Commun Nonlinear Sci Numer Simulat 20 (2015) 846–860

Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

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

A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism

Jun-xin Chen [a], Zhi-liang Zhu [b,*], Chong Fu [a], Hai Yu [b], Li-bo Zhang [b]

[a] School of Information Science and Engineering, Northeastern University, No. 11, Lane 3, WenHua Road, Shenyang 110004, Liaoning, China
[b] Software College, Northeastern University, No. 11, Lane 3, WenHua Road, Shenyang 110004, Liaoning, China

ARTICLE INFO

Article history:
Received 2 November 2013
Received in revised form 12 June 2014
Accepted 18 June 2014
Available online 3 July 2014

Keywords:
Image encryption
Dynamic state variables selection
Pixel-swapping based confusion
Snake-like diffusion

ABSTRACT

In recent years, a variety of chaos-based image cryptosystems have been investigated to meet the increasing demand for real-time secure image transmission. Most of them are based on permutation–diffusion architecture, in which permutation and diffusion are two independent procedures with fixed control parameters. This property results in two flaws. (1) At least two chaotic state variables are required for encrypting one plain pixel, in permutation and diffusion stages respectively. Chaotic state variables produced with high computation complexity are not sufficiently used. (2) The key stream solely depends on the secret key, and hence the cryptosystem is vulnerable against known/chosen-plain-text attacks. In this paper, a fast chaos-based image encryption scheme with a dynamic state variables selection mechanism is proposed to enhance the security and promote the efficiency of chaos-based image cryptosystems. Experimental simulations and extensive cryptanalysis have been carried out and the results prove the superior security and high efficiency of the scheme.

1. Introduction

With the dramatic development of communication technologies, digital image application and exchange across Internet have become much more prevalent than the past. Cryptographic approaches are therefore critical for secure image transmission and storage over public networks. However, traditional encryption algorithms are typically designed for textual information and have been found not suitable for image encryption due to some intrinsic features of images such as high pixel correlation and redundancy [1]. Since 1990s, many researchers have noticed that the fundamental features of chaotic systems such as ergodicity, mixing property, unpredictability, sensitivity to initial conditions/system parameters, *etc.* can be considered analogous to some ideal cryptographic properties for image encryption [2,3]. In 1998, permutation–diffusion architecture for chaos-based image encryption was firstly proposed by Fridrich [4]. Under this structure, a plain image is firstly shuffled by a two-dimensional area-preserving chaotic map with the purpose to erase the high correlation between adjacent pixels. Then pixel values are modified sequentially using pseudorandom key stream elements produced by a certain qualified chaotic map in the diffusion procedure. This architecture forms the basis of numerous chaos-based image cryptosystems proposed subsequently [5–28]. During the past decades, improvements to this architecture have been extensively developed in various aspects, such as novel pixel-level confusion techniques [5–9], bit-level permutation approaches

* Corresponding author. Tel.: +86 24 86581232.
E-mail address: zhuzhiliang.sc@gmail.com (Z.-l. Zhu).

http://dx.doi.org/10.1016/j.cnsns.2014.06.032
1007-5704/

[10–14], improved diffusion strategies [15–19], applications of plain-image related parameters [20–22] and enhanced key stream generators [23–28].

Meanwhile, recent cryptanalysis works have demonstrated that some chaos-based image cryptosystems are insecure against various attacks, and have been successfully broken [29–36]. The weaknesses in these insecure algorithms include insensitiveness to the changes of the plain image, weak secret keys, and the most serious one is that the key stream is completely depending on the secret key. That means identical key stream will be used to encrypt different plain images if the secret key remains unchanged. This property allows the attacker to launch known-plaintext attack [30–33,35] or chosen-plaintext attack [30,31,33,35,36] so as to retrieve the equivalent key stream elements. Therefore, to further enhance the security, the key stream elements extracted from the same secret key should better be distinct and related to the plain image [20]. Some general rules for evaluating the security performance of a chose-based cryptosystem can be found in [1,2]. Besides, the permutation and diffusion are treated as two independent procedures in most of the existing image cryptosystems. Two-dimensional and one-dimensional chaotic maps are generally employed to achieve cryptographic requirements in the permutation and diffusion stages, respectively. Chaotic state variables generated in both stages are merely used for respective encrypting operation, which means at least two chaotic state variables are required for ciphering one pixel. Chaotic state variables that are calculated with high computation complexity are not sufficiently used. The present paper proposes a novel chaos-based image encryption scheme with a dynamic state variables selection mechanism (DSVSM). This cryptosystem can satisfy the security requirements suggested in [1,2] and well address the flaws existing in the cracked algorithms by employing innovations in four aspects. (1) Chaotic state variables used in our cryptosystem are generated from three-dimensional or hyper chaotic systems, and will be shared in permutation and diffusion procedures. Accordingly, a slight change in the secret key will not only affect the diffusion module but also influence the permutation procedure simultaneously. Besides, the chaotic state variables sharing mechanism can also significant advance the utilization efficiency of the chaotic map iteration. (2) The state variable allotted for each pixel's encryption is decided by DSVSM, which is plain pixel-related. When ciphering different plain images, distinct key streams will be produced both in the permutation and diffusion procedures, even though adopt the same secret key. The attacker cannot obtain useful information by encrypting some special images, as the resultant information is self-related to the chosen-images. This property ensures the resistance to known/chosen-plaintext attacks. (3) Pixel-swapping based image confusion approach is proposed as a replacement of the traditional permutation approaches. This confusion strategy can produce confusion and certain diffusion effects simultaneously in the permutation stage, so as to accelerate the overall diffusion effect of the cryptosystem. (4) Image diffusion in our scheme is implemented in snake-like mode. In coordination with the pixel-swapping based confusion strategy and DSVSM, the difference spreading effect produced in the confusion stage can be further scattered to the whole cipher image in the first round diffusion. The efficiency of the cryptosystem is therefore remarkably improved. Experiment results demonstrate that the proposed scheme has a high security level and satisfactory operation efficiency for practical secure image applications.

The remaining of this paper is organized as follows. In next section, the architecture of typical chaos-based image cryptosystems is introduced. Then the proposed image encryption scheme will be described in detail in Section 3. Simulation results, the effectiveness and efficiency of the proposed scheme are reported in Section 4. Thorough security analyzes of the cryptosystem are carried out in Section 5. Finally, conclusions will be drawn in the last section.

2. Architecture of typical chaos-based image cryptosystems

The architecture of typical chaos-based image cryptosystems is shown in Fig. 1. There are two stages in the cryptosystems of this kind, namely, the permutation stage and diffusion stage.

In the permutation stage, image pixels are generally shuffled by a two-dimensional area-preserving chaotic map, without any modification to their values. Traditionally, three types of chaotic maps, Arnold cat map, standard map and baker map are employed, and their discretized versions are given by Eqs. (1)–(3), respectively.

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \bmod N, \tag{1}$$

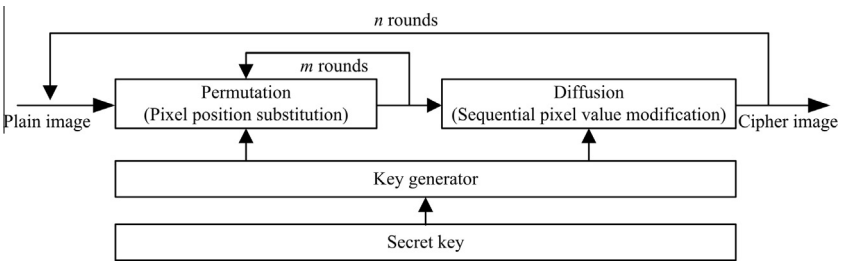

Fig. 1. The architecture of typical chaos-based image cryptosystems.

$$\begin{cases} x_{i+1} = (x_i + y_i) \bmod N \\ y_{i+1} = (y_i + K \sin \frac{x_{i+1} N}{2\pi}) \bmod N \end{cases}, \tag{2}$$

$$\begin{cases} x_{i+1} = \frac{N}{n_j}(x_i - N_j) + y_i \bmod \frac{N}{n_j} \\ y_{i+1} = \frac{k_j}{N}\left(y_i - y_i \bmod \frac{N}{n_j}\right) + N_j \end{cases} \quad where \begin{cases} n_0 + n_1 + \cdots + n_t = N \\ N_j = n_0 + n_1 + \cdots + n_j \\ 0 \leqslant y_i \leqslant N \\ N_j \leqslant x_i \leqslant N_j + n_{j+1} \\ 0 \leqslant j \leqslant t-1 \\ n_0 = 0 \end{cases}. \tag{3}$$

In these equations, N is the width or height of the square image, (x_i, y_i) and (x_{i+1}, y_{i+1}) are the original and permuted pixel positions, (p, q), $n_j (j = 1, 2, \ldots, t-1)$ and K represent the control parameters of these maps, respectively. All pixels are scanned sequentially from upper-left corner to lower-right corner, and then the confused image is produced.

In the diffusion phase, pixel values will be modified sequentially by mixing with the key stream elements that are generated by a one-dimensional chaotic map so as to confuse the relationship between cipher image and plain image. Generally, the modification to a particular pixel not only depends on the corresponding key stream element but also the accumulated effect of the previous pixel values. A typical diffusion operation is illustrated in Eq. (4), where $p(n)$, $k(n)$, $c(n)$, and $c(n-1)$ represent the current plain pixel, key stream element, output cipher-pixel and the previous cipher-pixel, respectively. Such diffusion approach can spread a slight change of the plain image to a large scale in the ciphered image and thus differential attack may be practically useless. Additionally, to cipher the first pixel, $c(-1)$ has to be set as a seed.

$$c(n) = k(n) \oplus p(n) \oplus c(n-1). \tag{4}$$

Now we can infer from Eqs. (1)–(3) that the confused image totally depends on the control parameters of the permutation maps, the difference between plain images will be moved to a new position rather than spread out in the permutation stage. In other words, there is no diffusion effect when using traditional permutation techniques. So the duty of image diffusion entirely relies on the diffusion module, which is the highest cost of the cryptosystem. Therefore, it is of great significance to investigate a novel permutation approach that can simultaneously produce certain diffusion effect so as to accelerate the diffusion performance of the cryptosystem. Besides, when using the same secret key, distinct key streams should better be produced for ciphering different plain images so as to effectively resist the known/chosen-plaintext attacks [20].

3. The proposed scheme

3.1. The DSVSM

In this section, DSVSM was given out to improve the security and efficiency of image cryptosystems. This mechanism is proposed for dynamically assigning chaotic state variables for encrypting each pixel, and can collaborate with any three-dimensional or hyper chaotic systems that are used for key stream generation. In this paper, the Chen's chaotic system is employed as an example for illustrating DSVSM clearly, as described by Eq. (5), in which a, b, and c are control parameters, when $a = 35, b = 3$, and $c \in [20, 28.4]$ the system is chaotic. The initial system variables x_0, y_0, z_0 and the control parameter c constitute the secret key.

$$\begin{cases} \frac{dx}{dt} = a(y - x) \\ \frac{dy}{dt} = (c - a)x - xz + cy \\ \frac{dz}{dt} = xy - bz \end{cases}. \tag{5}$$

With each of the Chen's chaotic map iteration, three state variables will be generated, denoted as X, Y and Z, respectively. In DSVSM, the chaotic state variables that will be used for key stream element generation are selected according to the previous processed pixel. The current processing image (plain image in the confusion phase or the confused image in the diffusion stage) with $M \times N$ pixels are viewed as a one-dimensional array, pixels are represented by $P = \{P(0), P(1), \ldots, P(M \times N - 1)\}$ from the upper-left corner to the lower-right corner.

In order to demonstrate the DSVSM properly, the following definitions have to be declared firstly.

1. The state variable used for each pixel's encryption will be selected from a state variables combination, which consists of a state value of X, Y and Z. Assume that $\{X_i, Y_j, Z_k\}$ is the current state variables combination, where X_i, Y_j and Z_k are the states of X, Y and Z in ith, jth and zth iteration, respectively. Note that, i, j and k are not necessarily equal to each other in our scheme. Without loss of generality, we suppose that $i \leqslant j \leqslant k$, as depicted in Fig. 2(a).
2. Let $P(L)$ denotes the current processing pixel, which means the state variables sequences $\{X_0, X_1, \ldots, X_{i-1}\}$, $\{Y_0, Y_1, \ldots, Y_{j-1}\}$ and $\{Z_0, Z_1, \ldots, Z_{k-1}\}$ had been selected for encrypting the pixels $\{P(0), P(1), \ldots, P(L-1)\}$. We can easily come to the conclusion that

$$i + j + k = L.$$

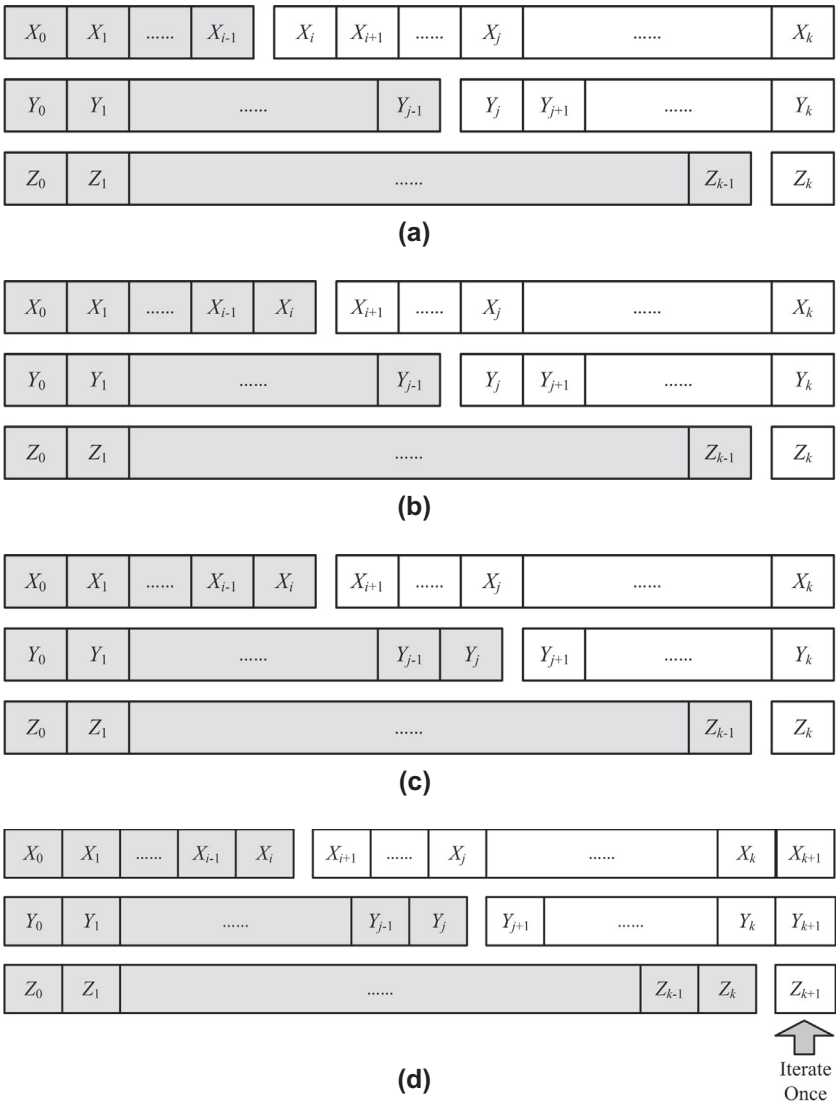

Fig. 2. The operation procedures of DSVSM.

3. We define $slt(L)$ as the selected variable in $\{X_i,\ Y_j,\ Z_k\}$ that will be used to produce the key stream element for $P(L)$. The decision will be made according to an indicator $index(L)$, defined as Eq. (6).

$$index(L) = P(L-1)\%3. \tag{6}$$

The state values X_i, Y_j and Z_k will be selected when $index(L) = 0, index(L) = 1$ and $index(L) = 2$, respectively. For the first pixel, initial value $P(\text{-}1)$ has to be set as a seed.

The main concepts of DSVSM are demonstrated in Fig. 2, and the operation procedures are described as follows:

1. Calculate $index(L)$ according to Eq. (6). Without loss of generality, we assume that $index(L) = 0$.
2. The state value X_i is chosen for $P(L)$, and the state variables combination is updated to $\{X_{i+1},\ Y_j,\ Z_k\}$, as illustrated by Fig. 2(b).
3. Calculate $index(L+1)$, and we assume that $index(L+1) = 1$.
4. The state value Y_j will be distributed to $P(L+1)$, and the state variables combination changes to $\{X_{i+1},\ Y_{j+1},\ Z_k\}$, as depicted in Fig. 2(c).
5. Calculate $index(L+2)$, and we suppose that $index(L+2) = 2$, without loss of generality.

6. The state value Z_k will be chosen for ciphering $P(L+2)$. As Z_k is the last element of the chaotic state Z, Chen's chaotic map will iterate once so as to produce sufficient state variables to make the state variables combination qualified for the subsequent encryption. Then the state variables combination will update to $\{X_{i+1},\ Y_{j+1},\ Z_{k+1}\}$, as shown in Fig. 2(d).
7. Repeat the above steps for all the pixels.

Now we can come to the following intrinsic features of DSVSM from the abovementioned procedures.

1. Iteration efficiency. In DSVSM, the chaotic map is iterated when necessary, as shown in Fig. 2(d). The iteration count is not necessarily equal to the pixel number of the plain image, and it depends on the distribution of the pixel values. In our simulation of ciphering Lena, little more than $1/3 \times (M \times N)$ times iteration are required in one overall encryption round. The simulation results and detailed analyzes will be reported in Section 4.
2. Dynamic property and diffusion effect. Suppose that there is a slight difference in $P(L-1)$, and the corresponding $index(L)$ changes from 0 to 1. Therefore, $slt(L)$ changes to Y_j, then the subsequent state variables combination varies to $\{X_i,\ Y_{j+1},\ Z_k\}$, and hence $slt(L+1)$ varies to Y_{j+1}. Analogously, the difference will spread out to all the subsequent state variables selection operations and bring about different key streams.

3.2. Pixel-swapping based confusion strategy

3.2.1. Pixel-swapping based confusion strategy

In the proposed confusion approach, the plain image and the confused image with size of $M \times N$ are viewed as one-dimensional arrays, and the pixels are represented by $P = \{P(0),\ P(1),\ \ldots,\ P(M \times N - 1)\}$ and $C = \{C(0),\ C(1),\ \ldots,\ C(M \times N - 1)\}$ from the upper-left corner to the lower-right corner, respectively. In this approach, the confusion effect is obtained by performing nonlinear pixel swapping operations. Each pixel in the plain image will be swapped with another one located after it. Suppose that X is the coordinate of the current pixel and X' is the position of the corresponding swapping one, X' is calculated by Eq. (7), where $k_c(X)$ is the current confusion key stream element.

$$X' = X + k_c(X). \tag{7}$$

Pixel swapping operation will be performed according to Eqs. (8) and (9).

$$C(X) = P(X') = P(X + k_c(X)), \tag{8}$$
$$P(X') = P(X). \tag{9}$$

As all pixel swapping operations are performed forward, and hence $C(X)$ cannot be changed again, so $C(X)$ will be the resultant pixel of the confused image. However, after $P(X)$ is swapped to X' and serves as the new value of $P(X')$, it may be swapped again with another pixel in the subsequent pixel swapping operations. Therefore, $P(X)$ can not be viewed as $C(X')$ whereas $C(X')$ will be produced when pixel swapping operation is implemented to location X'. Besides, the confusion operations are directly performed in the space of the original image, no extra space has to be built for storing the confused image. With each of the pixel swapping operation, the confused image is grown by one pixel, whereas the original image will lose one pixel. Pixel swapping operations are performed from the first to the last pixel and the confused image will be produced finally.

In our confusion scheme, $k_c(X)$ is produced according to Eq. (10), where $floor(x)$ returns the value nearest integers less than or equal to $x, abs(x)$ is the absolute value of x, $\mathrm{mod}(x, y)$ returns the remainder after division, $slt(n)$ represents the selected state variable for $P(n)$ according to DSVSM.

$$k_c(n) = \mathrm{mod}[(abs(slt(n)) - floor(abs(slt(n)))) \times 10^{14}, \quad M \times N - n]. \tag{10}$$

Collaborated with DSVSM, certain diffusion effects in addition to satisfactory confusion effects can be simultaneously obtained. When using traditional permutation methods, the differential pixel in the plain image will be moved to the corresponding shuffled location, which is completely determined by the permutation maps. The subsequent permutation process will be in-order implemented in the same way, as the differential pixel has no influence to the subsequent permutation

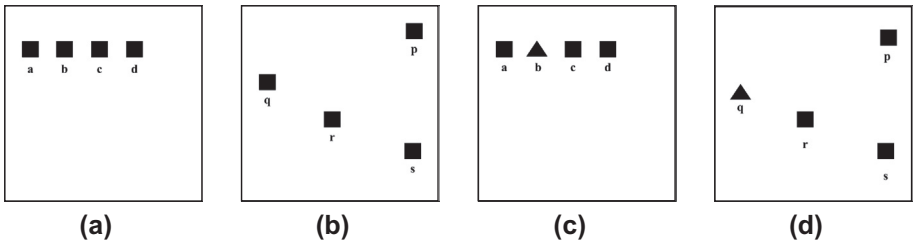

Fig. 3. Confusion and diffusion effects of traditional permutation methods: (a) plain image; (b) confused image of (a); (c) plain image with one differential pixel; (d) confused image of (c).

operations. As shown in Fig. 3(d), the difference in the pixel b has no influence on the subsequent permutation processes, and hence pixels at c and d will be shuffled to the location r and s, the same as the confusion result shown in Fig. 3(b). On the other hand, pixels at p, q, r and s will be swapped to the location a, b, c and d when using the pixel-swapping based confusion approach, as shown in Fig. 4(a) and (b). Suppose that there is a slight difference in the pixel at q in the plain image, as shown in Fig. 4(c), after it was swapped to b as the confused pixel $C(b)$, the difference will affect the subsequent pixel swapping operation for c according to Eqs. (7)–(10). Different from the confusion result shown in Fig. 4(b), pixels at y rather than r will swap to location c as the confusion result. Analogously, pixel at z will be swapped to d. Similarly, the remainder pixel swapping operation will be influenced, so the difference will spread out to the remaining confused image and certain diffusion effects is produced.

3.2.2. Simulation results of pixel-swapping based confusion strategy

Numbers of simulations have been performed to testify the confusion and diffusion effects of the pixel-swapping based confusion approach. Besides the pixel correlation performance that is the basic measurement of the confusion approach, two indices, namely, *NPCR* (number of pixels change rate) and *UACI* (unified average changing intensity) are utilized to measure the influence of one pixel difference in plain image on the entire cipher image. Suppose that $P_1(i,j)$ and $P_2(i,j)$ be the $(i,j)th$ pixel of two images P_1 and P_2, respectively, *NPCR* and *UACI* are defined in Eqs. (11)–(13).

$$NPCR = \frac{\sum_i \sum_j D(i,j)}{W \times H} \times 100\%, \tag{11}$$

where W and H are the width and length of P_1 and P_2 and $D(i,j)$ is defined as

$$D(i,j) = \begin{cases} 0 & if \quad P_1(i,j) = P_2(i,j) \\ 1 & if \quad P_1(i,j) \neq P_2(i,j) \end{cases}. \tag{12}$$

$$UACI = \frac{1}{W \times H}\left[\sum_i \sum_j \frac{|P_1(i,j) - P_2(i,j)|}{L-1}\right] \times 100\%. \tag{13}$$

Table 1 lists these indices of the proposed confusion approach and traditional permutation strategies. The testing images are the standard 256 gray scale 512×512 Lena image and its modified version obtained by changing the value of the lower-right pixel from 108 to 109. Our simulation computing platform is a personal computer with an Intel (R) Core (TM) i5 CPU (2.27 GHz), 2 GB memory and 320 GB hard-disk capacity. Chen's chaotic map with control parameters $(x_0 = 3.3,\ y_0 = -2.1,\ z_0 = 6.8,\ c = 28)$ is employed for chaotic state variables generation.

To clearly demonstrate the difference spreading effect of the proposed permutation strategy, the shuffled images produced by the proposed confusion approach and 3 rounds cat map are depicted in Fig. 5. Fig. 5(a) and (b) show the standard Lena and its modified version. Fig. 5(c) and (d) are the confused images after 1 round pixel-swapping based permutation, whereas Fig. 5(f) and (g) demonstrate the confused images using 3 rounds cat map. Fig. 5(e) shows the differential image between Fig. 5(c) and (d). We can see that the tiny difference in the plain image has led to continuous large scale different pixels in the permutated image, which implies that impressive diffusion effect is obtained. On the other hand, as there is no difference spreading effect when using cat map, there is only one different pixel between Fig. 5(f) and (g), and the differential image is shown in Fig. 5(h). Besides, the difference spreading effects in other positions have also been tested, and will be demonstrated in detail in Section 4.

3.3. Image diffusion in snake-like mode

In the diffusion stage, pixel values are modified sequentially according to Eq. (4). The key stream elements $k(n)$ used for masking is calculated by Eq. (14) in which $slt(n)$ denotes the selected state variable in the diffusion stage according to DSVSM and *Gray* is the gray-level of the plain image.

$$k_c(n) = \mathrm{mod}[(abs(slt(n)) - floor(abs(slt(n)))) \times 10^{14},\ Gray]. \tag{14}$$

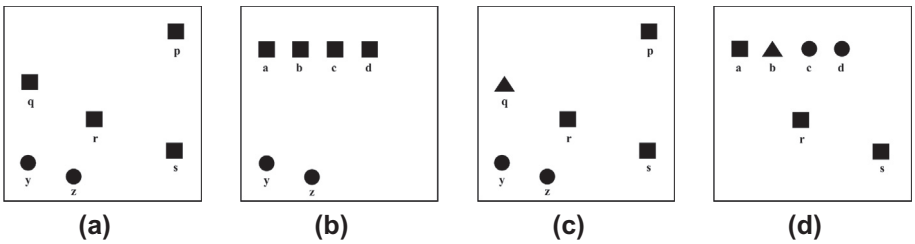

Fig. 4. Confusion and diffusion effects of the proposed approach: (a) plain image; (b) confused image of (a); (c) plain image with one differential pixel; (d) confused image of (c).

Table 1
Simulation results of the proposed and traditional permutation techniques.

Permutation approaches	Rounds	Pixel correlation			*NPCR*	*UACI*
		Horizontal	Vertical	Diagonal		
Plain image	–	0.9844	0.9697	0.9548	–	–
Proposed	1	−0.0032	0.0029	−0.0019	73.38%	15.87%
Arnold cat map	3	0.0276	−0.0226	−0.0199	3.8147e−006	1.4960e-008
Baker map	3	0.0135	−0.0154	0.0221	3.8147e−006	1.4960e−008
Standard map	3	0.0109	−0.0070	−0.0086	3.8147e−006	1.4960e−008

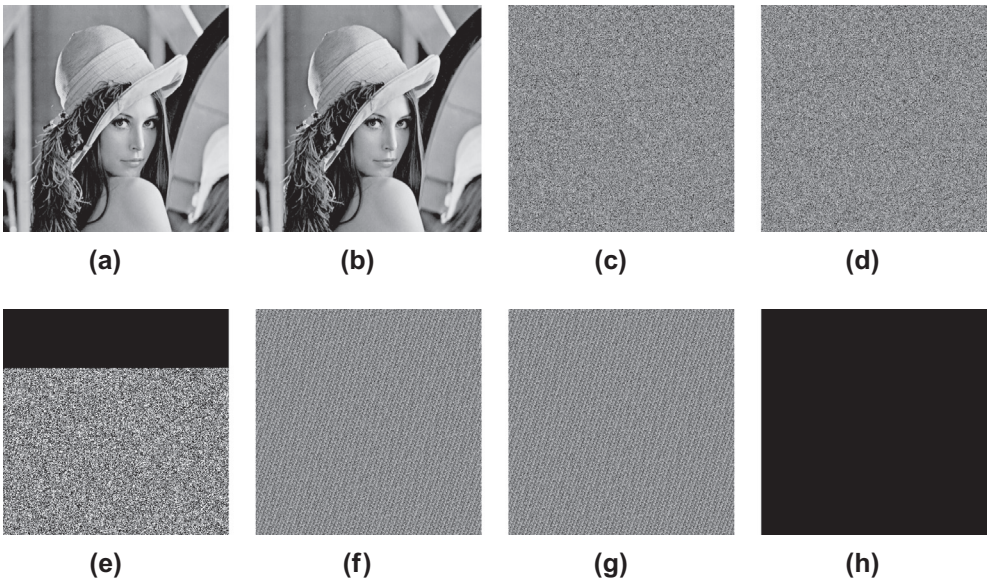

Fig. 5. Simulation results of confusion techniques: (a) Lena image; (b) modified Lena image; (c) confused image of (a) using pixel-swapping based confusion strategy; (d) confused image of (b) using pixel-swapping based confusion strategy; (e) differential image between (c) and (d); (f) confused image of (a) using 3 rounds cat map; (g) confused image of (b) using 3 rounds cat map; (h) differential image between (f) and (g).

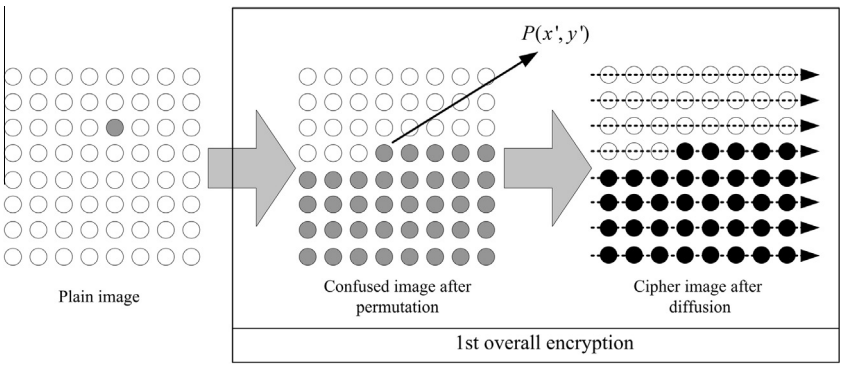

Fig. 6. Difference spreading effect in normal direction.

An efficient diffusion procedure should introduce the difference as quickly as possible so as to spread out the difference at an early age and get better diffusion effect. In the present method, image diffusion is performed in snake-like mode. The purpose is to introduce the difference produced in the permutation stage at the beginning of the diffusion procedure and spread it out to the whole cipher image within the first encryption round. The difference of the spreading effect between the diffusion process in normal direction and snake-like mode is depicted in Figs. 6 and 7. Suppose that there is only one different pixel between the two plain images, then the difference will spread to a larger scale from (x', y') to the last pixel after the pixel-swapping based confusion operation. As shown in Fig. 6, diffusion operation performed in normal direction will spread the difference from (x', y') to all the subsequent pixels in the same path, but cannot scatter the difference to a wider scale. In

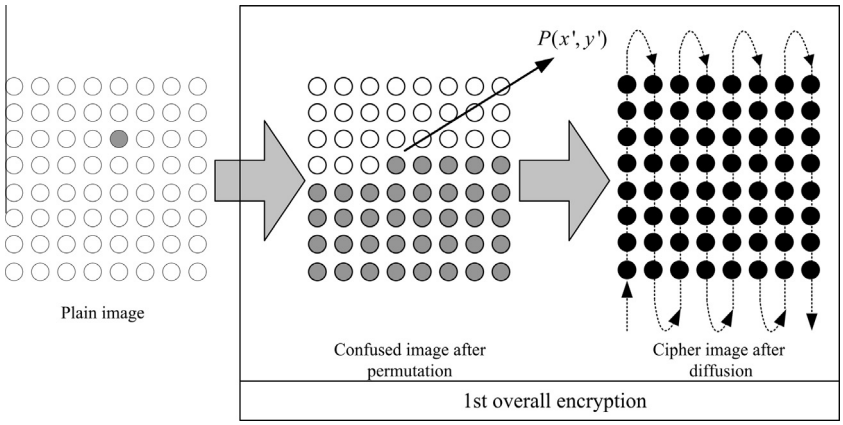

Fig. 7. Difference spreading effect in snake-like mode.

contrast, if the diffusion procedure is implemented in snake-like mode, the difference can be introduced at the beginning of the diffusion stage. By affecting the key stream element generation through DSVSM, the difference can spread out to the whole cipher image in the first encryption round, as illustrated by Fig. 7. Therefore, collaborated with the pixel-swapping based confusion approach and DSVSM, diffusion operation in snake-like mode can remarkably accelerate the difference spreading process, and hence the required encryption rounds can be reduced while maintaining the security level.

3.4. Architecture of the proposed image encryption scheme

The overall architecture of the proposed cryptosystem is shown in Fig. 8, and the operation procedures are described as follows:

Step 1: Perform one round pixel-swapping based confusion.

1. Iterate Eq. (5) with $(x_0,\ y_0,\ z_0,\ c)$ for N_0 times continuously to avoid the harmful effect of transitional procedure, where N_0 is a constant.
2. Get the current state variable according to DSVSM. For the first pixel, an initial value has to be set as a seed.
3. Calculate the location of the corresponding swapping pixel using Eqs. (7) and (10).
4. Swapping the pixels according to Eqs. (8) and (9).
5. Go back to 2 until all pixels are confused.

Step 2: Perform one round image diffusion in snake-like mode.

1. Obtain the current state variable using DSVSM. For the first pixel, an initial value has to be set as a seed.
2. Calculate the key stream element according to Eq. (14) and mask the plain pixel value using Eq. (4)
3. Go back to 6 until all pixels are encrypted.

Step 3 : Repeat the above steps n times to satisfy the security requirements.

4. Simulation results and discussions

A number of tests have been carried out with different plain images and numbers of encryption rounds to demonstrate the effectiveness and efficiency of the proposed cryptosystem. The algorithm is simulated by running the standard C program on our computing platform as described before and the compile environment is Code Blocks 10.05. The parameters of Chen's chaotic map is $x_0 = 3.3,\ y_0 = -2.1,\ z_0 = 6.8$ and $c = 28$.

4.1. Effectiveness analysis

The effectiveness of the proposed image cryptosystem is verified by the following two tests.

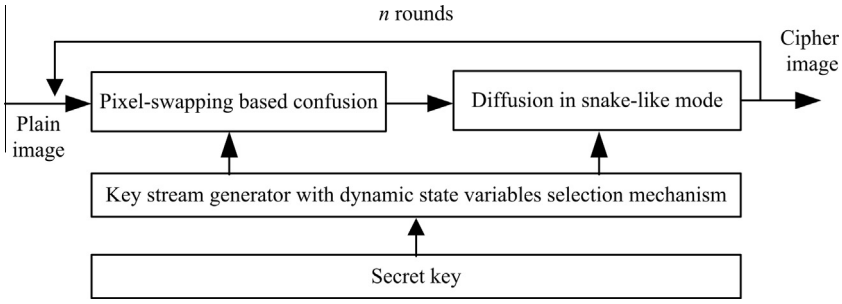

Fig. 8. The architecture of the proposed image cryptosystem.

Table 2
Effectiveness testing results of the proposed scheme (1).

Test images	1 Round				2 Rounds	
	Permutation only		Overall encryption		Overall encryption	
	NPCR (%)	*UACI* (%)	*NPCR* (%)	*UACI* (%)	*NPCR* (%)	*UACI* (%)
Lena	73.38	15.87	99.62	33.43	99.62	33.48
Baboon	63.24	12.06	99.61	33.51	99.61	33.49
Barb	27.87	5.95	99.61	33.46	99.63	33.51
Bridge	86.76	21.50	99.61	33.47	99.61	33.46
Peppers	7.07	1.84	99.60	33.54	99.60	33.54
Average			99.62	33.48	99.61	33.48

Table 3
Effectiveness testing results of the proposed scheme (2).

Modified images	1 Round				2 Rounds	
	Permutation only		Overall encryption		Overall encryption	
	NPCR (%)	*UACI* (%)	*NPCR* (%)	*UACI* (%)	*NPCR* (%)	*UACI* (%)
Lena-d0	16.02	3.46	99.60	33.42	99.60	33.45
Lena-d1	92.50	20.00	99.62	33.49	99.62	33.48
Lena-d2	40.10	8.65	99.62	33.42	99.60	33.51
Lena-d3	24.41	5.29	99.59	33.48	99.61	33.47
Lena-d4	49.37	10.72	99.61	33.43	99.63	33.53
Average			99.61	33.45	99.61	33.49

(1) The proposed image cipher is implemented to five standard test images and their modified versions obtained by altering the last bit of pixels at the same location, in the lower right corner. The results are listed in Table 2.
(2) Lena image and its five modified versions are encrypted by the proposed cryptosystem. These modified versions are produced by changing the last bit of the corresponding randomly selected pixels in the original Lena image, and are represented by Lena-d0, Lena-d1,. . ., Lena-d4, respectively. The simulation results are listed in Table 3.

Obviously, only one overall round is required to achieve a satisfactory and stable security performance such as $NPCR > 99.60\%$ and $UACI > 33.4\%$, no matter what testing images are used or where the difference of the plain image located, even in the circumstance that not so much diffusion effect is produced in the confusion phase. It is fully vindicated that our scheme has a steady and satisfactory security performance within the first encryption round.

4.2. Efficiency comparisons

During the previous works, K.W. Wong et al. have pointed out that the consumption time of an image cryptosystem is mainly resulted from the real number arithmetic operation and the quantization in the encryption process [8,18]. Therefore, when achieving the satisfactory security level, the average required the chaotic variables and quantization operations for encrypting one pixel can be employed to evaluate the efficiency of an image cryptosystem. Besides, in [17,37], researchers suggest that the required rounds of image-scanning (pixels traverse round) can also be used to evaluate the efficiency of an image encryption algorithm. In this subsection, four typical image encryption algorithms [5,17,19,20] are employed in comparison with the proposed scheme in terms of these three efficiency indices mentioned above.

Table 4
Efficiency analysis of the image encryption schemes to achieve a satisfactory security level.

	NPCR (%)	*UACI* (%)	Number of encryption rounds	Average required chaotic variables	Average required quantization operations	Required image-scanning rounds
Proposed	>99.61	>33.4	1	1.004*	2	2
Ref. [5]	>99.61	>33.4	3	9	3	3
Ref. [17]	>99.61	>33.4	2	6	2	2
Ref. [19]	>99.61	>33.4	1	4	2	3
Ref. [20]	>99.61	>33.4	2	7*	2	4

The indices of the proposed scheme and comparable algorithms are listed in Table 4. The lower-case superscript for the algorithm in [20] means the average number of required chaotic variables, as an extra chaotic map iteration is needed when ciphering pixels with odd gray values in that scheme. Besides, according to DSVSM, the chaotic map is iterated when necessary in our scheme. In our simulation, only 87766 iterations of Chen's chaotic map are required for ciphering the standard Lena image with 512×512 pixels. So a total number of $87766 \times 3 = 263298$ state variables have been generated during the iterations, and hence 1.004 chaotic variables are used for one pixel encryption in average.

The results demonstrate that in order to achieve a satisfactory security level such as $NPCR > 99.60\%$ and $UACI > 33.4\%$, only one overall round is required when using the proposed scheme or Fu's algorithm in [19], whereas at least two rounds encryption have to be performed when using other comparable cryptosystems. Compared with the algorithms in [5,17,19,20], our scheme has advantages in terms of at least one efficiency index, the required chaotic variables, the required quantization operations or the required rounds of image-scanning. These advantages lead to speed superiority accordingly. Besides, the comparable algorithms are designed for square images encryption. When encrypting a non-square plain image, extra pixels have to be padded to form a square image firstly, and that would downgrade the efficiency of these schemes. On the contrary, as there is no image size restriction when using our permutation and diffusion approaches, no redundant computation workload is needed when ciphering a non-square plain image. This property also leads to speed superiority of the proposed scheme.

5. Security analyzes

5.1. Key space analysis

The key space size is the total number of different keys that can be used in a cryptosystem. The key space should be sufficiently large to preclude the eavesdropping by implementing brute-force attack. In the proposed algorithm, the secret key consists of the four parameters $x_0,\ y_0,\ z_0$ and c of the Chen's chaotic system. According to the IEEE floating-point standard [38], the computational precision of the 64-bit double-precision number is about 10^{15}. Accordingly, the total number of possible values of the secret key is approximately

$$Key = 10^{15\times 4} \approx 2^{199},$$

which is large enough to resist brute-force attack.

5.2. Key sensitivity analysis

Extreme key sensitivity is an essential feature of an effective cryptosystem, and can be observed in two aspects: (i) completely different cipher images should be produced when slightly different keys are applied to encrypt the same plain image; (ii) the cipher image cannot be correctly decrypted even tiny difference exists between the encryption and decryption keys.

To evaluate the key sensitivity in the first case, the encryption is carried out 1 round at first to obtain a cipher image with coefficients ($x_0 = 3.3,\ y_0 = -2.1,\ z_0 = 6.8$ and $c = 28$). Then a slight change 10^{-14} is introduced to one of the parameters with all others remain the same, and repeat the encryption process. The corresponding cipher images and the differential images are shown in Fig. 9. The differences between the corresponding cipher images are computed and given in Table 5. The results obviously demonstrate that the cipher images exhibit no similarity one another and there is no significant correlation that could be observed from the differential images.

In addition, decryption using keys with slight difference are also performed in order to evaluate the key sensitivity of the second case. The deciphering images are shown in Fig. 10. The differences between the incorrect deciphering images to the plain image are $99.59\%,\ 99.62\%,\ 99.62\%$ and 99.63%, respectively.

The above two tests indicate that the proposed image encryption scheme is highly sensitive to the key. Even an almost perfect guess of the secret key does not reveal any valuable information about the cryptosystem.

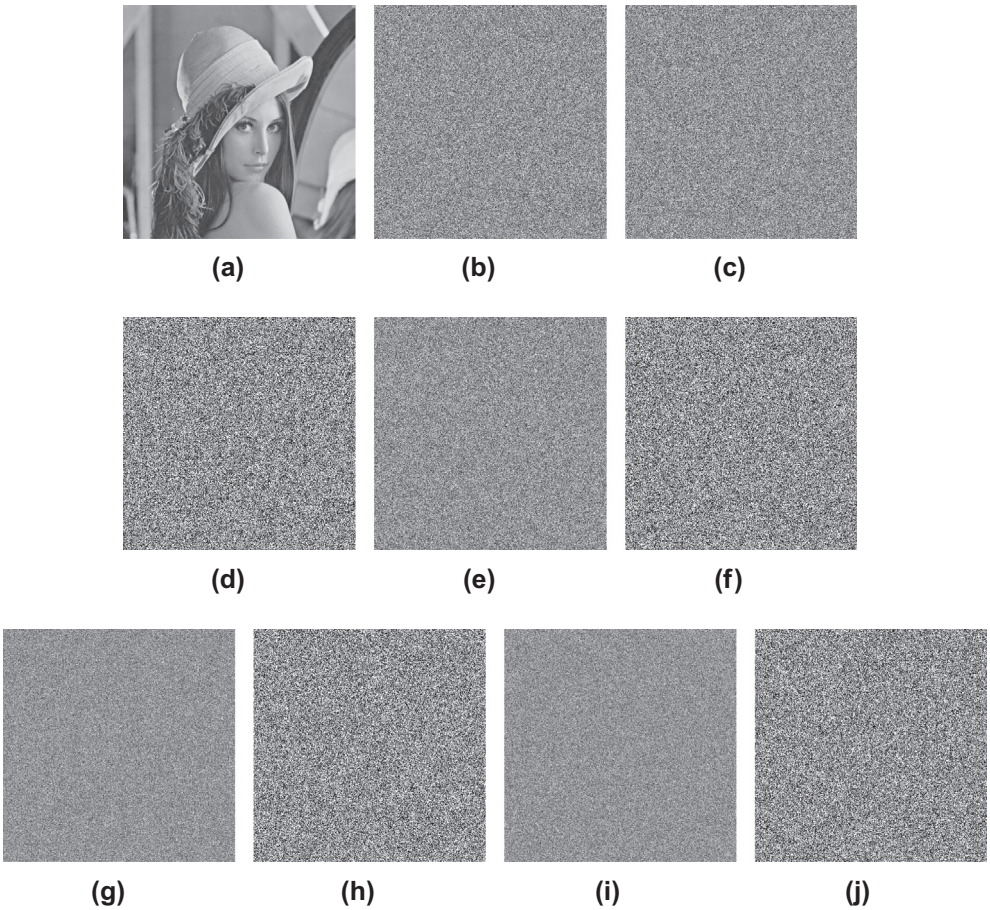

Fig. 9. Key sensitivity in the first case: (a) plain image; (b) cipher image ($x_0 = 3.3,\ y_0 = -2.1,\ z_0 = 6.8,\ c = 28$); (c) cipher image ($x_0 = 3.30000000000001,\ y_0 = -2.1,\ z_0 = 6.8,\ c = 28$); (d) differential image between (b) and (c); (e) cipher image ($x_0 = 3.3,\ y_0 = -2.10000000000001,\ z_0 = 6.8,\ c = 28$); (f) differential image between (b) and (e); (g) cipher image ($x_0 = 3.3,\ y_0 = -2.1, z_0 = 6.80000000000001,\ c = 28$); (h) differential image between (b) and (g); (i) cipher image ($x_0 = 3.3,\ y_0 = -2.1,\ z_0 = 6.8,\ c = 28.00000000000001$); (j) differential image between (b) and (i).

Table 5
Differences between cipher images produced by slightly different keys.

Figures	Encryption keys				Differences ratio between 9(b) (%)
	x_0	y_0	z_0	c	
9(c)	$3.3 + 10^{-14}$	-2.1	6.8	28	99.61
9(e)	3.3	$-2.1 + 10^{-14}$	6.8	28	99.61
9(g)	3.3	-2.1	$6.8 + 10^{-14}$	28	99.61
9(i)	3.3	-2.1	6.8	$28 + 10^{-14}$	99.59

5.3. Histogram analysis

Histogram of a digital image shows the distribution of the pixel values. The ideal histogram of an effectively ciphered image should be uniform and significantly different in comparison with that of the plain image so as to prevent the attacker from obtaining any useful statistical information. The histograms of the standard Lena image with size 512×512 and its cipher image produced by the proposed cryptosystem are shown in Fig. 11(b) and (d), respectively. It is obvious that the histogram of the encrypted image is uniformly distributed and quite different from that of the plain image, which implies that the redundancy of the plain image is successfully hidden after the encryption and consequently does not provide any clue to apply statistical attacks.

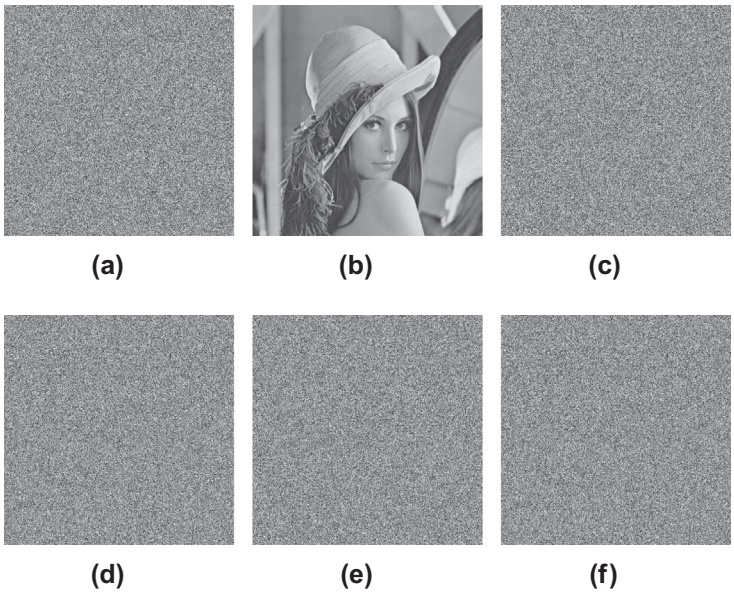

Fig. 10. Key sensitivity in the first case: (a) cipher image ($x_0 = 3.3,\ y_0 = -2.1,\ z_0 = 6.8,\ c = 28$); (b) decipher image ($x_0 = 3.3, y_0 = -2.1,\ z_0 = 6.8,\ c = 28$); (c) decipher image ($x_0 = 3.30000000000001,\ y_0 = -2.1,\ z_0 = 6.8,\ c = 28$); (d) decipher image ($x_0 = 3.3,\ y_0 = -2.10000000000001,\ z_0 = 6.8,\ c = 28$); (e) decipher image ($x_0 = 3.3,\ y_0 = -2.1,\ z_0 = 6.80000000000001,\ c = 28$); (f) decipher image ($x_0 = 3.3,\ y_0 = -2.1,\ z_0 = 6.8,\ c = 28.00000000000001$.)

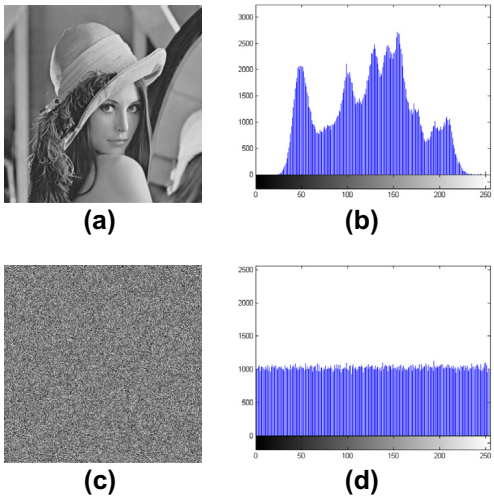

Fig. 11. Histogram analysis: (a) plain image; (b) histogram of plain image; (c) cipher image; (d) histogram of cipher image.

5.4. Correlation analysis

The correlation between adjacent pixels in the plain image is always high for a meaningful image since their pixel values are close to each other. An effective image cryptosystem should produce a cipher image with sufficiently low correlation between adjacent pixels. The following steps are performed to evaluate an image's correlation property. (1) 3000 pixels are randomly selected as samples; (2) the correlations between two adjacent pixels in horizontal, vertical and diagonal directions are calculated according to Eqs. (15)–(17), where x_i and y_i are gray-level values of the *ith* pair of the selected adjacent pixels, and N represents the total number of the samples. The correlation coefficients of adjacent pixels in the plain image and its cipher image are listed in Table 6. Moreover, the correlation of two horizontally adjacent pixels in the plain and the encrypted image are shown in Fig. 12(a) and (b), respectively. Both the calculated correlation coefficients and the figures can substantiate that the strong correlation among neighboring pixels of the plain image can be effectively de-correlated by the proposed cryptosystem.

Table 6
Correlation coefficients of adjacent pixels.

Direction	Plain image	Cipher image
Horizontal	0.9849	$-7.7491e-004$
Vertical	0.9693	0.0045
Diagonal	0.9562	0.0061

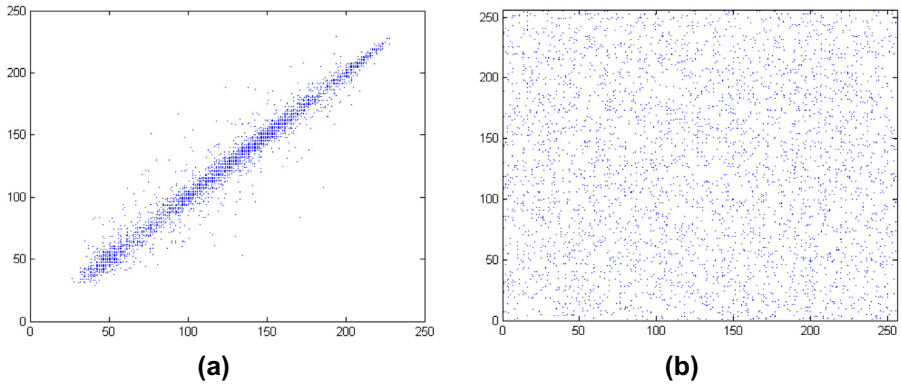

Fig. 12. Correlation of horizontal adjacent pixels: (a) plain image; (b) cipher image.

$$r_{xy} = \frac{E(x - E(x))E(y - E(y))}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{15}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{16}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2. \tag{17}$$

5.5. Information entropy

Information entropy is a mathematical property that reflects the randomness and the unpredictability of an information source that is firstly found in 1949 by Shannon [39]. The entropy $H(s)$ of a message source s is defined as

$$H(s) = -\sum_{i=0}^{2^N-1} P(s_i)\log_2 P(s_i),$$

where s is the source, N is the number of bits to represent the symbol s_i, and $P(s_i)$ is the probability of the symbol s_i. For a truly random source consists of 2^N symbols, the entropy is N. Therefore, for an effective cryptosystem, the entropy of the cipher image with 256 gray levels should ideally be 8. Otherwise, the information source is not sufficiency random and there exits a certain degree of predictability, which threatens it security.

Five 256 gray-scale test images with size 512×512 are encrypted 1 round with proposed cryptosystem and the information entropies are then calculated, as listed in Table 7. It is obvious that the entropies of the cipher images are very close to the theoretical value of 8, which means that information leakage in the encryption procedure is negligible and the proposed algorithm is secure against entropy analysis.

Table 7
Entropies of plain images and cipher images.

	Plain image	Cipher image
Lena	7.445568	7.999319
Baboon	7.357949	7.999399
Barb	7.466426	7.999321
Bridge	5.705560	7.999254
Barb	7.571478	7.999240

6. Conclusions

In the present paper, a fast chaos-based image encryption scheme with a dynamic state variables selection mechanism is proposed. By using this mechanism, the state variables generated from the three-dimensional or hyper chaotic systems are dynamically and pixel-related distributed to each pixel in both permutation and diffusion procedures. A tiny change in the plain image will bring about totally different key stream sequences even though the same secret key is used. In conjunction with pixel-swapping based confusion strategy and snake-like mode diffusion, the difference produced in state variables distribution will be transferred to the encrypting process and then spread out to the whole cipher image within the first encryption round. Simulation results and extensive security analyzes have shown that the *NPCR*, *UACI*, and information entropy of the proposed scheme are very satisfactory. All these results justify the superior security and computational efficiency of our encryption scheme.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 61271350, 61374178, 61202085), the Fundamental Research Funds for the Central Universities (No. N120504005), the Liaoning Provincial Natural Science Foundation of China (No. 201202076), the Specialized Research Fund for the Doctoral Program of Higher Education (No. 20120042120010) and the Ph.D. Start-up Foundation of Liaoning Province, China (Nos. 20111001, 20121001, 20121002).

References

[1] Furht B, Kirovski D. Chaos-based encryption for digital images and videos. Multimedia security handbook. CRC Press; 2004.
[2] Alvarez G, Li S. Some basic cryptographic requirements for chaos-based cryptosystem. Int J Bifur Chaos 2006;16(8):2129–51. http://dx.doi.org/10.1142/S0218127406015970.
[3] Dutta D, Bhattacharjee J. Period adding bifurcation in a logistic map with memory. Phys D Nonlinear Phenom 2008;237(23):3153–8. http://dx.doi.org/10.1016/j.physd.2008.05.014.
[4] Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifur Chaos 1998;8(6):1259–84. http://dx.doi.org/10.1142/S021812749800098X.
[5] Mao Y, Chen G, Lian S. A novel fast image encryption scheme based on 3D chaotic baker maps. Int J Bifur Chaos 2004;14(10):3613–24. http://dx.doi.org/10.1142/S021812740401151X.
[6] Chen G, Mao Y, Chui C. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Solitons Fractals 2004;21(3):749–61. http://dx.doi.org/10.1016/j.chaos.2003.12.022.
[7] Mirzaei O, Yaghoobi M, Irani H. A new image encryption method: parallel sub-image encryption with hyper chaos. Nonlinear Dyn 2012;67(1):557–66. http://dx.doi.org/10.1007/s11071-011-0006-6.
[8] Wong K, Kwok B, Law W. A fast image encryption scheme based on chaotic standard map. Phys Lett A 2008;372(15):2645–52. http://dx.doi.org/10.1016/j.physleta.2007.12.026.
[9] Huang C, Nien H. Multi chaotic systems based pixel shuffle for image encryption. Opt Commun 2009;282(11):2123–7. http://dx.doi.org/10.1016/j.optcom.2009.02.044.
[10] Zhu Z, Zhang W, Wong K, Yu H. A chaos-based symmetric image encryption scheme using a bit-level permutation. Inf Sci 2011;181(6):1171–86. http://dx.doi.org/10.1016/j.ins.2010.11.009.
[11] Fu C, Lin B, Miao Y, Liu X, Chen J. A novel chaos-based bit-level permutation scheme for digital image encryption. Opt Commun 2011;284(23):5415–23. http://dx.doi.org/10.1016/j.optcom.2011.08.013.
[12] Zhang W, Wong K, Yu H, Zhu Z. A symmetric color image encryption algorithm using the intrinsic features of bit distributions. Commun Nonlinear Sci Numer Simul 2013;18(3):584–600. http://dx.doi.org/10.1016/j.cnsns.2012.08.010.
[13] Wang X, Luan D. A novel image encryption algorithm using chaos and reversible cellular automata. Commun Nonlinear Sci Numer Simul 2013;18(11):3075–85. http://dx.doi.org/10.1016/j.cnsns.2013.04.008.
[14] Zhang Y, Xiao D. An image encryption scheme based on rotation matrix bit-level permutation and block diffusion. Commun Nonlinear Sci Numer Simul 2014;19(1):74–82. http://dx.doi.org/10.1016/j.cnsns.2013.06.031.
[15] Patidar V, Pareek N, Sud K. A new substitution–diffusion based image cipher using chaotic standard and logistic maps. Commun Nonlinear Sci Numer Simul 2009;14(7):3056–75. http://dx.doi.org/10.1016/j.cnsns.2008.11.005.
[16] Patidar V, Pareek N, Purohit G, Sud K. Modified substitution–diffusion image cipher using chaotic standard and logistic maps. Commun Nonlinear Sci Numer Simul 2010;15(10):2755–65. http://dx.doi.org/10.1016/j.cnsns.2009.11.010.
[17] Zhang W, Wong K, Yu H, Zhu Z. An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion. Commun Nonlinear Sci Numer Simul 2013;18(8):2066–80. http://dx.doi.org/10.1016/j.cnsns.2012.12.012.
[18] Wong K, Kwok B, Yuen C. An efficient diffusion approach for chaos-based image encryption. Chaos Solitons Fractals 2009;41(5):2652–63. http://dx.doi.org/10.1016/j.chaos.2008.09.047.
[19] Fu C, Chen J, Zou H, Meng W, Zhan Y, Yu Y. A chaos-based digital image encryption scheme with an improved diffusion strategy. Opt Express 2012;20(3):2363–78. doi:10.1364/OE.20.002363.
[20] Wang Y, Wong K, Liao X, Xiang T, Chen G. A chaos-based image encryption algorithm with variable control parameters. Chaos Solitons Fractals 2009;41(4):1773–83. http://dx.doi.org/10.1016/j.chaos.2008.07.031.
[21] Chen J, Zhu Z, Fu C, Yu H. An improved permutation–diffusion type image cipher with a chaotic orbit perturbing mechanism. Opt Express 2013;21(23):27873–90. http://dx.doi.org/10.1364/OE.21.027873.
[22] Zhang Y, Xiao D, Shu Y, Li J. A novel image encryption scheme based on a linear hyperbolic chaotic system of partial differential equations. Signal Process-Image 2013;28(3):292–300. http://dx.doi.org/10.1016/j.image.2012.12.009.
[23] Zhu C. A novel image encryption scheme based on improved hyperchaotic sequences. Opt Commun 2012;285(1):29–37. http://dx.doi.org/10.1016/j.optcom.2011.08.079.
[24] Pisarchik A, Zanin M. Image encryption with chaotically coupled chaotic maps. Phys D Nonlinear Phenom 2008;237(20):2638–48. http://dx.doi.org/10.1016/j.physd.2008.03.049.
[25] Gao T, Chen Z. A new image encryption algorithm based on hyper-chaos. Phys Lett A 2008;372(4):394–400. http://dx.doi.org/10.1016/j.physleta.2007.07.040.
[26] Guan Z, Huang F, Guan W. Chaos-based image encryption algorithm. Phys Lett A 2005;346(1–3):153–7. http://dx.doi.org/10.1016/j.physleta.2005.08.006.

[27] SaberiKamarposhti M, Mohammad D, Rahim M, Yaghobi M. Using 3-cell chaotic map for image encryption based on biological operations. Nonlinear Dyn 2014;75(3):407–16. http://dx.doi.org/10.1007/s11071-013-0819-6.
[28] Pareek N, Patidar V, Sud K. Image encryption using chaotic logistic map. Image Vis Comput 2006;24(9):926–34. http://dx.doi.org/10.1016/j.imavis.2006.02.021.
[29] Solak E, Cokal C, Yildiz O, Biyikoglu T. Cryptanalysis of Fridrich's chaotic image encryption. Int J Bifur Chaos 2010;20(5):1405–13. http://dx.doi.org/10.1142/S0218127410026563.
[30] Rhouma R, Solak E, Belghith S. Cryptanalysis of a new substitutiondiffusion based image cipher. Commun Nonlinear Sci Numer Simul 2010;15(7):1887–92. http://dx.doi.org/10.1016/j.cnsns.2009.07.007.
[31] Li C, Li S, Lo K. Breaking a modified substitutiondiffusion image cipher based on chaotic standard and logistic maps. Commun Nonlinear Sci Numer Simul 2011;16(2):837–43. http://dx.doi.org/10.1016/j.cnsns.2010.05.008.
[32] Li C, Liu Y, Xie T, Chen M. Breaking a novel image encryption scheme based on improved hyperchaotic sequences. Nonlinear Dyn 2013;73(3):2083–9. http://dx.doi.org/10.1007/s11071-013-0924-6.
[33] Li C, Li MAS, Nunez J, Alvarez GCG. On the security defects of an image encryption scheme. Image Vis Comput 2009;27(9):1371–82. http://dx.doi.org/10.1016/j.imavis.2008.12.008.
[34] Arroyo D, Li S, Amigo J, Alvarez G, Rhouma R. Comment on image encryption with chaotically coupled chaotic maps. Phys D Nonlinear Phenom 2010;239(12):1002–6. http://dx.doi.org/10.1016/j.physd.2010.02.010.
[35] Cokal C, Solak E. Cryptanalysis of a chaos-based image encryption algorithm. Phys Lett A 2009;373(15):1357–60. http://dx.doi.org/10.1016/j.physleta.2009.02.030.
[36] Wang K, Pei W, Zou L, Song A, He Z. On the security of 3d cat map based symmetric image encryption scheme. Phys Lett A 2005;343(6):432–9. http://dx.doi.org/10.1016/j.physleta.2005.05.040.
[37] Wang Y, Wong K, Liao X, Chen G. A new chaos-based fast image encryption algorithm. Appl Soft Comput 2011;11(1):514–22. http://dx.doi.org/10.1016/j.asoc.2009.12.011.
[38] IEEE Computer Society, IEEE standard for binary floating-point arithmetic. ANSI/IEEE std. 1985;754–1985.
[39] Shannon C. Communication theory of secrecy systems. Bell Syst Tech J 1949;18(4):656–715. http://dx.doi.org/10.1016/j.asoc.2009.12.011.