Commun Nonlinear Sci Numer Simulat 18 (2013) 2066–2080

Contents lists available at SciVerse ScienceDirect

Commun Nonlinear Sci Numer Simulat

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

An image encryption scheme using reverse 2-dimensional chaotic map and dependent diffusion

Wei Zhang [a], Kwok-wo Wong [b], Hai Yu [c], Zhi-liang Zhu [c,*]

[a] *College of Information Science and Engineering, Northeastern University, No. 11, Lane 3, WenHua Road, Shenyang, Liaoning, China*
[b] *Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong*
[c] *Software College, Northeastern University, No. 11, Lane 3, WenHua Road, Shenyang, Liaoning, China*

ARTICLE INFO

Article history:
Received 19 June 2012
Received in revised form 16 September 2012
Accepted 7 December 2012
Available online 4 January 2013

Keywords:
Dependent diffusion
Reverse cat map
Chaos
Cryptography
Image encryption

ABSTRACT

In recent years, a variety of chaos-based image cryptosystems have been studied. Most of them adopt the traditional confusion–diffusion architecture, which is considered insecure upon chosen/known plain-image attacks. In this paper, a nonlinear traverse on the plain-image using dependent diffusion and reverse cat map is proposed to replace the traditional linear traverse performed in the confusion phase. Two cryptosystems are designed and are implemented by software means. Simulation results and numerical analysis justify their high efficiency and sufficient strength.

1. Introduction

Due to the characteristics of easy-understanding and attractive presentation, multimedia contents such as video, image and audio, have been widely transmitted in the ever-growing Internet and mobile communications. The privacy of certain multimedia files such as personal videos and images needs to be protected when these files are sent via a public network. However, traditional cryptosystems such as DES and AES are found unfit for multimedia data [1].

The fundamental characteristics of chaotic systems, such as ergodicity, sensitivity to initial condition and control parameters, have attracted researcher's attention since such features can be considered analogous to the desired cryptographic properties. Chaos-based encryption algorithms [1–8,10–22,25] have been extensively studied due to their superior properties in security and complexity. In 1998, Fridrich firstly proposed a chaos-based image cryptosystem composing of permutation and diffusion [20]. Under this structure, the permutation of plain-image pixels is governed by a 2-D chaotic map, such as standard map, baker map and cat map. In the diffusion phase, a 1-D chaotic map is usually employed, as Fig. 1 shows. The Fridrich architecture has become the most popular structure adopted in many chaos-based image encryption algorithms subsequently proposed.

There are a variety of approaches for confusion and diffusion. In [2], Lian et al. proposed an encryption algorithm, in which a modified standard map is employed in the confusion step while a logistic map is adopted for diffusion. The fixed-point problem of the standard map is solved by shifting the origin to a randomly-selected point. In [3], a block cipher based on dynamic S-boxes was studied. A tent map is chosen to generate the S-box required in the permutation phase, and a left-cyclic-shift operation is used for diffusion. In [7], Wong et al. improved Lian et al.'s algorithm by introducing an "add-and-shift"

* Corresponding author.
E-mail address: swc.zhuzhiliang@gmail.com (Z.-l. Zhu).

1007-5704/$ - see front matter Crown Copyright © 2012 Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cnsns.2012.12.012

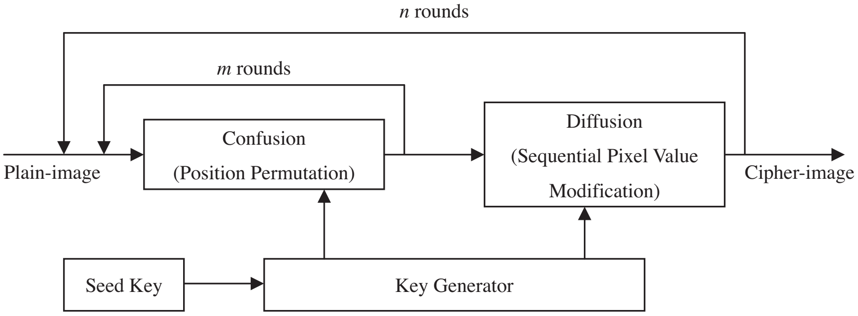


Fig. 1. The Fridrich image encryption architecture.

operation in the confusion stage. Since the execution time of a round of diffusion is much longer than that required by confusion, the operation efficiency is improved by introducing certain diffusion effect in the confusion phase. Patidar et al. [13] proposed a color image encryption scheme using two rounds of confusion and two rounds of diffusion. In the diffusion phase, the vertical and horizontal diffusions are performed using standard map and logistic map, respectively. In [28], a novel cryptosystem based on the Fridrich structure is proposed, in which the pixel level permutation is replaced by a bit level permutation. As a result, a permutation at bit level not only changes the position of a pixel but also alters its value.

In [4], Wang et al. pointed out that under the confusion–diffusion architecture with fixed parameters, the two processes will become independent if the plain-image is a homogeneous one with identical pixels. In this case, the confusion effect is removed and the security of the whole cryptosystem only relies on diffusion. Therefore, it can be concluded that the Fridrich confusion–diffusion structure can be attacked by the following steps: (1) a homogeneous image with identical pixels is chosen to eliminate the effect of confusion; (2) the key-stream of the diffusion phase is obtained via known- or chosen-plaintext attacks; (3) the remaining cipher-image can be considered as the output of a kind of permutation-only cipher, which has been proven insecure and can be easily cryptanalyzed by known- or chosen-plaintext attacks [9,23,24,26].

In this paper, two cryptosystems are designed to avoid the flaw of the Fridrich architecture. The first one makes use of dependent diffusion and the reverse cat map. It can be considered as an improved version of the Fridrich framework since there is no clear separation between the confusion and the diffusion phases. Therefore, the drawback of the conventional architecture is overcome. In the Fridrich architecture, all the pixels are permuted before the pixel values are diffused. This means that the basic unit of the confusion and diffusion operations is the whole image. However, in our scheme, the basic unit is a pixel. Once the new location of a pixel is calculated, we diffuse the pixel immediately rather than calculate the next pixel's location. The value of the ciphered pixel influences the next pixel's confusion and diffusion operations. The confusion effect governs the diffusion process at the pixel level. As a result, only one traverse of all the pixels is needed instead of two traverse rounds composed of one confusion and one diffusion as required in the traditional structure. Thus, the proposed scheme is more efficient.

In the second proposed cryptosystem, a modified mapping based on a 2-D chaotic map is adopted in the confusion phase and a simple diffusion is performed in the same phase. In the conventional confusion process, a mapping from an ordinary position to a pseudorandom position is defined. Take the cat map as an example, the input sequence to the map is the regular pixel position (usually from the upper-left corner to the lower-right corner) while the output sequence is considered as pseudorandom. Here we propose a new kind of mapping which maps a pseudorandom position in the plain-image to another pseudorandom position in the cipher-image. With the help of the new mapping operation and simple diffusion in the confusion phase, the confusion and diffusion effects cannot be separated using a plain-image with identical pixels. Thus the cryptanalysis for permutation-only ciphers become ineffective.

The paper is organized as follows. In Section 2, dependent diffusion and the reverse cat map are introduced using a simple example. The proposed cryptosystems are described in Section 3. Simulation results and performance analysis are reported in Section 4. A conclusion is drawn in the last section.

2. Dependent diffusion and the reverse use of 2-dimensional chaotic maps

2.1. Dependent diffusion

In traditional confusion–diffusion type image cryptosystems, these two processes are operated independently. Firstly, the new position of each pixel is calculated. Then the pixel values are modified in the diffusion phase. Under this structure, two rounds of traverse over all the pixels are needed.

To reduce the execution time, we investigate if both confusion and diffusion can be performed by only one traverse of the pixels. In the proposed scheme, dependent diffusion and pixel relocation using the reverse cat map (which is described in

Section 2.2) are employed. When the location of a pixel is calculated, its value is also obtained by a single dependent diffusion.

Eqs. (1) and (2) illustrate the processes of dependent diffusion. A cat map and a logistic map are employed, and only one traverse round of all the pixels is required. For each pixel, Eq. (1) is firstly used to calculate its new position. Then Eq. (2) is employed to diffuse that pixel. However, in the Fridrich architecture, the positions of all the pixels are calculated before the diffusion operation starts.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N, \tag{1}$$

$$ciphered(x', y') = arr(x, y) \oplus \{[\alpha \times (t/1000) \times [1-(t/1000)] \times 1000] \bmod 256\}. \tag{2}$$

In Eq. (1), (x, y) is the original position of the plain-image pixel while (x', y') is the pseudorandom position governed by the cat map, p and q are the cat map parameters, N denotes the width or height of the square image. In Eq. (2), $arr(x, y)$ and $ciphered(x', y')$ are the pixel values of the plain-image and the cipher-image, respectively. Moreover, t is the value of the previously-processed pixel, and α is the parameter of the logistic map.

If the diffusion process is not conducted in a sequential manner or the value of the pixel being processed is not influenced by the previous one, the cryptosystem will become insensitive to a slight modification of the plain-image. In dependent diffusion using an ordinary cat map, the pixel being processed is not adjacent to the previously-processed pixel in the ciphered image, as Fig. 2 illustrates. A variable named "t" is used to store the value of the previously-processed pixel.

To clearly illustrate the concept of dependent diffusion using an ordinary cat map, two pixels are first chosen, as shown in Fig. 2. $arr(x, y)$ and $arr(x, y+1)$ are two adjacent pixels in the plain-image while $ciphered(x', y')$ and $ciphered(x^*, (y+1)^*)$ are the two corresponding non-adjacent encrypted pixels in the cipher-image. Firstly, the new position (x', y') of $arr(x, y)$ is calculated. After that, the diffusion process on this pixel is performed, and the new pixel value $ciphered(x', y')$ is obtained. The temporary variable t is set to $ciphered(x', y')$. In the second step, the new location $(x^*, (y+1)^*)$ of $arr(x, y+1)$ is calculated. The diffusion process, which is influenced by the previously-processed pixel t, is performed.

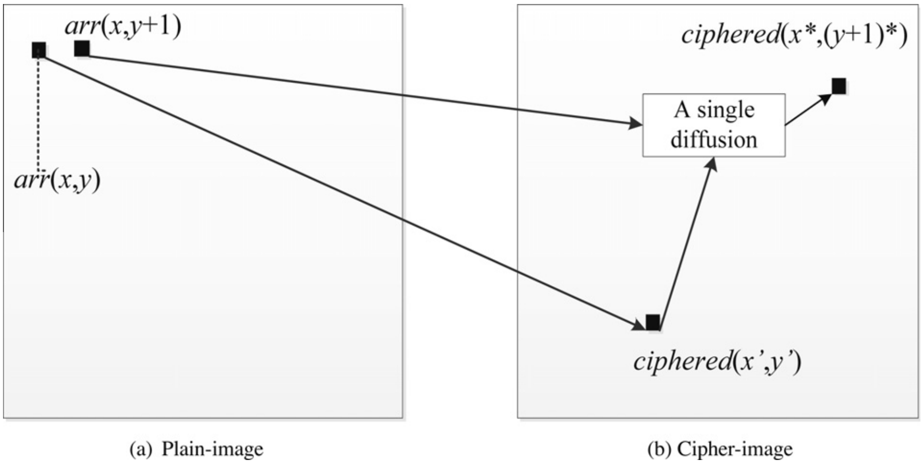
In the dependent diffusion architecture, the parameters and the initial values of confusion (or the effect of confusion) govern the diffusion order among the pixels. The pixel value of the cipher-image is influenced by both the secret key of the diffusion phase and the previously-processed pixel value. The relationship between the current and the previously-processed pixels is governed by the temporary variable t to increase the sensitivity to any modifications in the plain-image.

A more general situation of dependent diffusion is shown in Fig. 3. The relationship among the cipher-image pixels is clear. For example, Fig. 3(b) illustrates that the value of pixel 2 is influenced by that of pixel 1 while the value of pixel 3 is in turn influenced by that of pixel 2, etc.

The second part of the right-hand-side of Eq. (2) is extracted as Eq. (3).

$$\Phi(s) = [\alpha \times (s/1000) \times [1-(s/1000)] \times 1000] \bmod 256. \tag{3}$$

As the ranges of $\Phi(s)$ and s are both [0,255], a look-up table is used to reduce the execution time. We compare the operating efficiency of the two structures for achieving the same encryption effect that each plain-image pixel is relocated and diffused once, respectively. A confusion and a diffusion rounds are performed in the Fridrich architecture while one round of dependent diffusion is performed in the new structure. In the latter case, the initial condition of t is given by $t = \{[\alpha \times (kd/1000)] \times (1 - kd/1000)] \times 1000\} \bmod 256$, α is set to 4. (kd, p_d, q_d) is the encryption key, where p_d and q_d are



(a) Plain-image (b) Cipher-image

Fig. 2. Dependent diffusion.

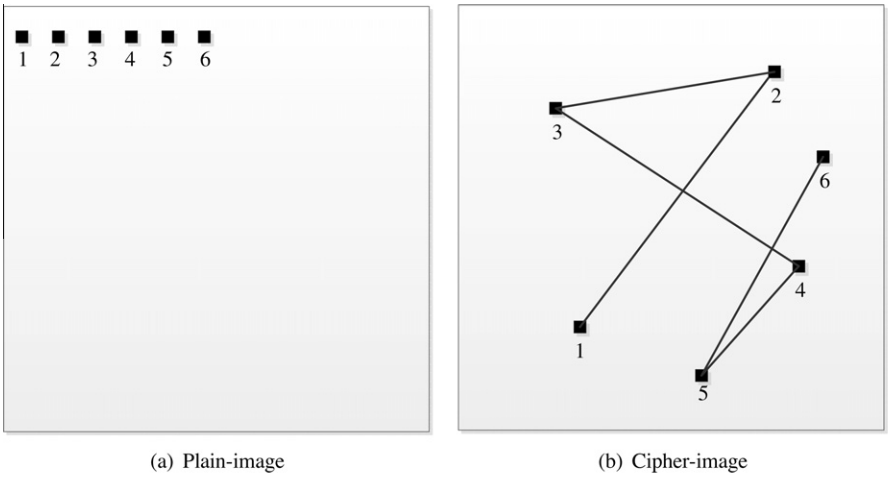


Fig. 3. A more general situation of dependent diffusion.

the cat map parameters. In the Fridrich structure, (p_f, q_f) are the cat map parameters while the *initial_value* is defined in [20], which is the key of the diffusion operation. More information can be found in [20]. Table 1 shows the execution times of the two structures using different test images and parameters.

As Table 1 shows, the execution time of dependent diffusion is much shorter than that of the Fridrich structure. Besides computation efficiency, the confusion effect of the dependent diffusion structure employing the reverse cat map will not be removed by a homogeneous image. An experiment has been performed and is described in Section 2.3.

2.2. The reverse use of 2-dimensional chaotic map

A two-dimensional chaotic map is usually employed in the confusion phase of most classical image cryptosystems. Take the cat map given by Eq. (1) as an example, almost all the pixels are relocated by the map in the confusion phase. It defines a mapping from the original position (usually from upper-left corner to lower-right corner) to a pseudorandom location. When the confusion phase starts, the pixels at different positions of the plain-image have different processing orders. For example, $arr(0,0)$, which is the upper-left corner pixel, is always the first one to be processed. On the other hand, $arr(511,511)$, which locates at the lower-right corner, is the last one in the processing sequence.

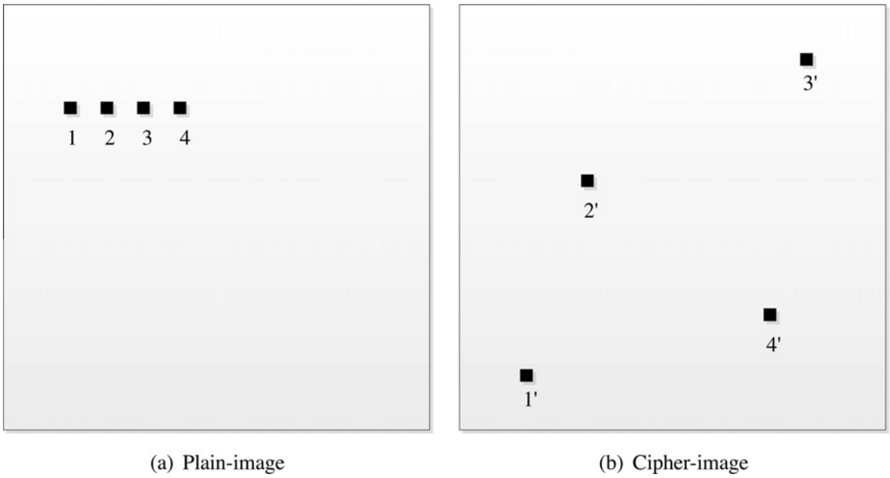
The processing order of a pixel in the plain-image is significant. Under the assumption that the output sequence of the cat map is pseudorandom, we can investigate if it is possible to use the map reversely to visit the plain-image pixels in a non-linear manner rather than a regular way. If the cat map defines a mapping from the original position to a pseudorandom one, the reverse cat map should give the mapping from a pseudorandom position to a regular one. Almost all the pixels have the same probability being relocated to any positions in the processing sequence. This means that the last pixel $arr(511,511)$ will not always be placed at the end of the processing sequence. The reverse cat map along with the dependent diffusion is formulated in Eqs. (1) and (4).

$$ciphered(x,y) = arr(x',y') \oplus \{[\alpha \times (t/1000) \times [1-(t/1000)] \times 1000] \bmod 256\}. \tag{4}$$

The difference between Eqs. (2) and (4) are the pixel positions: one is $ciphered(x',y')$ and $arr(x,y)$ but the other is $ciphered(x,y)$ and $arr(x',y')$. For example, if $arr(511,511)$ is slightly modified, the pixel will be processed in an earlier stage, not the last. The subsequent pixels will be influenced by this tiny modification in an earlier stage.

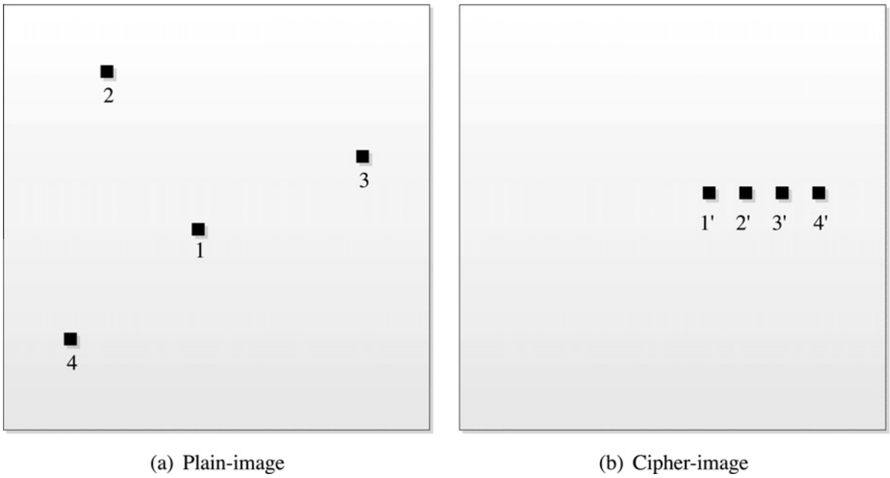
Table 1
Execution time of dependent diffusion and Fridrich structure.

Test image	Size	Dependent diffusion structure		Fridrich structure[20]	
		Parameters (kd, p_d, q_d)	Execution time (ms)	Parameters $(initial\_value, p_f, q_f)$	Execution time (ms)
Lena.bmp	256 × 256	(0.112,100,33)	0.9	(70,100,33)	1.3
Goldhill.bmp	256 × 256	(0.228,50,96)	0.9	(120,50,96)	1.4
Cameraman.bmp	256 × 256	(0.009,60,58)	0.8	(25,60,58)	1.3
Peppers.bmp	256 × 256	(0.478,12,24)	0.9	(103,12,24)	1.4
Barb.bmp	512 × 512	(0.283,124,212)	4.6	(79,124,212)	6.3
Lena.bmp	512 × 512	(0.334,223,144)	4.0	(210,223,144)	6.3
Goldhill.bmp	512 × 512	(0.872,199,68)	4.0	(32,199,68)	6.2
Boat.bmp	512 × 512	(0.526,78,83)	4.0	(61,72,83)	7.8



(a) Plain-image (b) Cipher-image

Fig. 4. Confusion effect by an ordinary 2-dimensional chaotic map.



(a) Plain-image (b) Cipher-image

Fig. 5. Confusion effect by the reverse use of 2-dimensional chaotic map.

The difference in the confusion effect between the ordinary 2-dimensional chaotic map and the reverse use of the map is depicted in Figs. 4 and 5. The pixels 1, 2, 3 and 4 in Figs. 4(a) and 5(a) are plain-image pixels while pixels $1'$, $2'$, $3'$ and $4'$ in Figs. 4(b) and 5(b) represent the permuted pixels in the cipher-image. Fig. 4 illustrates that by using an ordinary 2-dimensional chaotic map, the adjacent pixels are visited in a sequential manner, and are mapped to pseudorandom positions. However, in the reverse use of the 2-dimensional chaotic map, the pixels in the plain-image are visited in a pseudorandom manner, as governed by the output sequence of the map which is determined by both the parameters and the initial value. They are then mapped to ordinary positions in the cipher-image. Besides the cat map, which is used as an example, the standard map, henon map or baker map can also be employed in our dependent diffusion structure.

Based on the above analyses, we can investigate if a new kind of mapping, from a pseudorandom position in the plain-image to another pseudorandom position in the cipher-image, can be defined by combining the ordinary and the reverse use of 2-dimensional chaotic maps. This will be studied in detail in Section 3.2.

2.3. Confusion effect of plain-image with identical pixel value

In traditional confusion–diffusion cryptosystems, the confusion effect can be removed by a plain-image with identical pixels, which leads to the security problem mentioned in Section 1. However, in the proposed dependent diffusion architecture, the separation of confusion and diffusion is not as clear as it is in the Fridrich structure since the two operations are fully mixed in the encryption process. Furthermore, the confusion coefficients determine the diffusion order. These lead to the result that a homogeneous image cannot compromise the confusion phase of the cryptosystem.

If the confusion effect can be eliminated by a homogeneous image, the cipher-images obtained by the original and the modified confusion coefficients will be the same. Otherwise, the two cipher-images will be different because the modified coefficients influence the final cipher-image. To clearly illustrate this property, a simple experiment is performed using a homogeneous plain-image with all pixel values set to 170. Two different cryptosystems, one with the Fridrich structure whereas the other is composed of dependent diffusion using the reverse cat map, are tested by the following steps:

(1) The homogeneous plain-image is encrypted by the original confusion coefficients and a cipher-image *ci*1 is obtained.
(2) The same plain-image is encrypted by the modified confusion coefficients and cipher-image *ci*2 is obtained.
(3) Compare *ci*1 and *ci*2 to find their differences, if any.

In our simulations, four confusion and one diffusion rounds are chosen in the Fridrich structure while two rounds of dependent diffusion are adopted in the structure governed by Eqs. (1) and (4). The ordinary cat map is employed in the Fridrich structure but the reverse cat map is used in dependent diffusion structure. The original coefficients are both (235, 509) and the modified coefficients are obtained by adding 1 to each parameter, i.e., (236, 510).

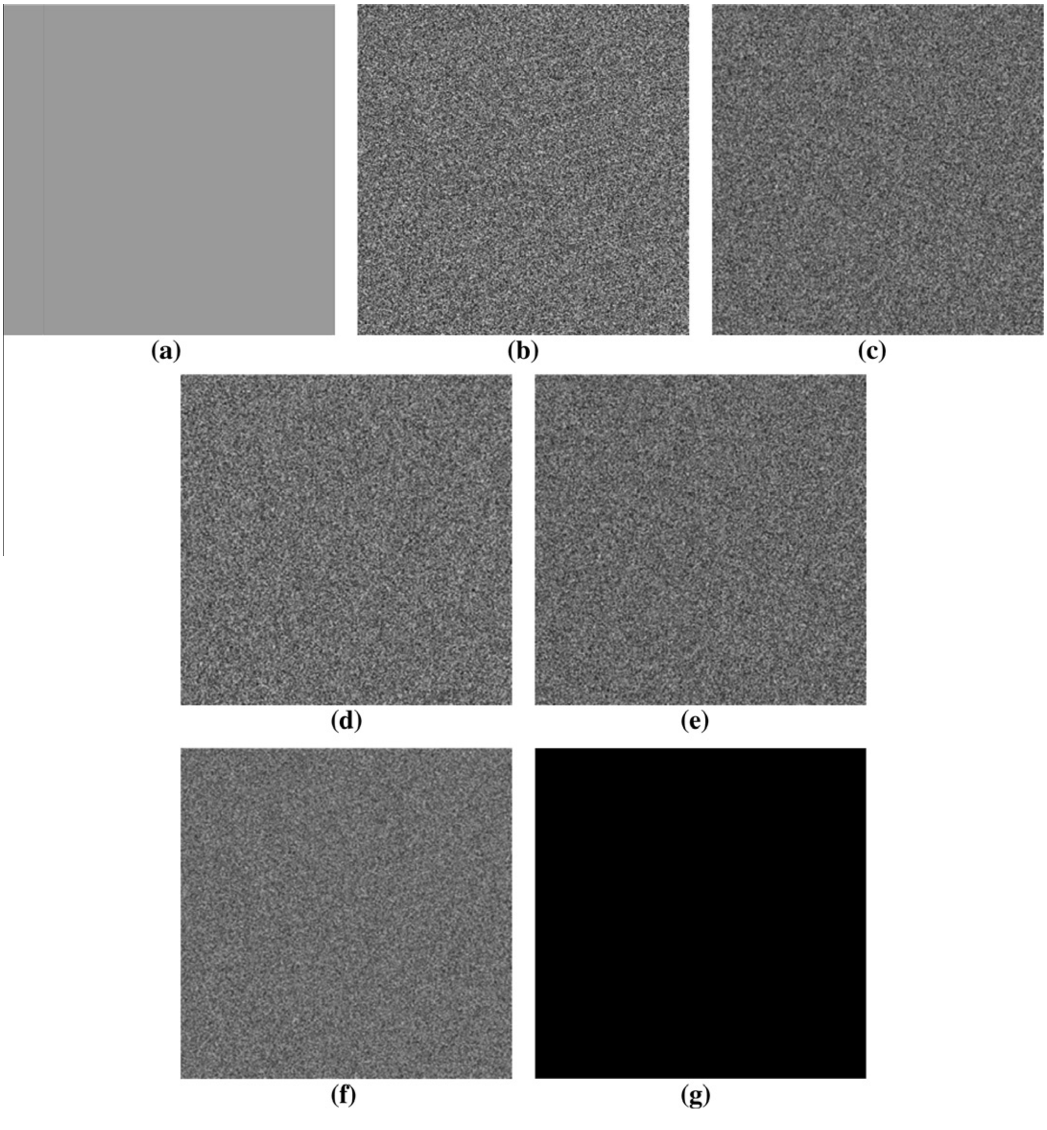


Fig. 6. (a) The plain-image with identical pixel value; (b) the cipher-image obtained by dependent diffusion structure with coefficients (235, 509); (c) the cipher-image obtained by the Fridrich structure with original coefficients (235, 509); (d) the cipher-image obtained by dependent diffusion structure with modified coefficients (236, 510); (e) the cipher-image obtained by the traditional structure with modified coefficients (236, 510); (f) the difference between (b) and (d); (g) the difference between (c) and (e).

The simulation results are shown in Fig. 6. Fig. 6(a) is the plain-image with identical pixel value of 170. Fig. 6(b) and (d) are the cipher-images obtained by the dependent diffusion structure using the original and the modified confusion coefficients, respectively. Fig. 6(c) and (e) are the cipher-images obtained by the Fridrich structure using the original and the modified confusion coefficients, respectively. Fig. 6(f) is the difference image between Fig. 6(b) and (d) whereas Fig. 6(g) is the difference image of Fig. 6(c) and (e). Since the range of the pixel values in the difference image is $[-255,255]$, which exceeds the range represented by 8 bits, it cannot be shown properly in a bitmap file. To show the difference between the two cipher-images graphically, a scaling is applied to map the difference image values from the range $[-255,255]$ to $[0,255]$, as Fig. 6(f) and (g) show.

The difference between two images obtained by dependent diffusion structure is shown in Fig. 6(f). It justifies that the confusion effect is not removed by a plain-image with identical pixels. The simulation results show that a slight modification of the confusion coefficients leads to 99.58% different pixels in the two cipher-images. This implies that the confusion effect influences the final cipher-image significantly even though all the plain-image pixels have the same value. On the other hand, the pixel values of Fig. 6(g) are all zero, which reveals that there is no difference between Fig. 6(c) and (e) although the confusion coefficients have been changed. Thus, we can conclude that the dependent diffusion structure can resist this kind of attack.

3. The proposed schemes

3.1. Algorithm 1

In Algorithm 1, the reverse cat map and dependent diffusion are employed. Its structure is shown in Fig. 7.

The dependent diffusion is governed by the reverse cat map and the logistic map. The ordinary cat map is defined by Eq. (1) whereas the logistic map is defined by Eq. (5).

$$f(x_n) = \alpha x_{n-1}(1 - x_{n-1}). \tag{5}$$

In Eq. (5), α is the logistic map coefficient. The output sequence is chaotic when $\alpha \in [3.57, 4]$. Two sequences SQ_1 and SQ_2 are generated by iterating Eq. (5) with two secret keys $conf\_key_1 = 0.12345678912345$ and $conf\_key_2 = 0.67856746347633$. α is set to 3.99999.

Denote p_i and q_i as the coefficients of the reverse cat map in the ith round of dependent diffusion. Moreover, r_x^j, r_y^j are randomly-selected pixel positions in the jth encryption round. These four parameters are generated according to Eqs. (6)–(9), respectively.

$$p_i = (SQ_1(2000 + i) \times 10^9) \bmod 512, \tag{6}$$

$$q_i = (SQ_2(2000 + i) \times 10^9) \bmod 512, \tag{7}$$

$$r_x^j = (SQ_1(2000 + 100 + j) \times 10^9) \bmod 512, \tag{8}$$

$$r_y^j = (SQ_2(2000 + 100 + j) \times 10^9) \bmod 512. \tag{9}$$

The process of dependent diffusion is governed by Eq. (10). As Eq. (3) indicates, a look-up table is used to reduce the execution time.

$$\begin{cases} \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & p_i \\ q_i & p_i q_i + 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N, \\ ciph(x,y) = arr(x',y') \oplus \{[\alpha \times (t/1000) \times [1 - (t/1000)] \times 1000] \bmod 256\}, \\ t = ciph(x,y). \end{cases} \tag{10}$$

In Eq. (10), t is a temporary variable storing the value of the previous ciphered pixel. Its initial value is defined by $t = [4 \times key\_d \times (1 - key\_d) \times 1000] \bmod 256$,N is the width (or height) of the test image, $x \in [0, N-1]$, $y \in [0, N-1]$. key_d is set to 0.33456434300001. $arr(x',y')$ denotes the plain-image pixel at the random position (x',y'). $ciph(x,y)$ represents

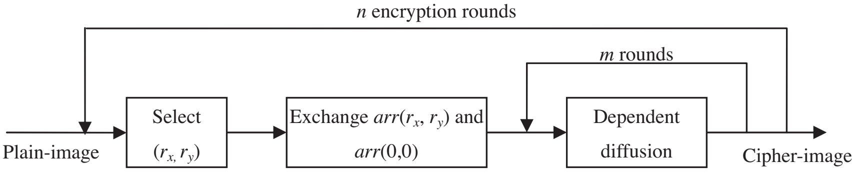


Fig. 7. Architecture of the proposed Algorithm 1.

the ciphered pixel locating at position (x, y), (p_i, q_i) is the coefficient pair of the reverse cat map in the ith round of dependent diffusion, $i = 1, 2, \ldots, m$. In Fig. 7, the number of dependent diffusion rounds m is selected as 2.

The operation procedures of the proposed scheme are described as follows:

Step 1: A random pair (r_x^j, r_y^j) is generated from Eqs. (8) and (9).
Step 2: The two pixels, $arr(0, 0)$ and $arr(r_x^j, r_y^j)$, are exchanged to solve the fixed point problem of the cat map.
Step 3: Perform m rounds of dependent diffusion.
In the ith round $(i = 1, 2, \ldots, m)$,

(1) (p_i, q_i) is given by Eqs. (6) and (7).
(2) A random position (x', y') is generated according to the reverse cat map with parameters (p_i, q_i). The value of $arr(x', y')$ in the plain-image is modified by a single diffusion operation. The value of each ciphered pixel is influenced by two factors: $arr(x', y')$ and the value of the previous ciphered pixels.
(3) The encrypted pixel is placed at the regular position (x, y) of the cipher-image.
(4) Go to (2) until all the pixels in the plain-image have been encrypted.

Step 4: Go to Step 3 to perform the $(i + 1)$th round of dependent diffusion until $i = m$.

In the proposed scheme, two rounds of dependent diffusion with different coefficients and initial values form an encryption round. Simulation results show that the NPCR and UACI values can reach 99.6% and 33.4%, respectively, in only two encryption rounds. The detailed simulations and performance analyses of this scheme are reported in Section 4.

3.2. Algorithm 2

The ordinary cat map defines a mapping from a regular position to a pseudorandom location whereas the reverse cat map gives a mapping from a pseudorandom position to a regular one. It is worthy to investigate if a new kind of mapping from a pseudorandom position to another pseudorandom position can be obtained by the combination of ordinary and reverse cat maps.

In the ordinary confusion phase using a cat map defined by Eq. (1), the input pair (x_i, y_i) usually indicates the pixel's position in the plain-image. The processing order is usually from the upper-left corner to the lower-right corner. Meanwhile, the output position (r_{xi}, r_{yi}) of the cat map is considered as pseudorandom. The confusion effect of the cat map is achieved by mapping each pixel from position (x_i, y_i) in the plain-image to the corresponding pseudorandom position (r_{xi}, r_{yi}) in the cipher-image.

The situation of reverse cat map is just the opposite. The confusion effect is achieved by mapping each pixel located at (r_{xi}, r_{yi}) in the plain-image to (x_i, y_i) in the cipher-image. Owning to the different characteristics of the two mapping operations, a new kind of confusion mapping a pseudorandom position to another pseudorandom position can be obtained by combining ordinary and reverse cat maps, as illustrated in Fig. 8.

Fig. 8(a) is the plain-image and Fig. 8(c) is the cipher-image after confusion. Fig. 8(b) is the transition required in the permutation stage. According to Eq. (11), (p_1, q_1) is the parameter pair of the reverse cat map in the first confusion step (from Fig. 8(a) to (b)) and (p_2, q_2) denotes the parameter of the ordinary cat map in the second confusion step (from Fig. 8(b) to (c)). In this new confusion process, four parameters, rather than two, govern the mapping from a pseudorandom position to another pseudorandom position.

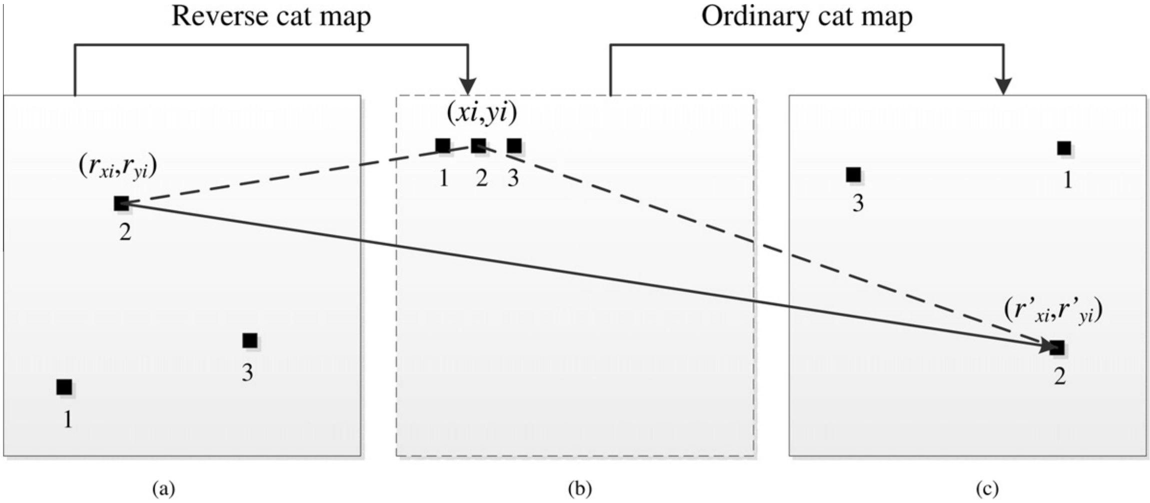


Fig. 8. A new confusion mapping from a pseudorandom position to another pseudorandom position.

Firstly, a random position (r_{xi}, r_{yi}) is generated using the reverse cat map. The plain-image pixel located at this position points to (x_i, y_i), which is a temporary position of the pixel, as illustrated in Fig. 8(a) and (b). Secondly, the cat map is employed to map the pixel locating at (x_i, y_i) to another random position (r'_{xi}, r'_{yi}) (Fig. 8(b) and (c)). Take pixel 2 in Fig. 8 as an example, the start position (Fig. 8(a)) and the final position (Fig. 8(c)) are governed by (p_1, q_1) and (p_2, q_2). Note that Fig. 8(b) does not really exist during the confusion phase, since it is only a transition state.

The same confusion effect can also be achieved by the combination of two different 2-D chaotic maps. For example, the combination of reverse standard map and the ordinary cat map, or the combination of reverse cat map and the ordinary baker map, etc.

In Algorithm 2, the Fridrich structure is employed with the following modifications to avoid the flaws mentioned in the Introduction: (1) the confusion stage is composed of a combination of reverse and ordinary cat maps; (2) pixel value modification is performed between two confusion steps. The confusion phase is governed by Eq. (11), in which *rand*1 is a random number array with 256 elements, generated by the logistic map with the initial value and coefficient 0.72345678912345 and 3.99999, respectively. If the output number is the same as any of the previous ones when *rand*1 is being calculated, that number is rejected and the algorithm proceeds to the next computation round to guarantee that there are no identical numbers in *rand*1. The first 2000 numbers generated by the logistic map are discarded. *temp*1 is a temporary variable storing the processed pixel value.

$$\begin{cases} \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & p_1 \\ q_1 & p_1q_1+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N, \\ mid(x,y) = arr(x',y') \oplus rand1(temp1); \\ \begin{bmatrix} x'' \\ y'' \end{bmatrix} = \begin{bmatrix} 1 & p_2 \\ q_2 & p_2q_2+1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \bmod N, \\ ciph(x'',y'') = mid(x,y), \\ temp1 = ciph(x'',y''). \end{cases} \tag{11}$$

In Eq. (11), the initial value of *temp*1 is set to $temp1 = \{[3.99999 \times (conf\_key_5) \times (1 - conf\_key_5)] \times 10^3\} \bmod 256$. SQ_3 and SQ_4 are generated from Eq. (5), with the initial values $conf\_key_3$ and $conf\_key_4$, respectively. p_i and q_i are calculated according to Eqs. (12) and (13), respectively. $conf\_key_3$, $conf\_key_4$ and $conf\_key_5$ are arbitrarily chosen as 0.12345678912340, 0.88795676859468 and 0.12345432167893, respectively. In Eq. (11), $arr(x', y')$ is the value of the randomly-selected pixel located at (x', y') in the plain-image by the reverse use of cat map, as Fig. 8(a) shows. $mid(x, y)$ represents the temporary position, as shown in Fig. 8(b), and $ciph(x'', y'')$ is the value of the encrypted pixel locating at position (x'', y'') after the confusion phase, as depicted in Fig. 8(c).

$$p_i = (SQ_3(2000+i) \times 10^9) \bmod 512, \tag{12}$$

$$q_i = (SQ_4(2000+i) \times 10^9) \bmod 512. \tag{13}$$

In the diffusion phase of Algorithm 2, Eq. (14) is employed to diffuse the intermediate image. In that equation, *rand*2 is calculated in the same way as for *rand*1. The initial value and the coefficient of the logistic map are set to 0.33798657654353 and 3.99999, respectively.

$$\begin{cases} ciph\_d(i) = ac(i) \oplus rand2(temp2), \\ temp2 = ciph\_d(i). \end{cases} \tag{14}$$

In Eq. (14), the initial value of *temp*2 is set to $temp2 = \{[3.99999 \times (key\_d_1) \times (1 - key\_d_1)] \times 10^3\} \bmod 256$, where key_d_1 is selected as 0.54567894324298. $ac(i)$ represents the value of the *i*th pixel in the one-dimensional sequence obtained in the confusion phase, and $ciph\_d(i)$ is the value of the *i*th pixel after diffusion. *temp*2 is a temporary variable storing the processed pixel value.

4. Experimental results

In this section, experimental results and performance analyses for the proposed schemes and a comparable algorithm called Bit Level Permutation (BLP) [28] are provided. The secret keys for BLP are the same as those used in [28]. All the simulations are performed on a computer equipped with an Intel Xeon 3.2 GHz CPU, 6 GB memory and 1 TB harddisk space running Windows 7 Professional. The compilation platform is Visual C++ 6.0 whereas some graphs are plotted using MATLAB 2009(a). In Algorithm 1, the round numbers *m* and *n* as shown in Fig.7 are selected as 2 and 1, respectively.

4.1. Histogram analysis

A histogram shows the distribution of pixel values in an image. The ideal histogram of the cipher-image should be uniform and is significantly different from that of the plain-image, so as to prevent the attacker from obtaining any useful statistical information. In our simulation, the test image is Barb at size 512 × 512.

As Fig. 9 illustrates, histograms of the cipher-images obtained by both Algorithms 1 and 2 are fairly uniform and are significantly different from that of the plain-image. They imply that the cipher-images cannot provide any useful information for the attacker to launch any statistical attacks on the cryptosystem.

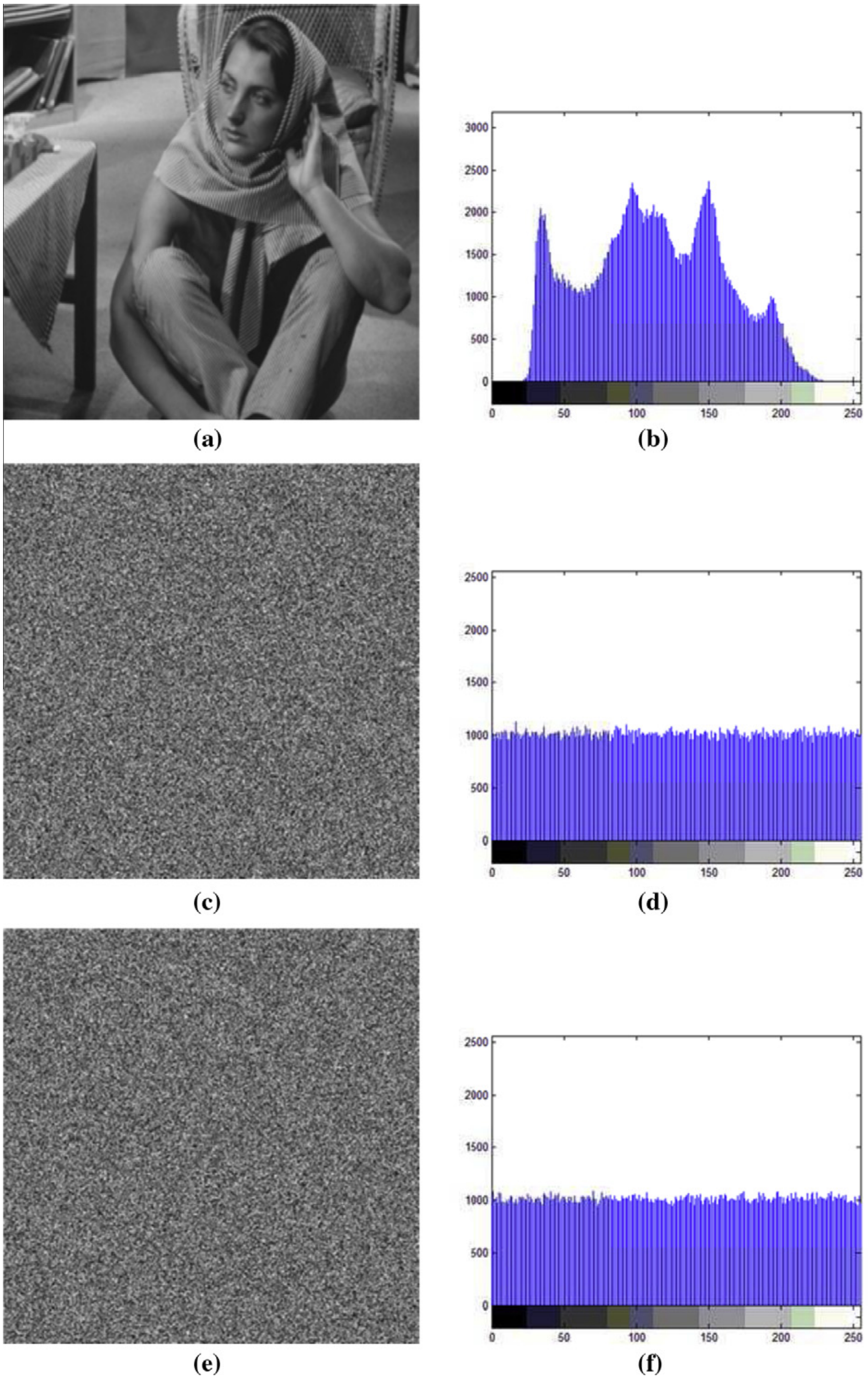


Fig. 9. (a) The original Barb image; (b) histogram of the plain-image; (c) cipher-image obtained by Algorithm 1; (d) histogram of the cipher-image obtained by Algorithm 1; (e) cipher-image obtained by Algorithm 2; (f) histogram of the cipher-image obtained by Algorithm 2.

4.2. Correlation analysis

There are two kinds of statistical analysis: histogram and correlation. In [27], Shannon pointed out that quite a large portion of cryptosystems can be attacked by statistical analysis. In this section, correlation analysis is given and the comparisons among the two proposed schemes and the comparable cryptosystem are made. The correlation between adjacent pixels in the plain-image is always high for an intelligible image since their pixel values are close to each other. The following steps are performed to evaluate an image's correlation property: (1) 2000 pixels are randomly selected as samples, (2) the correlations between two adjacent pixels in horizontal, vertical and diagonal directions are calculated by Eqs. (15)–(17).

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

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

$$D(x) = \frac{1}{S}\sum_{i=1}^{S}[x_i - E(x)]^2, \tag{17}$$

where x and y are the gray-levels of two adjacent pixels. $E(x)$ and $D(x)$ are the expectation and the variance of x, respectively. S denotes the total number of samples. The correlation coefficients of the cipher images obtained by the proposed schemes and BLP [28] using three test images are listed in Tables 2–4. The average coefficients for these images are calculated using Eq. (18), and the corresponding values can be found in Table 5.

$$average\_coefficient = \frac{|a| + |b| + |c|}{3}, \tag{18}$$

where $a,\ b$ and c are the 3 correlation coefficients along the same direction for 3 test images, Baboon, Elain and Barb, all having size 512×512.

In Table 5, only the horizontal correlation of Algorithm 1 is larger than that of BLP while other correlation coefficients of both Algorithms 1 and 2 are all smaller than the comparable scheme. Since the values of adjacent pixels in a meaningful image are similar, the correlation coefficients in the three directions (horizontal, vertical and diagonal) are very close to 1. Once the image is ciphered, there is hardly any relationship among the adjacent pixels, as Fig. 10 illustrates. The correlation distribution of the cipher-image are scattered over the entire plane.

4.3. Differential attack analysis

In differential attack analysis, two performance indices are usually adopted to investigate the influence of a 1-bit change in the plain-image to the corresponding cipher-image. They are *number of pixels change rate* (*NPCR*) and *unified average changing intensity* (*UACI*), as defined by Eqs. (19) and (21), respectively.

$$\text{NPCR} = \frac{\sum_{i,j} D(i,j)}{M \times N} \times 100\%, \tag{19}$$

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

$$D(i,j) = \begin{cases} 1, & c_1(i,j) \neq c_2(i,j), \\ 0, & otherwise, \end{cases} \tag{20}$$

Table 2
Correlation coefficients of the original image Baboon and the cipher-images obtained by the proposed schemes and BLP after the first encryption round.

	Plain-image	Algorithm 1	Algorithm 2	BLP
Horizontal	0.866406381	−0.000730460	0.002747506	0.003848277
Vertical	0.749632418	0.001095698	0.002014844	0.002382263
Diagonal	0.708438396	−0.002011907	−0.001647457	−0.002012074

Table 3
Correlation coefficients of the original image Elain and the cipher-images obtained by the proposed schemes and BLP after the first encryption round.

	Plain-image	Algorithm 1	Algorithm 2	BLP
Horizontal	0.975483358	−0.002568336	−0.001280413	0.000550055
Vertical	0.972883761	0.001834695	0.000731596	0.001650165
Diagonal	0.970997453	−0.000734822	0.001098201	−0.004590108

Table 4
Correlation coefficients of the original image Barb and the cipher-images obtained by the proposed schemes and BLP after the first encryption round.

	Plain-image	Algorithm 1	Algorithm 2	BLP
Horizontal	0.860607147	−0.008247047	0.000914915	−0.004214784
Vertical	0.959820449	−0.000366569	−0.002561757	0.003417700
Diagonal	0.877414346	0.001284758	0.001464938	−0.002557567

Table 5
Average correlation coefficients of the three cipher-images obtained by the proposed schemes and BLP after first encryption round.

	Plain-image	Algorithm 1	Algorithm 2	BLP
Horizontal	0.900832295	0.004031487	0.001647611	0.002871038
Vertical	0.894112209	0.001098987	0.001769399	0.002483376
Diagonal	0.852283398	0.001343829	0.001403532	0.003053249

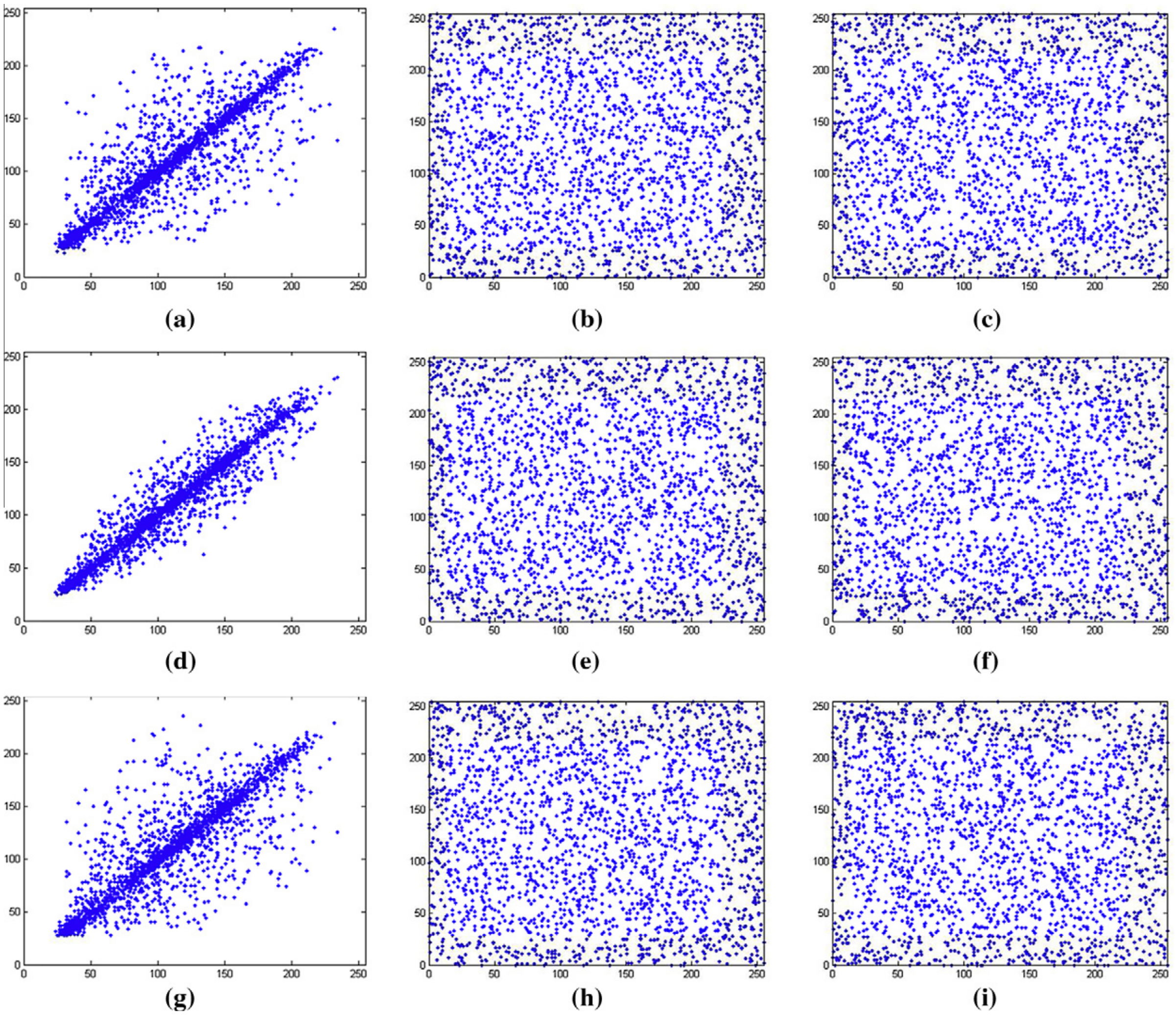


Fig. 10. Correlation plot of two adjacent pixels in the plain-image Barb in (a) horizontal; (d) vertical; (g) diagonal directions. Correlation plot of two adjacent pixels of the cipher-image obtained by Algorithm 1 in (b) horizontal; (e) vertical; (h) diagonal directions. Correlation plot of two adjacent pixels of the cipher-image obtained by Algorithm 2 in (c) horizontal; (f) vertical; (i) diagonal directions.

$$\text{UACI} = \frac{1}{M \times N}\left[\sum_{i,j}\frac{|c_1(i,j) - c_2(i,j)|}{255}\right] \times 100\%. \tag{21}$$

Table 6
NPCR performance.

Round	Algorithm 1 (%)	Algorithm 2 (%)	BLP (%)
1	79.7409058	99.5826721	0.1113892
2	99.6334076	99.6028900	51.8753052
3	99.6006012	99.6124268	99.5948792
4	99.6128082	99.6040344	99.6234894
5	99.5925903	99.5841980	99.6105194

Two test images are used. One is the original plain-image Barb while another is the 1-bit modified version of Barb obtained by changing the lower-right pixel from "01100010" to "01100011". In Eqs. (20) and (21), c_1 and c_2 are respectively the cipher-images corresponding to the two plain-images. They are obtained by encrypting the plain-image for several rounds using the same key.

The NPCR and UACI data are listed in Tables 6 and 7, respectively. They show that NPCR and UACI of the two proposed schemes can reach 99.6% and 33.4% in the second encryption round. Furthermore, the NPCR of Algorithm 2 can reach 99.58% even in the first encryption round. In Figs. 11 and 12, the NPCR and UACI data in the first five rounds are plotted, respectively. The graphs show that the two performance indices of our schemes rise at a higher rate than BLP and so fewer encryption rounds are required.

4.4. Speed performance

To compare the speed performance, the test image Barb at size 512×512 is encrypted by each cryptosystem and the time required for an encryption/decryption round is listed in Table 8.

Although some modifications and improvements have been made in Algorithms 1 and 2, the basic framework of the three cryptosystems is still the confusion–diffusion architecture. For Algorithm 1, one encryption round is composed of two dependent diffusion stages while for Algorithm 2, it is formed by one confusion and one diffusion stages. In BLP, an encryption round consists of 5 confusion and one diffusion stages. Furthermore, BLP needs 3 encryption rounds to reach

Table 7
UACI performance.

Round	Algorithm 1 (%)	Algorithm 2 (%)	BLP (%)
1	26.9027720	33.4217072	0.0366211
2	33.4751129	33.4533691	17.5086975
3	33.5353851	33.4342957	33.4701538
4	33.5403442	33.4747314	33.4617615
5	33.4915161	33.4392548	33.4877014

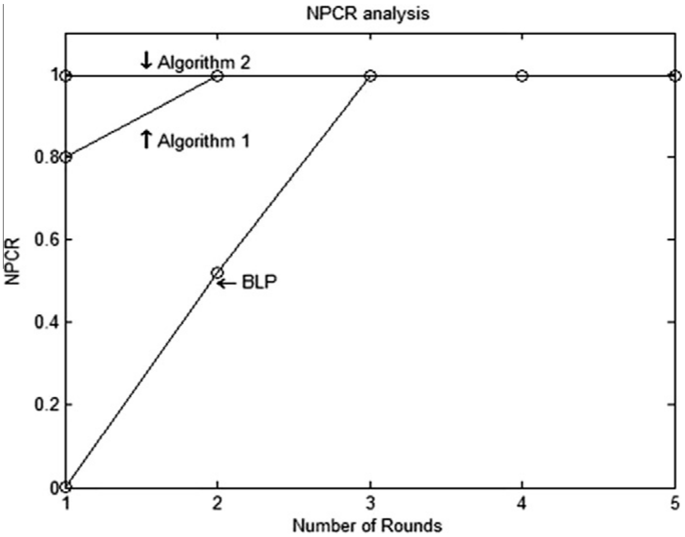


Fig. 11. NPCR analysis.

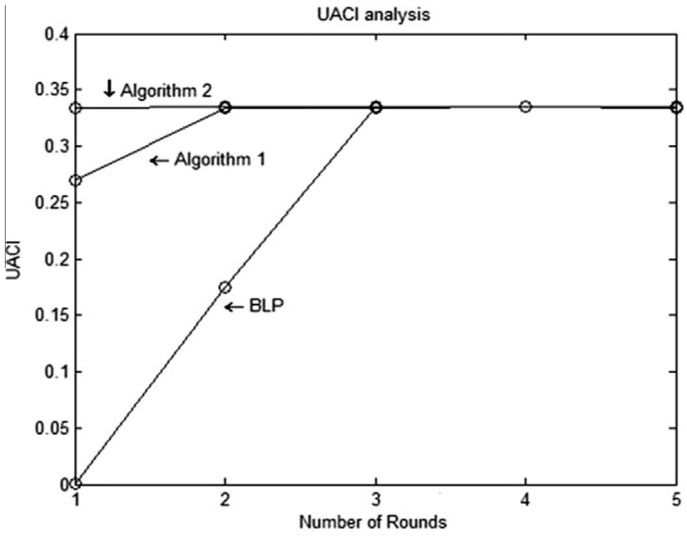


Fig. 12. UACI analysis.

Table 8
Time required for one encryption round.

Scheme	Average encryption time (ms)	Average decryption time (ms)	Average total time (ms)
Algorithm 1	10	10	20
Algorithm 2	11	10	21
BLP	29	30	59

Table 9
Information entropy of cipher-images obtained by the three schemes after the first round.

	Test image	Algorithm 1	Algorithm 2	BLP
1	Baboon	7.9993064	7.9991471	7.9992658
2	Frog	7.9990109	7.9993080	7.9992669
3	Elain	7.9992234	7.9994010	7.9993441
4	Clown	7.9989516	7.9994198	7.9993544
5	Girlface	7.9992966	7.9992513	7.9993126
6	Boat	7.9992699	7.9992597	7.9992474
7	Bridge	7.9992465	7.9993754	7.9993052
8	Crowd	7.9993157	7.9992847	7.9992794
9	Tank	7.9991996	7.9992811	7.9992868
Total		71.9928206	71.9937281	71.9936626
Average		7.9992023	7.9993031	7.9992958

the required NPCR level of 99.6%. Only one encryption round is required by Algorithm 2 to reach a NPCR level at 99.5826%. For Algorithm 1, two encryption rounds are needed. Therefore, we can conclude that the proposed schemes are more efficient in terms of the running time and the number of encryption rounds required.

4.5. Information entropy analysis

Information entropy is one of the criteria to measure the strength of a cryptosystem, which was firstly proposed by Shannon in 1949 [27]. The entropy $H(ms)$ of a message source ms is defined by the following formula:

$$H(ms) = \sum_{i=0}^{2^{N_b-1}} p(ms_i) log \frac{1}{p(ms_i)}. \tag{22}$$

In Eq. (22), N_b is the number of bits used to represent a pixel, $p(ms_i)$ means the probability of occurrence of ms_i and log denotes the base 2 logarithm so that the entropy is expressed in bits. For an ideal random source emits 2^8 symbols, i.e., $ms = \{ms_1, ms_2, \ldots, ms_{2^8}\}$, its information entropy is 8, as given by Eq. (22). Therefore, the information entropy of an encrypted image having 256 gray levels should be very close to 8. Otherwise, the information source is not sufficiently random and there exists a certain degree of predictability for breaking the cryptosystem.

The information entropy of the proposed schemes and BLP can be found in Table 9. As indicated by the calculated values, the information entropy of Algorithm 1 is smaller than that of BLP, while that of Algorithm 2 is larger than that of BLP. Nevertheless, all of them are very close to the ideal value 8. This means that the information leakage by the cipher-images is negligible and that the cryptosystems are secure against entropy attacks.

5. Conclusions

Two chaos-based image cryptosystems are designed using dependent diffusion and the reverse cat map. Unlike the traditional architecture, the confusion effect of the proposed cryptosystems cannot be removed by a homogeneous plain-image. Thus, our cryptosystems cannot be compromised by conventional known/chosen plaintext attacks. In the second proposed cryptosystem, a new kind of mapping from a pseudorandom position to another pseudorandom position is suggested to improve the confusion effect in the permutation stage. Simulation results show that the NPCR, UACI, and information entropy of the proposed schemes are better than those of a comparable cryptosystem, BLP. All these results justify the superior security and computational efficiency of our cryptosystems.

Acknowledgments

The work described in this paper was supported by the National Natural Science Foundation of China (Grant No. 60872040), the Program for Liaoning Excellent Talents in University, the Natural Science Foundation of Liaoning Province, China (Grant No. 20082037) and the Fundamental Research Funds for the Central Universities (Grant No.N110617001).

References

[1] Li S, Chen G, Zheng X. Chaos-based encryption for digital images and videos. Multimedia security handbook. CRC Press; 2005. p. 133–67, [Chapter 4].
[2] Lian SG, Sun JS, Wang ZQ. A block cipher based on a suitable use of the chaotic standard map. Chaos Solitons Fract 2005;26(1):117–29.
[3] Wang Y, Wong KW, Liao XF, Xiang T. A block cipher with dynamic S-boxes based on tent map. Commun Nonlinear Sci Numer Simul 2009;14(7):3089–99.
[4] Wang Y, Wong KW, Liao XF, Xiang T, Chen GR. A chaos-based image encryption algorithm with variable control parameters. Chaos Solitons Fract 2009;41(4):1773–83.
[5] Amin M, Faragallah OS, Abd El-Latif AA. A chaotic block cipher algorithm for image cryptosystems. Commun Nonlinear Sci Numer Simul 2010;15(11):3484–97.
[6] Pisarchik AN, Flores-Carmona NJ, Carpio-Valadez M. Encryption and decryption of images with chaotic map lattices. Chaos 2006;16(3):033118.
[7] Wong KW, Kwok BSH, Law WS. A fast image encryption scheme based on chaotic standard map. Phys Lett A 2008;372(15):2645–52.
[8] Alvarez G, Li SJ. Some basic cryptographic requirements for chaos-based cryptosystem. Int J Bifur Chaos 2006;16(8):2129–51.
[9] Li SJ, Li CQ, Chen GR, Bourbakis NG, Lo KT. A general quantitative cryptanalysis of permutation-only multimedia ciphers against plain-image attacks. Signal Process Image Commun 2008;23(3):212–23.
[10] Yang HQ, Liao XF, Wong KW, Zhang W, Wei PC. A new block cipher based on chaotic map and group theory. Chaos Solitons Fract 2009;40(1):50–9.
[11] Chiaraluce F, Ciccarelli L, Gambi E, Pierleoni P, Reginelli M. A new chaotic algorithm for video encryption. IEEE Trans Consum Electron 2002;48(4):838–44.
[12] Gao TG, Chen ZQ. A new image encryption algorithm based on hyper-chaos. Phys Lett A 2008;372(4):394–400.
[13] Patidar V, Pareek NK, Sud KK. A new substitution–diffusion based image cipher using chaotic standard and logistic maps. Commun Nonlinear Sci Numer Simul 2009;14(7):3056–75.
[14] Behnia S, Akhshani A, Mahmodi H, Akhavan A. A novel algorithm for image encryption based on mixture of chaotic maps. Chaos Solitons Fract 2008;35(2):408–19.
[15] Chen GR, Mao YB, Chui CK. A symmetric image encryption scheme based on 3-D chaotic cat maps. Chaos Solitons Fract 2004;21(3):749–61.
[16] Wong KW, Kwok BSH, Yuen CH. An efficient diffusion approach for chaos-based image encryption. Chaos Solitons Fract 2009;41(5):2652–63.
[17] Li SJ, Chen GR, Wong KW, Mou XQ, Cai YL. Baptista-type chaotic cryptosystems: problems and countermeasures. Phys Lett A 2004;332(5–6):368–75.
[18] Yen JC, Guo JI. Design of a new signal security system. In: Proc. IEEE international symposium circuits and systems (ISCAS 2002), vol. 4. Scottsdale, AZ, USA; 2002. p. 121–24.
[19] Yang HQ, Wong KW, Liao XF, Zhang W, Wei PC. A fast image encryption and authentication scheme based on chaotic maps. Commun Nonlinear Sci Numer Simul 2010;15(11):3507–17.
[20] Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifur Chaos 1998;8(6):1259–84.
[21] Rhouma R, Solak E, Arroyo D, Li SJ, Alvarez G, Belghith S. Comments on Modified Baptista type chaotic cryptosystem via matrix secret key. Phys Lett A 2008;372(33):5427–30.
[22] Mao YB, Chen GR, Lian SG. A novel fast image encryption scheme based on the 3-D chaotic baker map. Int J Bifur Chaos 2004;13(10):3613–24.
[23] Li CQ, Li SJ, Chen GR, Chen G, Hu L. Cryptanalysis of a new signal security system for multimedia data transmission. EURASIP J Appl Signal Process 2005;8:1277–88.
[24] Li CQ, L SJ, Alvarenz G, Chen GR, Lo KT. Cryptanalysis of two chaotic encryption schemes based on circular bit shift and XOR operations. Phys Lett A 2007;369(1–2):23–30.
[25] Chen HC, Guo JI, Huang LC, Yen JC. Design and realization of a new signal security system for multimedia data transmission. EURASIP J Appl Signal Process 2003;13:1291–305.
[26] Li SJ, Zheng X. On the security of an image encryption method. In: IEEE international conference on image proceessing (ICIP'02), vol. 2. Rochester, NY, USA; 2002. p. 925–28.
[27] Shannon CE. Communication theory of secrecy system. Bell Syst Tech J 1949;28:656–715.
[28] Zhu ZL, Zhang W, Wong KW, Yu H. A chaos-based symmetric image encryption scheme using a bit-level permutation. Inform Sci 2011;181(6):1171–86.