images of all white and all black.

Images	Entropies	Correlation coefficients		
		Horizontal	Vertical	Diagonal
All white	0	–	–	–
Cipher of white	7.9993	0.0140	0.0011	0.0071
All black	0	–	–	–
Cipher of black	7.9992	0.0018	0.0154	–0.0242

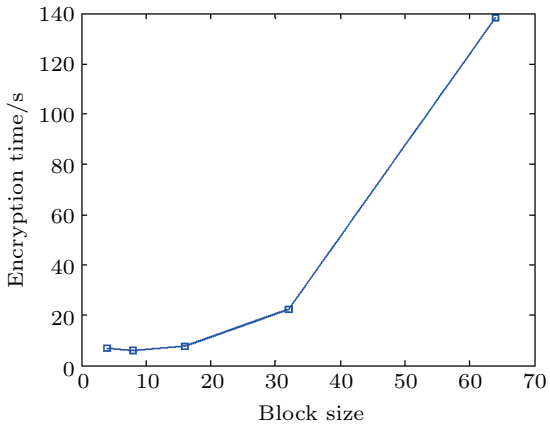


Fig. 14. (color online) Relationship between block size and encryption time.

6. Conclusions

A novel chaos-based image encryption algorithm using the three-dimensional (3D) Brownian motion is given in this paper. We present the block confusion based on 3D Brownian motion (BCB3DBM) and block confusion based on position sequence group (BCBPSG). The BCB3DBM is used to confuse the position of the bits within the plain-image sub blocks, and the subsequent BCBPSG is employed to permutate the positions of the sub blocks. In theory, the permutation process can make the bits in one bit plane appear at any position and in any bit plane, and our confusion scheme has a good scrambling result. Moreover, we also utilize confusion to enhance the encryption effect. In the encryption scheme, the four-order memristive chaotic system and LTS are used to generate the chaotic sequences. Besides, SHA 256 hash function of the plain image is employed to produce the initial values and parameters of the chaotic systems, and therefore our encryption scheme has strong sensitivity to the plain image.

In the security analyses, we employ 10 different methods, i.e., key space analysis, histogram analysis, information entropy analysis, correlations between two adjacent pixels, and resistance to differential attacks, key sensitivity analysis, resisting occlusion attack analysis, resisting noise attack analysis, resisting known-plaintext and chosen-plaintext attack analysis and speed analysis to assess the performance of this algorithm. The experiment results show that our method is secure and effective. It has potential applications in multimedia communication.

References

[1] Zhang X Y, Zhang G J, Li Xuan, Ren Y Z and Wu J H 2016 *Chin. Phys. B* **25** 054201

[2] Luo Y L, Cao L C, Qiu S H, Lin H, Harkin Jim and Liu J X 2016 *Nonlinear Dyn.* **83** 2293

[3] Wang Z, Huang X, Li Y X and Song X N 2013 *Chin. Phys. B* **22** 010504
[4] Matthews R 1989 *Cryptologia* **4** 29
[5] Wang X Y and Wang Q 2014 *Chin. Phys. B* **23** 030503
[6] Zhou Y C, Hua Z Y, Pun C M and Philip Chen C L 2015 *IEEE T. Cybernetics* **45** 2001
[7] Ye G D 2014 *Nonlinear Dyn.* **75** 417
[8] Chai X L, Gan Z H, Chen Y R and Zhang Y S 2017 *Signal Process.* **134** 35
[9] Zhang Y Q and Wang X Y 2014 *Inf. Sci.* **273** 329
[10] Tong X J, Wang Z, Zhang M, Liu Y, Xu H and Ma J 2015 *Nonlinear Dyn.* **80** 1493
[11] Yen J C and Guo J I 2000 *IEEE Proc. Vis. Image Signal Process.* **147** 167
[12] Li C Q 2016 *Signal Process.* **118** 203
[13] Li H, Wang Y, Yan H, Li L, Li Q and Zhao Z 2013 *Opt. Lasers Eng.* **51** 1327
[14] Chen J X, Zhu Z L, Fu C, Zhang L B and Yu H 2015 *Opt. Lasers Eng.* **66** 1
[15] Liu Y S, Fan H, Xie Y Eric, Cheng G and Li C Q 2015 *Int. J. Bifur. Chaos* **25** 1550188
[16] Sam I S, Devaraj P and Bhuvaneswaran R S 2012 *Multimed Tools Appl.* **56** 315
[17] Zhang G and Liu Q 2011 *Opt. Commun.* **284** 2775
[18] Wang X and He G 2011 *Opt. Commun.* **284** 5804
[19] Eslami Z and Bakhshandeh A 2013 *Opt. Commun.* **286** 51
[20] Akhavan A, Samsudin A and Akhshani A 2015 *Opt. Commun.* **350** 77
[21] Wang X Y and Liu L T 2013 *Nonlinear Dyn.* **73** 795
[22] Mirzaei O, Yaghoobi M and Irani H 2012 *Nonlinear Dyn.* **67** 557
[23] Li C Q, Liu Y S, Xie T and Chen Michael Z Q 2013 *Nonlinear Dyn.* **73** 2083
[24] Zhu C 2012 *Opt. Commun.* **285** 29
[25] Xie Eric Y, Li C Q, Yu S M and Lü J H 2017 *Signal Process.* **132** 150
[26] Shannon C E 1949 *Bell Syst. Tech. J.* **28** 656
[27] Wang X Y, Liu C M, Xu D H and Liu C X 2016 *Nonlinear Dyn.* **84** 1417
[28] Ye G D and Huang X L 2016 *Secur. Commun. Netw.* **9** 2015
[29] Yao W, Zhang X, Zheng Z M and Qiu W J 2015 *Nonlinear Dyn.* **81** 151
[30] Xu L, Li Z, Li J and Hua W 2016 *Opt. Lasers Eng.* **78** 17
[31] Liu H J and Wang X Y 2011 *Opt. Commun.* **284** 3895
[32] Zhou Y C, Cao W J and Philip Chen C L 2014 *Signal Process.* **100** 197
[33] Safwan EI Assad and Mousa Farajallah 2016 *Signal Process. Image Commun.* **41** 144
[34] Zhu Z L, Zhang W, Wong K W and Yu H 2011 *Inform. Sci.* **181** 1171
[35] Liu H J and Wang X Y 2011 *Opt. Commun.* **284** 3895
[36] Martin Del Rey A and Rodriguez Sanchez G 2015 *Int. J. Mod. Phys. C* **26** 1450069
[37] Hua Z Y and Zhou Y C 2016 *Inform. Sci.* **339** 237
[38] Zhang Y S and Xiao D 2014 *Commun. Nonlinear Sci. Numer. Simul.* **19** 74
[39] Zhang W, Wong K, Yu H and Zhu Z L 2013 *Commun. Nonlinear Sci. Numer. Simul.* **18** 584
[40] Fu C, Meng W H, Zhan Y F, Zhu Z L, Francis C M Lau, Chi K Tse and Ma H F 2013 *Comput. Biol. Med.* **43** 1000
[41] Wang X Y and Zhang H L 2015 *Opt. Commun.* **342** 51
[42] Chua L O 1971 *IEEE Trans. Circuit Theory* **18** 507
[43] Duan S K, Zhang Y, Hu X, Wang L D and Li C D 2014 *Neural Comput. Appl.* **25** 1437
[44] Adhikari S P, Yang C, Kim H and Chua L O 2012 *IEEE Trans. Neural Netw. Learn. Syst.* **23** 1426
[45] Theesar S J S and Balasubramaniam P 2014 *Circuits Syst. Signal Process.* **33** 37
[46] Li Y X, Huang X and Song Y W 2015 *Int. J. Bifur. Chaos* **25** 1550151
[47] Zhou Y C, Bao L and Philip Chen C L 2014 *Signal Process.* **97** 172
[48] Àlvarez G and Li S 2006 *Int. J. Bifur. Chaos* **16** 2129
[49] Mirzaei O, Yaghoobi M and Irani H 2012 *Nonlinear Dyn.* **67** 557
[50] Wang X Y and Xu D H 2014 *Nonlinear Dyn.* **75** 345
[51] Hsiao Hung-I and Lee Junghsi 2015 *Signal Process.* **117** 281
[52] Liu H, Wang X and Kadir A 2012 *Appl. Soft Comput.* **12** 1457