Multimed Tools Appl (2017) 76:1159–1175
DOI 10.1007/s11042-015-3088-1

An image encryption algorithm based on bit level Brownian motion and new chaotic systems

Xiuli Chai[1]

Received: 28 May 2015 / Revised: 19 October 2015 / Accepted: 17 November 2015 /
Published online: 2 December 2015

Abstract In this paper, a new image encryption algorithm based on Brownian motion and new 1D chaotic system is introduced. Firstly, SHA 256 hash value of the plain image is used to generate the initial values and system parameters of chaotic systems for confusion and diffusion process. Then, 8 bitplanes of the plain image are scrambled based on Brownian motion, respectively, and the position and value of all pixels are changed simultaneously. After the confusion process, a two directional diffusion process is carried out, and it is made up of row diffusion (RD) and column diffusion (CD). The whole process can be repeated many rounds in order to get better encryption effect. Simulation results and security analysis show that our scheme has properties of large key space, high sensitivity to key, strong resisting statistical and differential attack. So, it has high security and important practical application in image transmission and image encryption.

Keywords Brownian motion · Directional diffusion · Row diffusion (RD) · Column diffusion (CD) · High security

1 Introduction

With the development of multimedia technology and internet, the security of images and videos has drawn many researchers' attentions [25]. Until now, a variety of encryption algorithms such as advanced encryption standard (AES) and data encryption standard (DES) have been proposed. But these algorithms are not efficient for encrypting images due to image inherent features such as bulky data, high correlation among pixels and so on.

Chaotic systems have many important characteristics, such as ergodicity and sensitivity to initial condition and control parameters, and these features can be considered analogous to the

✉ Xiuli Chai
chaixiuli@henu.edu.cn

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

desired cryptographic properties [23]. The first chaos-based encryption scheme was proposed in 1989 [15], and then many chaotic image encryption algorithms have been developed, and chaos-based image encryption has become one of efficient and excellent encryption methods [3, 21, 24].

Image encryption methods mostly use two kinds of chaotic systems, one-dimensional (1D) chaotic system and high-dimensional (HD) chaotic system [8]. The HD chaotic systems have complex structures and multiple parameters, these add to the security of encryption systems, but increase the difficulty of hardware implementations and computation complexity [22]. In contrast, 1D chaotic systems mostly contain one variable and a few parameters (Such as Logistic, Tent, Gaussian maps), have simple structures and are easy to implement, but their chaotic orbits are rather simple and can be predicted with the development of signal processing technology [11]. Hence, designing and utilizing new 1D chaotic system in encrypting images is important for the large scale practical application of encryption systems. Fortunately, some new systems with better chaotic performance have been developed and can be used in image encryption [26, 28, 29].

Image encryption is mostly divided into two stages: permutation (or confusion) and diffusion [12]. In the general permutation stage, the image pixels are permuted using some transformation method, such as baker map, Arnold map, magic square, while the pixel values remain unchanged, the histograms of the cipher image and the plain image are the same [2]. Therefore, the confusion is not resistant to the statistical analysis [6]. In the diffusion stage, the pixel values are changed, and diffusion may lead to higher security [27]. Recently, an interesting technique based on image bitplane decomposition and encryption at bit level has presented excellent encryption performance and is easily implemented in hardware [1]. Firstly, the technique decomposes an image into several binary bitplanes using traditional binary decomposition, gray code decomposition, Fibonacci p-code decomposition. Then, the pixels at bit level are manipulated using various methods, and finally bitplanes are combined into the cipher images. Some image encryption schemes have been introduced based on this technique [14, 17, 30]. However, these algorithms have a low secure level because of their predictable decomposition results and/or smaller key spaces [31]. So, new encryption methods with higher security need to be studied.

Brownian motion is the random motion of particles suspended in a fluid (a liquid or a gas) resulting from their collision with the quick atoms or molecules in the gas or liquid. The term “Brownian motion” can also refer to the mathematical model used to describe such random movements, which is often called a particle theory. At present, Brownian motion is used to analyze the financial share price in the market. In 2014, Wang and Xu [19] took each pixel of the image as a Brownian particle, used the Monte Carlo method to simulate the Brownian motion, and effectively scrambled the image. In 2015, Zhu [32] had broken the encryption algorithm of Wang et al. [19] because the permutation vector and diffusion sequence of Wang et al. [19] are not related with the plaintext image, and this make the method unfeasible to resist the chosen plaintext attack.

In this paper, we introduce an image encryption algorithm based on bit level Brownian motion and new 1D chaotic systems. Firstly, we decompose the image into 8 bitplanes, then takes the pixels at bit level as Brownian particles, uses the Monte Carlo method to simulate the Brownian motion, scrambles the image pixels, and then combines the bitplanes to get the confused image. In the algorithm, we use three new 1D chaotic systems, they have wide range of parameter settings and the uniform-distributed variant density function, and this can increase the key space and enhance the encryption security. And confusion step is manipulated at the bit level, eight bitplanes of the image are scrambled based on different Brownian motions, respectively, and this increases the ability of resisting the statistical analysis and chosen plaintext attack. In order to achieve high security, a two directional diffusion step is followed.

The rest of the paper is organized as follows: In Section 2, a brief description of the chaotic system and Brownian motion simulation is provided. Section 3 provides the encryption algorithm. Simulation results and security analyses are given in Sections 4 and 5. Section 6 reaches a conclusion.

2 Chaotic systems and Brownian motion simulation

2.1 Chaotic systems

Logistic map, Sine map and Tent map, illustrated as the follows, which are simple and classic dynamical equation with complex chaotic behaviors, are used in many image encryption algorithms in the past few years.

$$X_{n+1} = rX_n(1-X_n) \tag{1}$$

$$X_{n+1} = \begin{cases} rX_n/2, X_n < 0.5 \\ r(1-X_n)/2, X_n \geq 0.5 \end{cases} \tag{2}$$

$$X_{n+1} = r\sin(\pi X_n)/4 \tag{3}$$

Where r is the system parameter.

Unfortunately, these systems have the shortcomings of limited chaotic range and uneven distribution of sequences. To solving these defects, new chaotic systems are proposed to get better chaotic behaviors which are defined as following:

2.1.1 The logistic-sine system (LSS)

Combined Logistic map and Sine map, Logistic-Sine System is presented in the following equation:

$$u_{n+1} = LSS(r, u_n) = (ru_n(1-u_n) + (4-r)\sin(\pi u_n)/4)\mathrm{mod}1 \tag{4}$$

Where parameter $r \in (0,4]$.

2.1.2 The logistic-tent system (LTS)

Combined Logistic map and Tent map, Logistic-Tent System is illustrated in the following equation:

$$v_{n+1} = LTS(r, v_n) = \begin{cases} (rv_n(1-v_n) + (4-r)v_n/2)\mathrm{mod}1, v_n < 0.5 \\ (rv_n(1-v_n) + (4-r)(1-v_n)/2)\mathrm{mod}1, v_n \geq 0.5 \end{cases} \tag{5}$$

Where parameter $r \in (0,4]$.

2.1.3 The tent-sine system (TSS)

Combined Tent map and Sine map, Tent-Sine System is shown in the following equation:

$$s_{n+1} = TSS(r, s_n) = \begin{cases} (rs_n/2 + (4-r)\sin(\pi s_n)/4)\mathrm{mod}1, s_n < 0.5 \\ (r(1-s_n)/2 + (4-r)\sin(\pi s_n)/4)\mathrm{mod}1, s_n \geq 0.5 \end{cases} \tag{6}$$

Where parameter $r \in (0,4]$.

The bifurcation diagrams of Logistic map, Sine map, Tent map, Logistic-Sine System (LSS)、Logistic-Tent System (LTS) and Tent-Sine System (TSS) are illustrated in Fig. 1. Compared with its corresponding seed maps, Logistic-Sine System (LSS)、Logistic-Tent System (LTS) and Tent-Sine System (TSS) have a wider chaotic range and chaotic behaviors, and they are more suitable for image encryption [29].

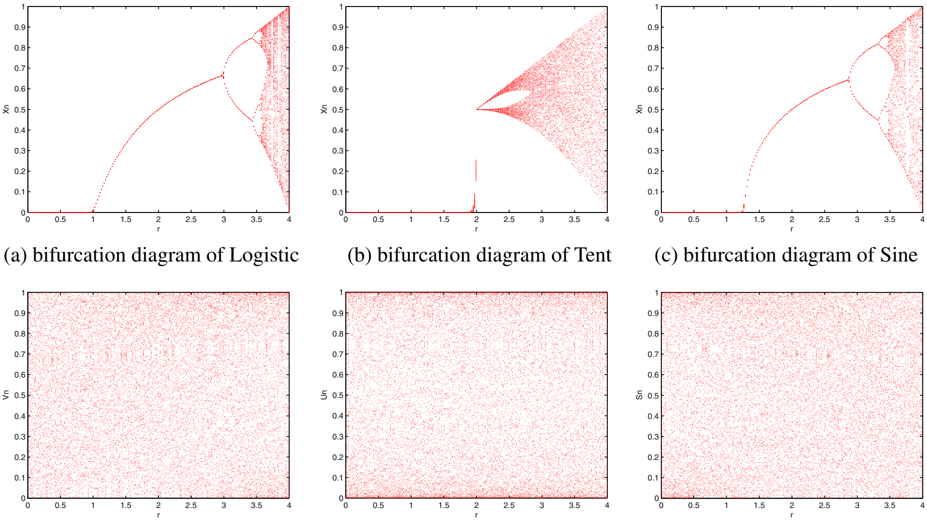
2.2 Brownian motion

The current moving distance of any one point in space can be expressed in 2D plane,

$$X = r\,\sin a\,\cos b, Y = r\,\sin a\,\sin b \tag{7}$$

$$a = u_i \times 2 \times \pi, b = v_i \times \pi \tag{8}$$

Where, X and Y is the horizontal and vertical direction distance, respectively; r is the step length of the movement, $0 \leq r \leq +\infty$; and a and b denote the movement direction of the particle, $0 \leq b \leq 2\pi$, $0 \leq a \leq \pi$; u_i, v_i is the chaotic sequences generated by chaotic systems. We simulated the Brownian motion of 300 particles, the movement trace of a particle is shown in Fig. 2, and the figures indicate that the motion has strong randomness at different times.



(a) bifurcation diagram of Logistic (b) bifurcation diagram of Tent (c) bifurcation diagram of Sine

(d) bifurcation diagram of LTS (e) bifurcation diagram of LSS (f) bifurcation diagram of TSS

Fig. 1 Bifurcation diagram of Logistic, Tent, Sine, LTS, LSS and TSS

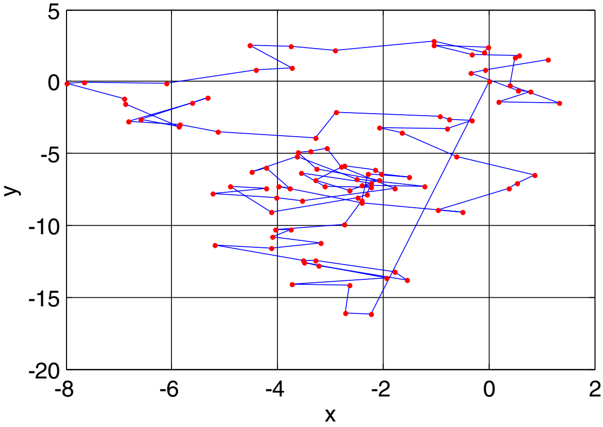


Fig. 2 Single Brownian particle motion path

3 The encryption algorithm

The novel image encryption algorithm is presented in Fig. 3. The algorithm first decomposes the plain image into 8 bit planes. Then confusion at bit level based on Brownian motion changes all bit locations, and after $k1$ iterations, the 8 bit planes can be combined. After $k2$ iterations of a two directional diffusion, the encrypted image appears. The confusion and diffusion can be manipulated $k3$ rounds in order to get a required security level. The parameters (encryption keys) used in confusion and diffusion are generated through the plain image, and we can get "one plain image, one key", so the security level of our algorithms increased a lot. The detailed encryption algorithm is presented as follows.

3.1 Confusion

First, we decompose the plain image into 8 binary bit level images, and then scramble the 8 images using Brownian motion. Confusion at bit level not only changes the position of each pixel, but also changes its values. Besides, the permutation vector depends on the plain image, and the ability of resisting the chosen plaintext attack increases.

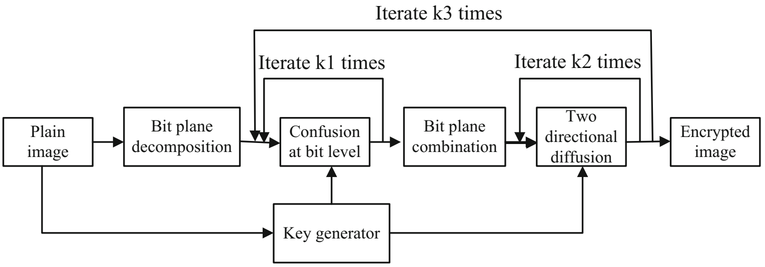


Fig. 3 the flow chart of encryption schemes

For a $M \times N$ plain image P, the Logistic-Sine System (LSS) and Logistic-Tent System (LTS) are both used in the confusion. The confusion process can be described as follows.

Step 1 the SHA256 hash function is used to generate the 256-bit key stream based on the plain image, the key stream is represented in hexadecimal number.

Step 2 Dividing the 256-bit key stream into 32 groups denoted by k_l, $K=\{k_l\}$,$(l=1,2,\cdots 32)$, then the sum of the 32 groups is computed using the equation $sum = \sum_{l=1}^{32} k_l$, denoted by sum.

Step 3 the 8 system parameters and 16 initial values of the Logistic-Sine System and Logistic-Tent System are computed according to the following formula:

$$r_{i0} = \mathrm{mod}\left((k_{3i-2} \oplus k_{3i-1} + k_{3i} + sum)/2^8, 1\right)/5 + 3, i = 1, 2, \cdots, 8 \tag{9}$$

$$u_{i0} = v_{i0} = \mathrm{mod}\left((k_{38-6i} \oplus k_{36-6i} + k_{34-6i} + sum)/2^8, 1\right), i = 1, 2, 3, 4, 5 \tag{10}$$

$$u_{i0} = v_{i0} = \mathrm{mod}\left((k_{6i-35} \oplus k_{6i-33} + k_{6i-31} + sum)/2^8, 1\right), i = 6, 7, 8 \tag{11}$$

Where, r_{10}, r_{20},…, r_{80} present the system parameters of LSS and LTS used for the bit plane 1, 2, …, 8; u_{10}, u_{20},…, u_{80} denote the initial values of LSS used for the bit plane 1, 2, …, 8; v_{10}, v_{20},…, v_{80} are the initial values of LTS used for the bit plane 1, 2, …, 8, respectively. For the sake of computation, the system parameters of LSS and LTS used for each plane are equal, and the initial values of LSS and LTS for every plane are the same.

Step 4 the plain image is decomposed into 8 bit planes, every bit plane is converted to one-dimensional sequence (from left to right, up to down). For each bit plane, use the corresponding initial value and system parameters and the LSS and LTS to generate two chaotic sequences u_i, v_i ($i=1$, 2, …, MN).

Step 5 Taking the bit pixel in each bit plane as Brownian motion particle, set the step length r of the movement in every bit plane, use Eqs. (7) and (8) to get the current horizontal and vertical direction distance, iterate the movement for R rounds, record the positions of all particles, establish a one-to-one map between the bit plane sequence and the positions, then the confused bit plane image can be gotten. After the 8 bit planes have been scrambled, the bit planes are combined to a confused image. The bit plane confusion can be iterated $k1$ times in order to get better encryption effect.

3.2 Diffusion

The flow chart of diffusion process is shown in Fig. 4. The diffusion step is mainly made up of row diffusion (RD) and column diffusion (CD). The RD consists of random pixel insertion, row separation, forward row mapping (FRM), reverse row mapping (RRM) and row combination. The CD consists of random pixel insertion, column separation, downward column mapping (DCM), reverse column mapping (RCM) and column combination. Firstly, set the diffusion encryption rounds as $k2$. When $k2\%2=0$, the confused image is manipulated by column diffusion (CD). When $k2\%2\neq0$, the confused image is processed

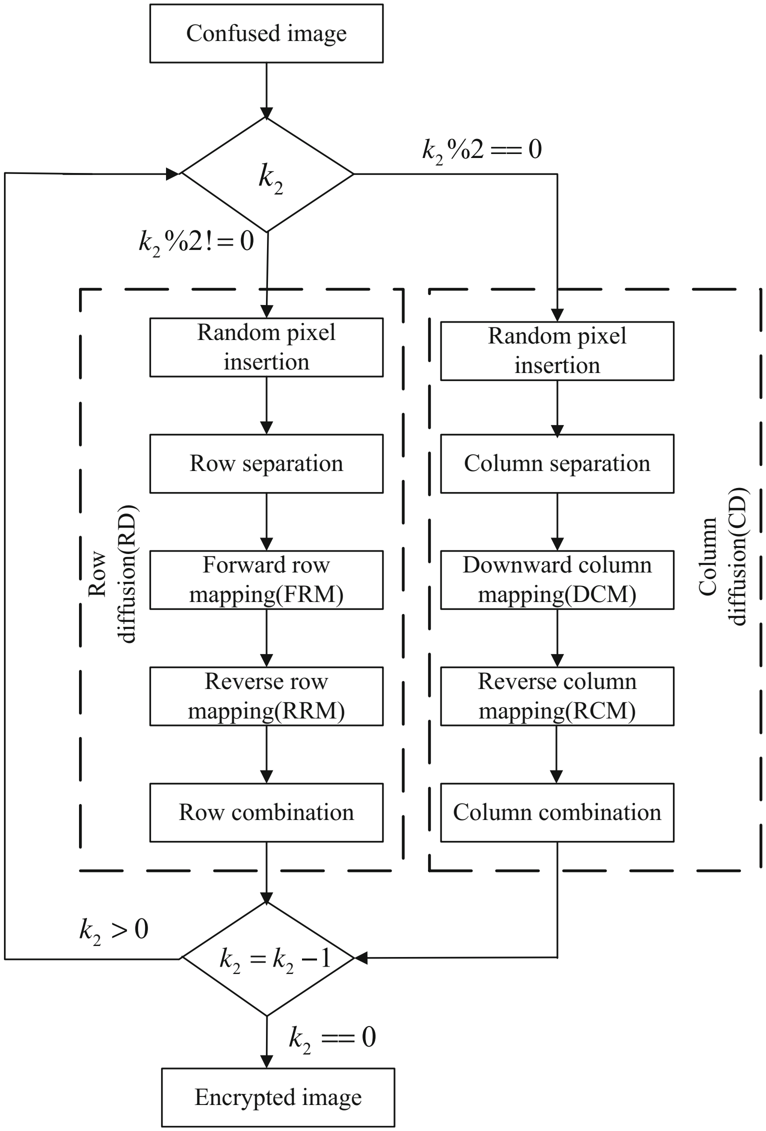


Fig. 4 Flow chart of diffusion step

by row diffusion (RD). Then $k2$=$k2$-1, the diffusion process continues until $k2$=0, the diffusion process is completed.

3.2.1 Row diffusion (RD)

The detailed steps of RD are as follows:

Step 1 Random pixel insertion

Insert two random numbers at the beginning and end of every row in the confused image, respectively, $2M$ random numbers are needed, then the image is changed to

M×(2+N). Random numbers can be produced by any random generators, and that is unpredictable for each encryption round.

Step 2 Row separation

Separate every row in the image and transfer the image row by row into 1D matrices.

Step 3 Forward row mapping (FRM)

FRM process is used to change the pixel values in every 1D matrix. It is defined by the following equations:

$$c_{i,j} = \begin{cases} p_{i,j}, j = 1 \\ c_{i,j-1} \oplus p_{i,j} \oplus s(i,j), N+1 \geq j \geq 2 \\ p_{i,j}, j = N+2 \end{cases} \tag{12}$$

$$s(i,j) = \mathrm{TSS}(r, s(i,j-1)) \tag{13}$$

Where $p_{i,j}$ denotes jth pixel of the ith row of the plain image, $c_{i,j}$ is the jth pixel of the ith row of the cipher image, and $s(i,j)$ is the chaotic sequences produced by TSS, i=1, 2, …, M; j=1, 2, …, N+2, the parameter r_{r0} and the initial value s_{r0} can be gotten by the following equations,

$$r_{r0} = \mathrm{mod}\left((k_{25} \oplus k_{26} + k_{27} + sum)/2^8, 1\right)/5 + 3 \tag{14}$$

$$s_{r0} = \mathrm{mod}\left((k_{19} \oplus k_{21} + k_{23} + sum)/2^8, 1\right) \tag{15}$$

Step 4 Reverse row mapping (RRM)

In order to increase the encryption effect, RRM process is taken. RRM is illustrated as the following equations:

$$c'_{i,j} = \begin{cases} c_{i,j}, j = N+2 \\ c'_{i,j+1} \oplus c_{i,j} \oplus s(i,j), N+1 \geq j \geq 2 \\ c_{i,j}, j = 1 \end{cases} \tag{16}$$

Where $c_{i,j}$ denotes the jth pixel of the ith row of the cipher image after FRM, $c'_{i,j}$ is the jth pixel of the ith row of the cipher image after RRM, and $s(i,j)$ is the same chaotic sequences produced by TSS as that of FRM process, i=1, 2, …, M, j=1, 2, …, N+2.

Step 5 Row combination

After all the pixel values have been changed, the first and last pixel in each row is removed, and then we combine all 1D matrices back into a 2D image. The RD process is finished.

3.2.2 Column diffusion (CD)

The steps of column diffusion (CD) are as follows:

Step 1 Insert two random numbers at the beginning and end of every column in the confused image, respectively, 2 N random numbers are needed, then the image is changed to (M + 2) × N. Afterwards, separate every column in the image and transfer the image column by column into 1D matrices.

Step 2 Downward column mapping (DCM) process and reverse column mapping (RCM) process are successively operated as the following equations, respectively,

$$c_{i,j} = \begin{cases} p_{i,j}, i = 1 \\ c_{i-1,j} \oplus p_{i,j} \oplus s(i,j), m + 1 \geq i \geq 2 \\ p_{i,j}, i = m + 2 \end{cases} \tag{17}$$

$$c'_{i,j} = \begin{cases} c_{i,j}, i = m + 2 \\ c'_{i+1,j} \oplus c_{i,j} \oplus s(i,j), m + 1 \geq i \geq 2 \\ c_{i,j}, i = 1 \end{cases} \tag{18}$$

$$s(i,j) = \mathrm{TSS}(r, s(i-1,j)) \tag{19}$$

Where $p_{i,j}$ denotes the ith pixel of the jth column of the plain image, $c_{i,j}$ is the ith pixel of the jth column of the cipher image after DCM, $c'_{i,j}$ is the ith pixel of the jth column of the cipher image after RCM, and $s(i,j)$ is the chaotic sequences produced by TSS, i=1, 2, …, M+2; j=1, 2, …, N, the parameter r_{l0} and the initial value s_{l0} can be attained by the following equations,

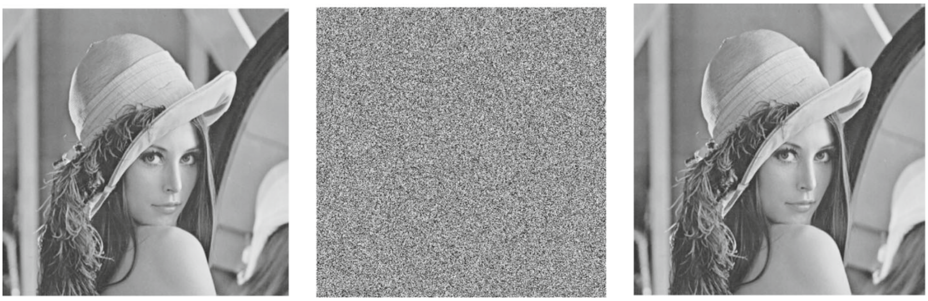
$$r_{l0} = \mathrm{mod}\left((k_{28} \oplus k_{29} + k_{30} + sum)/2^8, 1\right)/5 + 3 \tag{20}$$

$$s_{l0} = \mathrm{mod}\left((k_{25} \oplus k_{27} + k_{29} + sum)/2^8, 1\right) \tag{21}$$

Step 3 After changing the entire pixel values, remove the first and last pixel in each column, and combine all 1D matrices back into a 2D image. The CD process is ended.

4 Simulation results

We use Matlab 2014a to run the encryption and decryption process in computer with 3 GHz CPU, 4 GB memory and Windows 7 operating system. We choose the Lena.bmp (512×512) as the plain image. The key is set as: SHA256('Lena.bmp') = 2D74A6D08182A5B41E4ADD30C9A6CEEF4AE5822D7462224996D71877E8314F1-A, The rounds of Brownian motion R=100, the step length of the movement in every bit plane is 2, the rounds of confusion k1=1, the rounds of diffusion k2=4, the rounds of the complete encryption process k3=1. Figure 5 illustrates the encryption and decryption results. Figure 5a is the plain image, Fig. 5b is the cipher image and Fig. 5c is the decryption image.



(a) original plain image (b) cipher image (c) decryption image

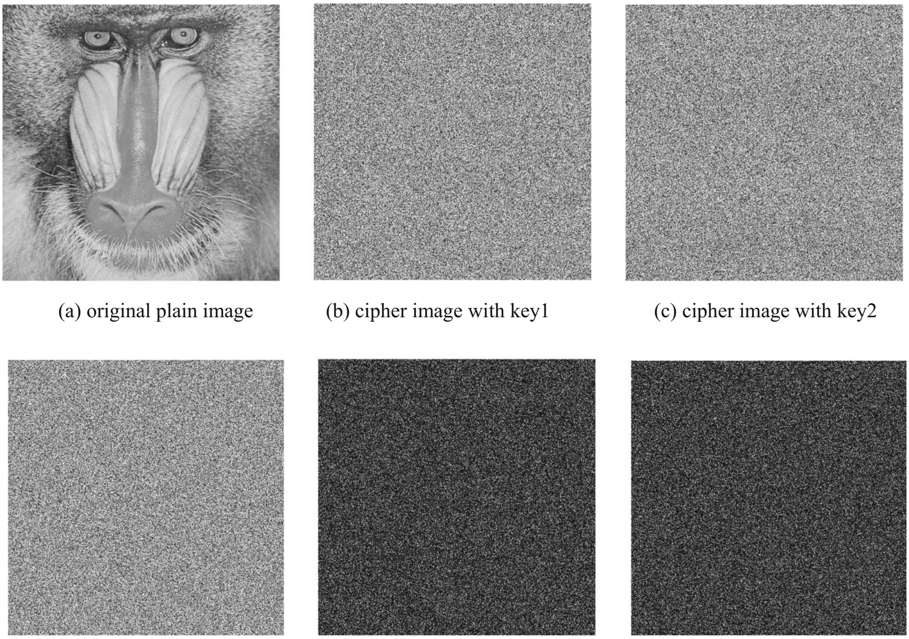
Fig. 5 Encryption result

5 Security analyses

5.1 Key analysis

5.1.1 Key space analysis

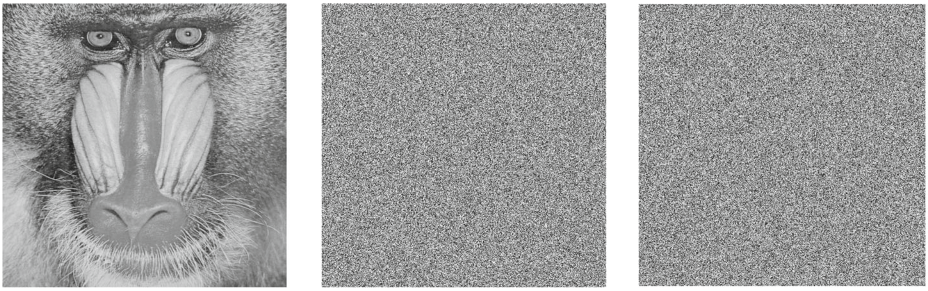
It is generally known that a good encryption scheme should have big key space and be sensitive to the key. In this algorithm, 256-bit key stream based on the plain image is designed, and all initial conditions, parameters are produced by the secret key. Besides, the step length of the movement in every bit plane, the rounds of Brownian motion *R*, confusion *k*1, diffusion *k*2 and complete encryption process *k*3 are all keys. Suppose the precision of a floating-point



(a) original plain image (b) cipher image with key1 (c) cipher image with key2

(d) cipher image with key3 (e) the subtraction of (b) and (c) (f) the subtraction of (d) and (b)

Fig. 6 Key sensitivity at encryption process



(a) decoding correctly (b) decoding with key2 (c) decoding with key3

Fig. 7 Key sensitivity at decryption process

number is 10^{-10}, then our cryptosystem key space is larger than 2^{256}, sufficiently larger than 2^{128} [5]. Therefore, a sufficiently large key space makes the brute force attack infeasible and is ensured for practical applications.

5.1.2 Key sensitivity analysis

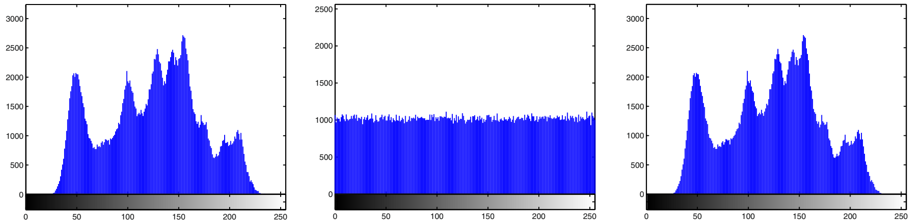
A test image 'Baboon (512×512)' is encrypted, and its 256-bit key stream key1 is '31ED9B78FA294E49148EB892E0801C4E85F5183B8B929D0E971C6ACD5331CB66'. Two slightly changed keys are: key2 = '32ED9B78FA294E49148EB892E0801C4E85F5183B8B929D0E971C6ACD5331CB66' and key3 = '32ED9B78FA294E49148EB892E0801C4E85F5183B8B929D0E971C6ACD5331CB6F'. We use the slightly changed keys to encrypt the plain image, and the results are shown in Fig. 6. Otherwise, the changed keys and the correct key are used to decrypt the cipher image, respectively. The results can be seen in Fig. 7. It is clear that the encrypted image using a slight different key is completely different and the decryption process using a different key completely fails to obtain the plain image.

5.2 Statistical analysis

In order to prove the security of our algorithm resisting statistical analysis, some tests are analyzed.

5.2.1 Histogram

An image histogram illustrates that how pixels in an image are distributed by plotting all the pixels intensity level. 'Lena (512×512)' is used as the plain image, the encryption and decryption results are shown in Fig. 5, and the histogram analysis is shown in Fig. 8.



(a) histogram of original plain image (b) histogram of cipher image (c) histogram of decryption image

Fig. 8 Histogram of plain image, cipher image and decryption image

Table 1 Correlation coefficients of two adjacent pixels in the plain and cipher images

Name	Plain image			Cipher image		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Lena(512×512)	0.9761	0.9626	0.9448	−0.0285	0.0014	0.0013
Baboon(512×512)	0.7142	0.8280	0.6998	0.0013	−0.0281	0.0128
Peppers(512×512)	0.9783	0.9737	0.9498	−0.0282	0.0100	−0.0012
Cameraman(256×256)	0.9593	0.9353	0.9189	0.0139	0.0034	0.0107
Barbara(512×512)	0.9712	0.9339	0.9026	−0.0206	−0.0314	0.0220

It is shown that the histogram of the cipher image is uniformly distributed in contrast to that of the plain image and the histogram of the decryption image is similar to that of the plain image.

5.2.2 Correlation of two adjacent pixels

Five thousand pairs of adjacent pixels from the plain image and cipher image are selected randomly to compute the correlation coefficients. The formulas of correlation coefficients are given as:

$$r_{x,y} = \frac{E((x-E(x))(y-E(y)))}{\sqrt{D(x)D(y)}} \tag{22}$$

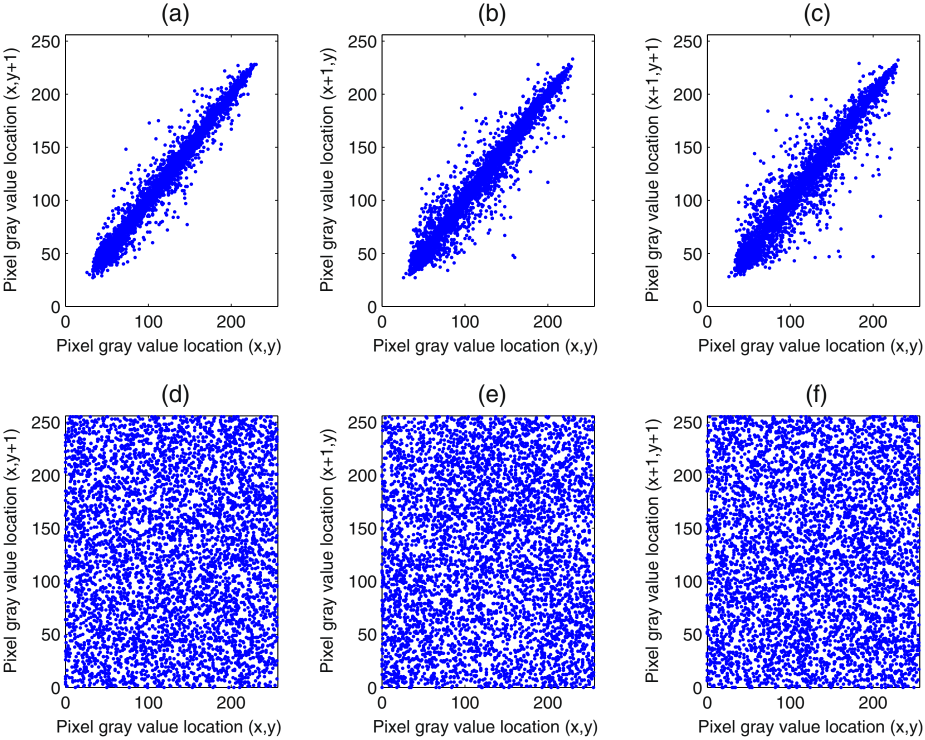


Fig. 9 Correlations of two adjacent pixels, **a** horizontal direction in plain image, **b** vertical direction in plain image, **c** diagonal direction in plain image, **d** horizontal direction in cipher image, **e** vertical direction in cipher image, **f** diagonal direction in cipher image

Table 2 NPCRs and UACIs between different ciphers while plain images only have one different pixel

Name	Average		Minimum		Maximum	
	NPCR	UACI	NPCR	UACI	NPCR	UACI
Lena(512×512)	0.9962	0.3346	0.9956	0.3337	0.9972	0.3354
Baboon (512×512)	0.9961	0.3342	0.9954	0.3334	0.9976	0.3358
Peppers(512×512)	0.9962	0.3344	0.9952	0.3336	0.9974	0.3356
Cameraman(256×256)	0.9960	0.3341	0.9946	0.3326	0.9969	0.3361
Barbara(512×512)	0.9962	0.3345	0.9950	0.3330	0.9970	0.3355

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

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2 \tag{24}$$

Where $E(x)$ and $D(x)$ are the expectation and variance of variable x, respectively. Table 1 lists the results of the correction coefficients. Moreover, the correlation of 5000 pairs of adjacent pixels from the plain image and cipher image of 'Lena (512×512)' are shown in Fig. 9. From Table 1 and Fig. 9, we can see that there are no detectable correlation existing between the plain image and the cipher images.

5.3 Resisting differential attack analysis

In differential attack analysis, two performance indices are usually used to investigate the influence of a 1bit change in the plain image to the corresponding cipher image. They are number of pixels change rate (NPCR) and unified average changing intensity (UACI). NPCR is the number of pixels change rate while one pixel of plain image is changed. The closer NPCR reaches 100 %, the more sensitive the cryptosystem is to the changing of the plain image. UACI is the average intensity of differences between the plain image and cipher image. The bigger UACI is, the more effective is the encryption scheme in resisting differential attack. We calculate the values of NPCR and UACI according to the following formula:

Table 3 NPCR comparison between different algorithms

Name	Ours	Ref. [24]	Ref. [20]	Ref. [4]	Ref. [9]
Lena(512×512)	0.9962	0.9961	0.9963	0.9960	0.9961
Baboon (512×512)	0.9961	0.9962	0.9962	0.9960	0.9961
Peppers(512×512)	0.9962	0.9962	0.9962	0.9962	0.9962
Cameraman(256×256)	0.9960	0.9961	0.9961	0.9961	0.9960
Barbara(512×512)	0.9962	0.9961	0.9961	0.9960	0.9961

Table 4 UACI comparison between different algorithms

Name	Ours	Ref. [24]	Ref. [20]	Ref. [4]	Ref. [9]
Lena(512×512)	0.3346	0.3345	0.3356	0.3342	0.2861
Baboon (512×512)	0.3342	0.3350	0.3342	0.3343	0.2875
Peppers(512×512)	0.3344	0.3342	0.3345	0.3344	0.2957
Cameraman(256×256)	0.3341	0.3340	0.3342	0.3341	0.2955
Barbara(512×512)	0.3345	0.3345	0.3344	0.3342	0.2960

$$\mathrm{NPCR} = \frac{\sum\limits_{i,j} D(i,j)}{W \times H} \times 100\% \tag{25}$$

$$\mathrm{UACI} = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|C_1(i,j){-}C_2(i,j)|}{255}\right] \times 100\% \tag{26}$$

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

$$D(i,j) = \begin{cases} 1, C_1(i,j) \neq C_2(i,j) \\ 0, otherwise \end{cases} \tag{27}$$

Where, W and H denotes the width and height of the image respectively, and $C_1$and C_2 represent the cipher images before one pixel of the plain image is changed.

In the experiment, a pixel of the plain image is randomly changed, and we can have a new plain image. The new plain image and original plain image are chosen to calculate the NPCR and UACI. For every test image, 500 new plain images are gotten, 500 cipher images appear after encryption, the average value, the minimum and the maximum value of NPCR and UACI are calculated through comparing the new cipher image and the old cipher image. The results are illustrated in Table 2. We can see that our algorithm has good properties in resisting differential attacks. In order to compare with other algorithms, the average NPCR and UACI values of each image encrypted by different algorithms are shown in Tables 3 and 4. It can be seen that our method has the same excellent performance as the methods in Refs. [4, 20, 24] and is better than Ref. [9]. Moreover, the two quantitative metrics are used to quantify the key sensitivity, and the results are shown in Table 5. From these results, it can be found that the proposed encryption scheme is highly sensitive to the secret keys.

Table 5 Quantified key sensitivity

Name	Encryption key (Decryption key)	NPCR	UACI
Fig. 6c	key2	0.9957	0.3344
Fig. 6d	key3	0.9961	0.3340
Fig. 7b	key2	0.9958	0.3342
Fig. 7c	key3	0.9963	0.3345

Table 6 Information entropies of the plain and cipher images

Name	Plain image	Ours	Ref. [19]	Ref.[9]	Ref. [18]	Ref. [7]
Lena(512×512)	7.4455	7.9993	7.9992	7.9972	7.9991	7.9992
Baboon (512×512)	7.3579	7.9993	7.9992	7.9973	7.9991	7.9991
Peppers(512×512)	7.5715	7.9993	7.9992	7.9958	7.9991	7.9989
Cameraman(256×256)	7.0097	7.9974	7.9968	7.9970	7.9970	7.9969
Barbara(512×512)	7.4664	7.9992	7.9993	7.9973	7.9992	7.9988

5.4 Information entropy

We use $H(m)$ to represent the information entropy of an information source m, and it can be calculated as

$$H(m) = \sum_{i=0}^{2^n-1} p(m_i)\log\frac{1}{p(m_i)} \tag{28}$$

Where $p(m_i)$ denotes the probability of symbol m_i and the entropy is expressed in bits. For a random image with 256 grey levels, the entropy should ideally be 8. Actually, given that a practical information source seldom generates random messages, then its information entropy value is smaller than 8. The value obtained is very close to 8, and this means information leakage in the encryption process is negligible and the encryption system is secure against entropy attack. The information entropy results are shown in Table 6. By the comparison with Ref. [7, 9, 18, 19], it can be seen that the information entropy of our algorithm are closer to eight, so our scheme is very secure, stable and efficient.

5.5 Computational speed analysis

The encryption speed is an important parameter to evaluate an encryption method. The tests are implemented in Matlab on a personal computer with 3 GHz Intel(R) Core2Duo CPU, 4 GB RAM, the operating system is Windows 7. The speeds of the proposed algorithm and other algorithms are shown in Table 7. From Table 7, we can see that the speed of our algorithm is quicker. Besides, in the proposed encryption method, the computational cost of each round encryption depends on the permutation and diffusion operations. In the permutation process, the plain image is decomposed into 8 bitplanes, then the confusion of 8 bitplanes based on Brownian motion can be implemented in parallel, and this can save much time. Moreover, in the diffusion process, in the detailed steps of row diffusion (RD) and column diffusion (CD), every row or every column can be mapped in parallel. And so our encryption scheme can run fast.

Table 7 Encryption speed comparison

Encryption algorithm	Ours	AES (128-bitkey)	AES (192-bitkey)	AES (256-bitkey)	Ref. [10]	Ref. [13]	Ref. [16]
Speed (M bit/s)	13.02	10.81	9.06	8.94	1.86	0.25	8.46

6 Conclusions

In the paper, a new image encryption algorithm based on bit level Brownian motion and chaotic systems is proposed and the security of the algorithm is analyzed. SHA256 hash function is used to produce 256 bit key stream, the initial values and system parameters of the new 1D chaotic systems for the confusion and diffusion are generated, they are related to the plain image and so the scheme can resist the known plaintext and chosen plaintext attack. Moreover, the plain image is decomposed into 8 bitplanes, the pixel position is scrambled based on Brownian motion in every plane, the bitplanes are combined to get the confused image, the positions and values of all pixels are changed simultaneously in confusion process, and then our algorithm has high security. In addition, two directional diffusions consisting of row diffusion (RD) and column diffusion (CD) are introduced, the change of any pixel may have impact on all pixels, and then our method has higher security. Besides, many security analysis experiments are carried out, including key space analysis, key sensitivity analysis, histogram analysis, correlation analysis, resisting differential attack analysis, information entropy and speed analysis. The results show that the proposed image encryption scheme is secure and has important practical application in secure communication.

Acknowledgments This work is supported by the National Natural Science Foundation of China (Grant No. 61203094), Science and Technology Foundation of Henan Province of China (Grant No. 152102210048), Natural Science Foundation of Educational Committee of Henan Province of China (Grant No. 14A413015), the joint funds between Henan Provincial Government and Ministry of Education of China (Grant No. SBGJ090603), the Research Foundation of Henan University (Grant No. xxjc20140006).

References

1. Auli-Llinas F, Marcellin MW (2012) Scanning order strategies for bitplane image coding. IEEE Trans Image Process 21(4):1920–1933
2. Chen G, Mao Y, Chui CK (2004) A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos, Solitons Fractals 21(3):749–761
3. Chen JX, Zhu ZL, Fu C, Yu H (2014) A fast image encryption scheme with a novel pixel swapping-based confusion approach. Nonlinear Dyn 77(4):1191–1207
4. Fouda JSAE, Effa JY, Sabat SL, Ali M (2014) A fast chaotic block cipher for image encryption. Commun Nonlinear Sci 19(3):578–588
5. Francoisa M, Grosgesa T, Barchiesia D, Errab R (2012) A new image encryption scheme based on a chaotic function. Signal Process Image Commun 27:249–259
6. Fu C, Meng WH, Zhan YF, Zhu ZL, Lau FC, Tse CK, Ma HF (2013) An efficient and secure medical image protection scheme based on chaotic maps. Comput Biol Med 43(8):1000–1010
7. Gao TG, Chen ZQ (2008) A new image encryption algorithm based on hyper-chaos. Phys Lett A 372(4):394–400
8. Hua ZY, Zhou YC, Pun CM, Chen CLP (2015) 2D sine Logistic modulation map for image encryption. Inf Sci 297:80–94
9. Kulsoom A, Xiao D, Aqeel-ur-Rehman, Abbas SA (2014) An efficient and noise resisitive selective image encryption scheme for gray images based on chaotic maps and DNA complementary rules. Multimed Tools Appl. doi:10.1007/s11042-014-2221-x
10. Liao XF, Lai SY, Zhou Q (2010) A novel image encryption algorithm based on self-adaptive wave transmission. Signal Process 90:2714–2722
11. Ling C, Wu X, Sun S (1999) A general efficient method for chaotic signal estimation. IEEE Trans Signal Process 47(5):1424–1428
12. Liu Q, Li PY, Zhang MC, Sui YX, Yang HJ (2015) A novel image encryption algorithm based on chaos maps with Markov properties. Commun Nonlinear Sci 20(2):506–515

13. Liu SB, Sun J, Xu ZQ (2009) An improved image encryption algorithm based on chaotic system. J Comput 4:1091–1100
14. Liu HJ, Wang XY (2011) Color image encryption using spatial bit-level permutation and high-dimension chaotic system. Opt Commun 284(16–17):3895–3903
15. Matthes R (1989) On the derivation of a chaotic encryption algorithm. Cryptologia 13(1):29–42
16. Mazloom S, Eftekhari-Moghadam AM (2009) Color image encryption based on coupled nonlinear chaotic map. Chaos, Solitons Fractals 42:1745–1754
17. Moon D, Chung Y, Pan SB, Moon K, Chung K (2006) An efficient selective encryption of fingerprint images for embedded processors. Electron Telecommun Res Inst J 28(4):444–452
18. Wang Y, Wong KW, Liao XF, Chen GR (2011) A new chaos-based fast image encryption algorithm. Appl Soft Comput 11(1):514–522
19. Wang XY, Xu DH (2014) A novel image encryption scheme based on Brownian motion and PWLCM chaotic system. Nonlinear Dyn 75(1–2):345–353
20. Wang XY, Xu DH (2015) A novel image encryption scheme using chaos and Langton's Ant cellular automaton. Nonlinear Dyn 79(4):2449–2456
21. Xu Y, Wang H, Li YG, Pei B (2014) Image encryption based on synchronization of fractional chaotic systems. Commun Nonlinear Sci 19(10):3735–3744
22. Ye G (2010) Image scrambling encryption algorithm of pixel bit based on chaos map. Pattern Recogn Lett 31(5):347–354
23. Ye GD (2014) A block image encryption algorithm based on wave transmission and chaotic systems. Nonlinear Dyn 75(3):417–427
24. Zhang XP, Fan X, Wang JY, Zhao ZM (2014) A chaos-based image encryption scheme using 2D rectangular transform and dependent substitution. Multimed Tools Appl. doi:10.1007/s11042-014-2372-9
25. Zhang YQ, Wang XY (2014) A symmetric image encryption algorithm based on mixed linear-nonlinear coupled map lattice. Inf Sci 273:329–351
26. Zhang YQ, Wang XY (2014) Spatiotemporal chaos in mixed linear-nonlinear coupled logistic map lattice. Phys A 402:104–118
27. Zhang W, Wong Kwok-wo YH, Zhu ZL (2013) An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion. Commun Nonlinear Sci 18(8):2066–2080
28. Zhou YC, Bao L, Chen CLP (2013) Image encryption using a new parametric switching chaotic system. Signal Process 93(11):3039–3052
29. Zhou YC, Bao L, Chen CLP (2014) A new 1D chaotic system for image encryption. Signal Process 97:172–182
30. Zhou YC, Cao WJ, Chen CLP (2014) Image encryption using binary bitplane. Signal Process 100:197–207
31. Zhou YC, Panetta K, Agaian S, Chen CLP (2012) Image encryption using P-Fibonacci transform and decomposition. Opt Commun 285(5):594–608
32. Zhu CX, Xu SY, Hu YP, Sun KH (2014) Breaking a novel image encryption scheme based on Brownian motion and PWLCM chaotic system. Nonlinear Dyn 79(2):1511–1518

Xiuli Chai received the PhD Degree in 2008 from South China University of Technology. Currently, working as an associate professor at Henan University, she is mainly engaged in information security, multimedia security, cryptography aspects of chaos.