Optics Communications 284 (2011) 2775–2780

Contents lists available at ScienceDirect

Optics Communications

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

A novel image encryption method based on total shuffling scheme

Guoji Zhang [a], Qing Liu [b,*]

[a] Department of Mathematical Sciences, South China University of Technology, 381 Wushan RD, Guangzhou 510641, PR China
[b] School of Computer Science and Engineering, South China University of Technology, Higher Education Mega Centre, Guangzhou 510006, PR China

ARTICLE INFO

Article history:
Received 8 December 2010
Received in revised form 12 February 2011
Accepted 15 February 2011
Available online 3 March 2011

Keywords:
Skew tent map
Chaotic system
Image encryption
Cryptography

ABSTRACT

In this paper, a novel image encryption method based on skew tent chaotic map and permutation–diffusion architecture is proposed. In the proposed method, the P-box is chosen as the same size of plain-image, which shuffles the positions of pixels totally. The keystream generated by skew tent chaotic map is related to the plain-image. Statistical analysis, information entropy analysis, and sensitivity analysis to plaintext and key on the proposed scheme are provided in this paper. It can be seen that this algorithm is efficient and reliable, with high potential to be adopted for network security and secure communications.

1. Introduction

With the rapid growth of the requirement of image transmission on Internet, protection of digital information against illegal usage becomes more and more important. Due to bulky data capacity and high correlation among pixels in image files, traditional techniques are not suitable for image encryption [1]. Compared with traditional methods (such as AES and DES), chaos-based image encryption schemes have shown superior performance [2–4].

The general permutation–diffusion procedure [5] for chaos based image encryption is composed of two steps: pixel permutation and diffusion. Fig. 1 illustrates its architecture. In the permutation process, the position of image pixels is changed. In the diffusion process, the pixel values are modified sequentially so that a tiny change for one pixel can spread out to almost all pixels in the whole image.

Recently, some chaos-based image encryption algorithms with a permutation–diffusion structure are broken [6–11]. The common characteristic of these algorithms is: the keystream in the diffusion step only depends on the key. The keystream used to encrypt different plain-images are the same if the key keeps unchanged. The attacker can obtain the keystream by known-plaintext attack and chosen-plaintext attack [7,9,11]. Correspondingly, the algorithm degenerates permutation-only architecture. The permutations based on two-dimensional chaotic maps have short periodicity. Furthermore, a general practical method was proposed recently which can break any permutation-only encryption [12].

To enhance the security, a good encryption algorithm with a permutation–diffusion structure should use a keystream related to plain-image and avoid the short periodicity of permutation.

In this paper, a chaos-based image encryption method is proposed. With Wang's version of the permutation–diffusion architecture [13], the algorithm uses different keystreams when encrypting different plain-images (even with the same key). The algorithm generates a P-box with the same size of plain-image by a skew tent chaos map, which shuffles the positions of pixels totally. The keystream in the diffusion step depends on both the key (the initial value x_0 and the control parameters p of the skew tent map) and the plain-image. Both theoretical analyses and computer simulations verify the feasibility and superiority of the proposed encryption algorithm.

This paper is organized as follows. In Section 2, skew tent chaotic system and the permutation operator based on it are introduced. In Section 3, the proposed algorithm is described in detail, including how to generate a total shuffling P-box and how to employ a feedback from the permuted plain-image in diffusion step. In Section 4 performance analyses and simulation results are reported. Section 5 gives some conclusions.

2. Permutation based on skew tent map

2.1. The skew tent map

The skew tent chaotic map [14] can be described as follows:

$$F(x) = \begin{cases} x/p, & \text{if } x \in [0,p] \\ (1-x)/(1-p), & \text{if } x \in (p,1] \end{cases} \quad (1)$$

where $x \in [0,1]$ is the state of the system, and $p \in (0,1)$ is the control parameter. For any $p \in (0,1)$, the piecewise linear map (1) has a

* Corresponding author.
E-mail addresses: magjzh@scut.edu.cn (G. Zhang), mstlqbk@gmail.com (Q. Liu).

0030-4018/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.optcom.2011.02.039

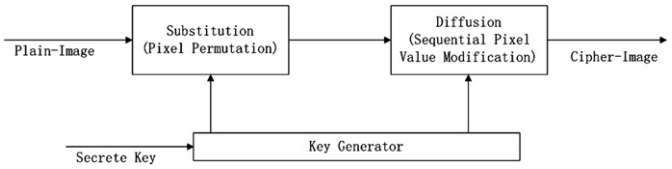

Fig. 1. General permutation–diffusion architecture.

positive Lyapunov exponent and thus is always chaotic [15]. So all the (x_0,p) where $x_0\in[0,1]$ and $p\in(0,1)$ can be used as secret keys. Figs. 2 and 3 show the chaotic and bifurcation behavior of system (1) for $x_0=0.27, p=0.4$.

2.2. Permutation operator based on skew tent map system

For a 256 gray-scale image of size $M\times N$, it is an integer matrix of M rows N columns, in which the values range from 0 to 255. Its data can be treated as a one-dimensional vector $P=\{p_0,p_1,\cdots,p_{MN-1}\}$, where p_i denotes the gray level of the image pixel in the row **floor** (i/N) column **mod** (i,N).

Given x_0 and p, to change the position of image pixel, we take the following steps:

step 1 Iterate the skew tent map $x_{i+1}=F(x_i)$ by using Eq. (1) for L times to get rid of transient effect, where L is a constant;
step 2 Continue to iterate the skew tent map for MN times, take out the state values $\{x_{L+1},x_{L+1},\cdots,x_{L+MN}\}$;
step 3 Sort the MN values above and get $\{\overline{x}_{L+1},\overline{x}_{L+1},\cdots,\overline{x}_{L+MN}\}$;
step 4 Find the position of values $\{\overline{x}_{L+1},\overline{x}_{L+1},\cdots,\overline{x}_{L+MN}\}$ in $\{x_{L+1},x_{L+1},\cdots,x_{L+MN}\}$ and mark down the transform positions $T=\{t_1,t_2,\cdots,t_{MN}\}$, where $\overline{x}_{L+i}$ is exactly the value of x_{L+t_i}. T is used as our P-box;
step 5 Shuffle the values in P by P-box T, getting $P'=\{p'_0,p'_1,\cdots,p'_{MN-1}\}$, where $p'_{i-1}=p_{t_i-1}(i=1,2,\cdots,MN)$.

Unlike the traditional block encryption methods such as DES and AES, the proposed algorithm shuffles the positions of image pixels totally, using a P-box with the same size of plain-image. Compared with the permutations based on two-dimensional chaotic maps, the permutation operator proposed gets rid of the flaw of short periodicity, since the relationship of original and shuffled position of one pixel is not directly related to the chaotic map.

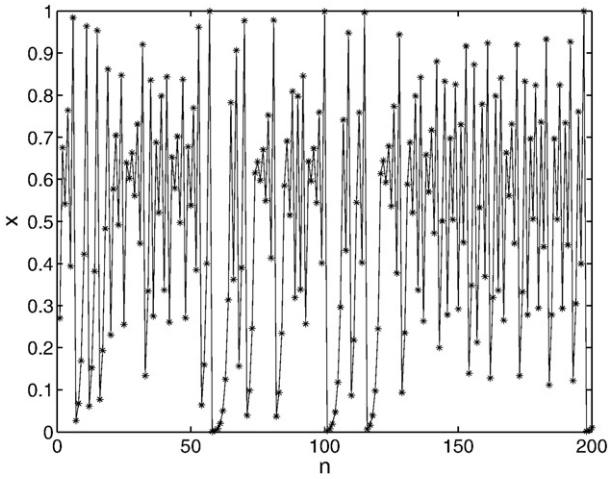

Fig. 2. Chaotic behavior of skew tent system $(x_0=0.27, p=0.4)$.

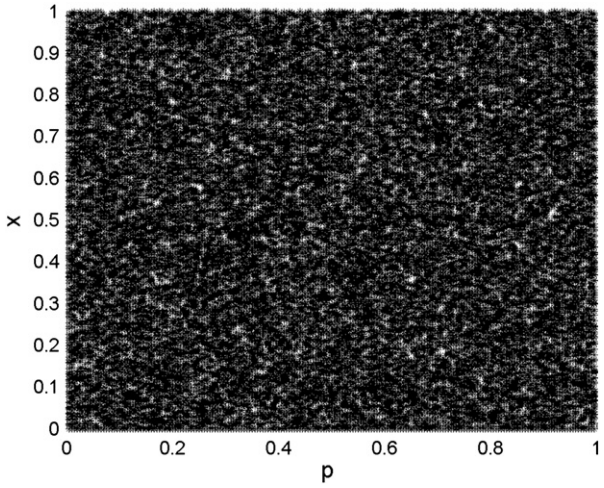

Fig. 3. Bifurcation behavior of skew tent system $(x_0=0.27, p=0.4)$.

3. The scheme of image encryption and decryption

The encryption algorithm proposed in this paper is based on permutation–diffusion architecture. The initial value x_0 and the control parameter p of skew tent map are used as secret key. The permutation step is described in Section 2.2. With Wang's diffusion scheme [13], step 5 in encryption makes the diffusion keystream not only related on the key but also the plain-image. For a gray image of size $M\times N$, we treat its data as a one-dimensional vector $P=\{p_0,p_1,\cdots,p_{MN-1}\}$. The algorithm can be described as follows (take 256 gray-scale image as an example).

3.1. Encryption algorithm

step 1 Iterate the skew tent map $x_{i+1}=F(x_i)$ by Eq. (1). Generate a P-box T and shuffle the values in P by T to get P', as described in Section 2.2;
step 2 Let $i\leftarrow 0$;
step 3 Obtain an 8-bit random code d_i according to the following formula:

$$d_i = \mathrm{mod}\left(\mathrm{floor}\left(x\times 2^{48}\right),256\right) \tag{2}$$

where x is the current state value of the skew tent map system;
step 4 Compute the corresponding pixel data of the cipher-image by using the values of the currently operated pixel and the previously operated pixels, according to the following formula:

$$c_i = p'_i \oplus \mathrm{mod}\left(p_{i-1}+d'_i,2^8\right) \tag{3}$$

where $\oplus$ is bitwise XOR operator, and c_i is the output pixel data. One may set the initial value p'_{-1} as a constant. The inverse form of Eq. (3) is

$$p'_i = c_i \oplus \mathrm{mod}\left(p_{i-1}+d'_i,2^8\right); \tag{4}$$

step 5 Compute k according to the follow formula:

$$k = 1 + \mathrm{mod}(c_i,2). \tag{5}$$

Then, iterate the skew tent map $x\leftarrow F(x)$ for k times;
step 6 Let $i\leftarrow i+1$, return to step 4 until i reaches MN.

3.2. Decryption algorithm

The decryption procedure is similar to that of the encryption process in the reverse order:

step 1 Generate a P-box T by skew tent map according to the x_0 and p;
step 2 Obtain $P' = \{p'_0, p'_1, \cdots, p'_{MN-1}\}$ from $C = \{c_0, c_1, \cdots, c_{MN-1}\}$. Perform the reverse operations to remove the effect of diffusion. All operations are the same as steps 3–6 in the encryption process except that Eq. (3) is replaced by Eq. (4) here;
step 3 Perform the reverse operation to remove the effect of permutation by using the P-box T.

4. Performance test and analysis

4.1. Key space analysis

Key space size is the total number of different keys which can be used in the encryption. A good encryption algorithm should be sensitive to the secret keys, and the key space should be large enough to make brute-force attack impossible. In the proposed algorithm, the key consists of the initial value x_0 and the parameter p, where $x_0 \in (0,1)$, and $p \in (0,1)$. According to the IEEE floating-point standard [16], the computational precision of the 64-bit double-precision numbers is 2^{-52}. The total number of different values which can be used as x_0 is more than 2^{52}, so is the number of p. Therefore, the key space is bigger than $2^{52} \cdot 2^{52} = 2^{104}$. Such a big key space can provide a sufficient security against brute-force attacks [15,17,18].

4.2. Statistical analysis

Shannon suggested that diffusion and confusion should be employed in a cryptosystem [19] for the purpose of frustrating the powerful statistical analysis. In the proposed encryption algorithm, one dynamic P-box generated by skew tent map is used to permute the plain-image, which can be considered as a confusion process. After permutation, a random number sequence was generated by skew tent map to modify the pixel values sequentially. Therefore, the diffused image is randomly distributed. This is shown by a test on the histograms of the cipher-images in Section 4.2.1, the correlations of adjacent pixels in the cipher-image and cipher-image in Section 4.2.2, and the information entropy of the cipher-image in Section 4.2.3.

4.2.1. Histograms of cipher-image

Figs. 4 and 5 depicts the histograms of the plain-image "lenna" and the corresponding cipher-image. The histogram of the encrypted image is nearly uniformly distributed, which can well protect the information of the image to withstand the statistical attack [18].

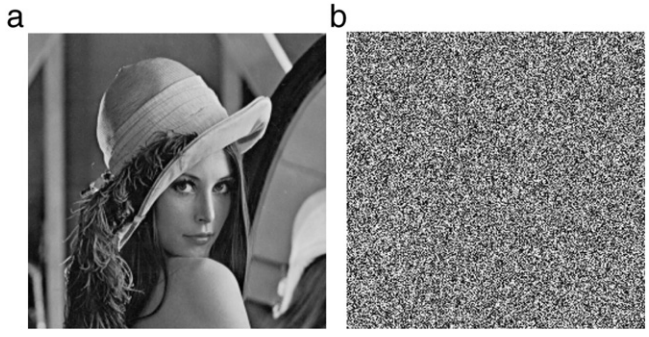

Fig. 4. (a) Plain-image "lenna"; and (b) Corresponding cipher-image ($x_0 = 0.123456$, $p = 0.654321$).

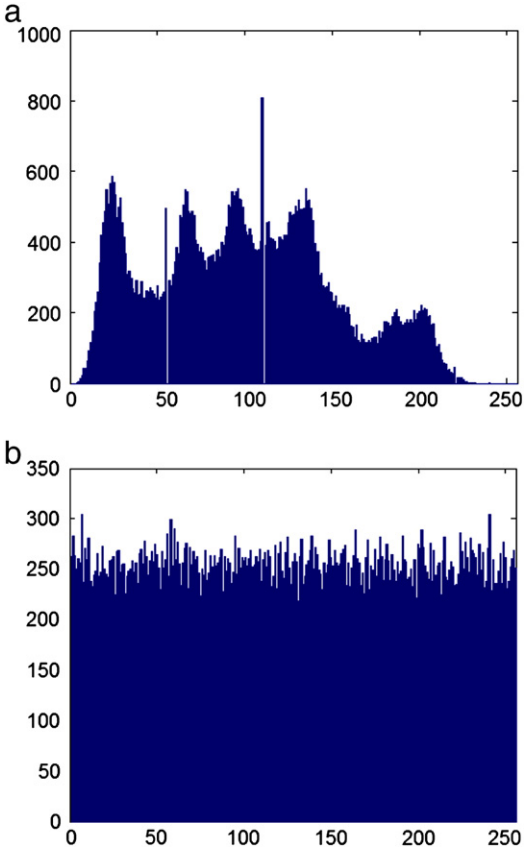

Fig. 5. Histogram (a) Plain-image "lenna", and (b) Cipher-image using proposed method.

4.2.2. Correlation of two adjacent pixels

There exists a high correlation between pixels of an image which is called intrinsic feature. Thus, a secure encryption scheme should remove it to improve resistance against statistical analysis. To test the correlation between two-adjacent pixels in plain-image and cipher-image, we randomly select 1000 pairs of two-adjacent pixels (in vertical, horizontal, and diagonal direction) from plain-image and cipher-image, and calculate the coefficient of each pair by Eq. (9), as Akhshani did [20].

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{6}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))^2 \tag{7}$$

$$\mathrm{cov}(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x))(y_i - E(y)) \tag{8}$$

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{9}$$

where x and y are gray-scale values of two-adjacent pixels in the image. The correlation of two horizontally adjacent pixels in the plain-image and cipher-image "lenna" is shown in Fig. 6.

Table 1 shows the correlation coefficients of two adjacent pixels in the plain and the cipher-image. This correlation analysis proves that the chaotic encryption scheme satisfies zero co-correlation, which is a private high-level security. Compared with Sun's algorithm [18], it shows superior performance.

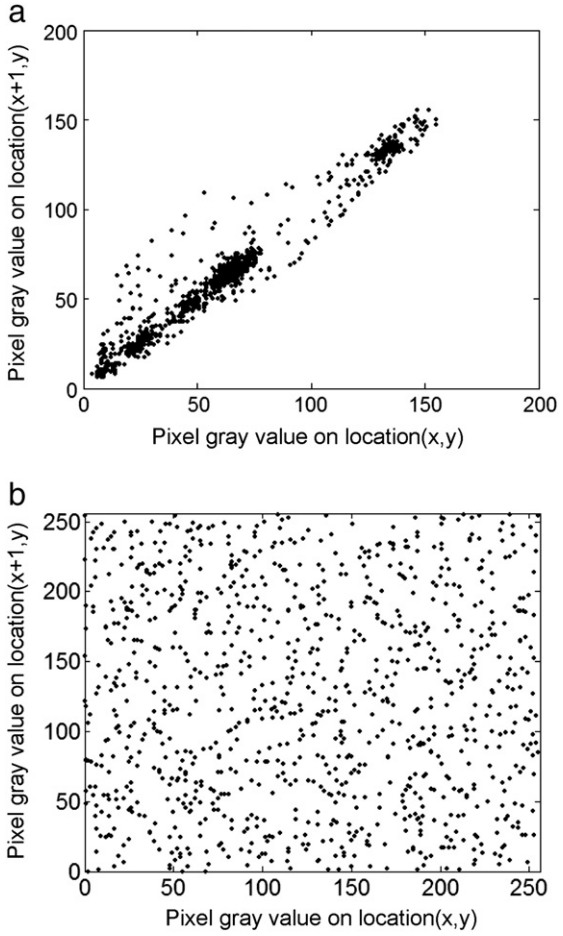

Fig. 6. Correlations of two horizontally adjacent pixels: (a) Correlation of the plain-image; and (b) Correlation of the cipher-image.

4.2.3. Information entropy analysis

The entropy is the most outstanding feature of the randomness. The information entropy $H(s)$ of a message source s can be calculated as [21]:

$$H(s) = \sum_{i=0}^{2^N-1} p(s_i)\log_2 \frac{1}{p(s_i)} \tag{10}$$

where $p(s_i)$ denotes the probability of symbol s_i. For a true random source emitting 2^N symbols, the entropy should be N. Take 256 gray-scale image for example, and the pixel data have 2^8 possible values, so the entropy of a "true random" image must be 8.

The entropy of the cipher-images is shown in Table 2. The obtained values are very close to the theoretical value 8. This means that information leakage in the encryption process is negligible and the encryption scheme is secure upon entropy attack. Table 3 shows that the proposed algorithm compared with the existing algorithms mentioned in Ref. [18] is better.

Table 1
Correlation coefficients of two adjacent pixels in the plain and the cipher-image.

Direction	Plain-image	Cipher-image by proposed algorithm	Cipher-image by Sun's algorithm
Horizontal	0.939918	−0.000848277	0.00128
vertical	0.969233	0.00370914	−0.00261
diagonal	0.937176	−0.000188985	0.00014

Table 2
Entropy value for the cipher-images.

Cipher-images	Entropy value
Lenna	7.99748
Airplane	7.99925
Baboon	7.9993

Table 3
Entropy value for the cipher-images of "lenna".

Algorithm	Entropy value
Baptista's	7.926
Wong's	7.969
Xiang's	7.995
Sun's	7.9965
Proposed	7.99748

4.3. Sensitivity analysis

In order to test the different range between two images, we measure the *NPCR* [22] (number of pixels change rage) and *UACI* [23] (unified average changing intensity) by Eqs. (11) and (12).

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

$$UACI = \frac{1}{W \times H}\left[\sum_{i,j} \frac{|c_1(i,j) - c_2(i,j)|}{255}\right] \times 100 \tag{12}$$

where C_1 and C_2 are two images with the same size $(M \times N)$. If $c_1(i,j) = c_2(i,j)$ then $D(i,j) = 1$, otherwise $D(i,j) = 0$.

4.3.1. Key sensitivity

We encrypted a gray image "baboon" with size 512×512, using $(x_0 = 0.123456789, p = 0.23)$ as key, then encrypted the same plain-image with a slightly different key $(x_0' = x_0 + 10^{-10}, p' = p)$, and the

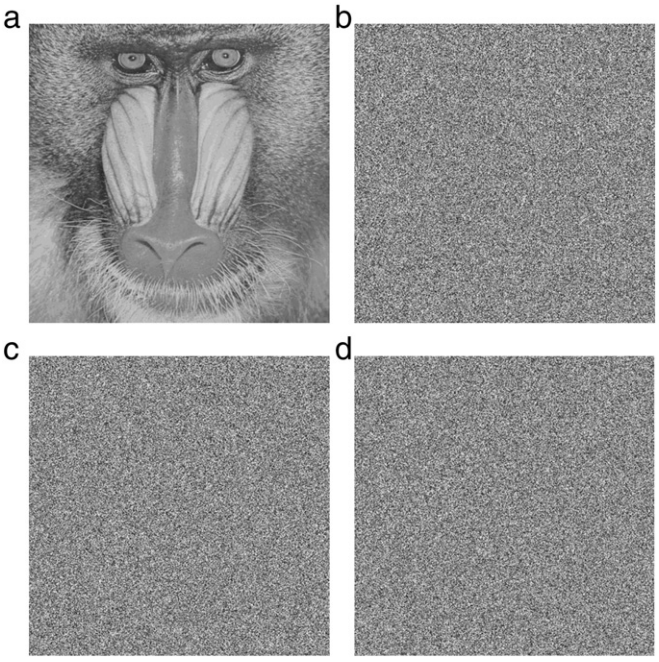

Fig. 7. (a) Plain-image "baboon"; (b) Cipher-image with key (x_0, p); (c) Cipher-image with a slightly different key $(x_0 + 10^{-10}, p)$; and (d) Decrypted image of b with the key $(x_0' = x_0 + 10^{-10}, p' = p)$.

Table 4
NPCR and *UACI* between cipher-image with key (x_0, p) and other cipher-images with slightly different keys.

Key	NPCR (%)	UACI (%)
(0.1234567891, 0.23)	99.6052	33.4132
(0.1234567892, 0.23)	99.6227	33.4865
(0.1234567890, 0.230000001)	99.6002	33.4608
(0.1234567890, 0.230000002)	99.6204	33.4657

corresponding cipher-images are shown in Fig. 7. The *NPCR* and *UACI* between the two cipher-images are 99.6052% and 33.4132%. A slight change in key makes the cipher-image totally different. More results of similar test are shown in Table 4, Figs. 8 and 9.

More than 99% pixels in cipher-image change their gray rate when the key just changes 10^{-10}. The proposed algorithm provides high key sensitivity.

4.3.2. Plaintext sensitivity

In order to resist differential attack, a minor alternation in the plain-image should cause a substantial change in the cipher-image. To test the influence of one-pixel change on the whole image encrypted by the proposed algorithm, we encrypted ($x_0 = 0.123456789$, $p = 0.23$) plain-images which have only one pixel different, marking down the *NPCR* and *UACI* between the cipher-images. We test 3 groups of plain-images, and the results are shown in Table 5.

In our tests, changing one pixel value in plain-image can cause at lease more than 30% pixel change in cipher-image. We encrypted the 3 groups of plain-images for two rounds (the 2nd round key: $x_0 = 0.987654321, p = 0.1234$), the change between cipher-images becomes even bigger, which is shown in Table 6.

In order to obtain high plaintext sensitivity, we suggest iterating the proposed algorithm more than 2 times to get a good ability to resist differential attack.

4.4. Speed analysis

The speed of the proposed image encryption/decryption technique has been analyzed on a personal computer using MinGW complier (GCC4.4.3). The average time of encryption/decryption on 256 gray-scale images of size 512×512 is shorter than 40 ms. The computer used in this test is 2.00 GHz AMD Duel core with 2 GB RAM running on Windows XP pro. The proposed algorithm can be used in internet applications, where the encryption/decryption time is much shorter than the transmission time.

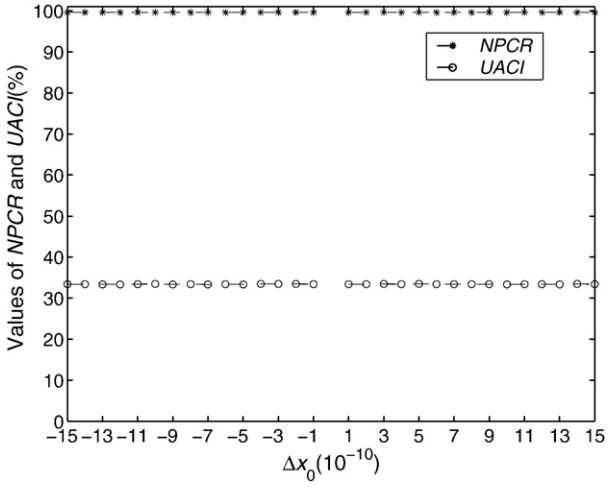

Fig. 8. *NPCR* and *UACI* between cipher-image with key $(x_0 = 0.123456789, p = 0.23)$ and other cipher-images with slightly different keys $(x_0 + \Delta x_0, p)$.

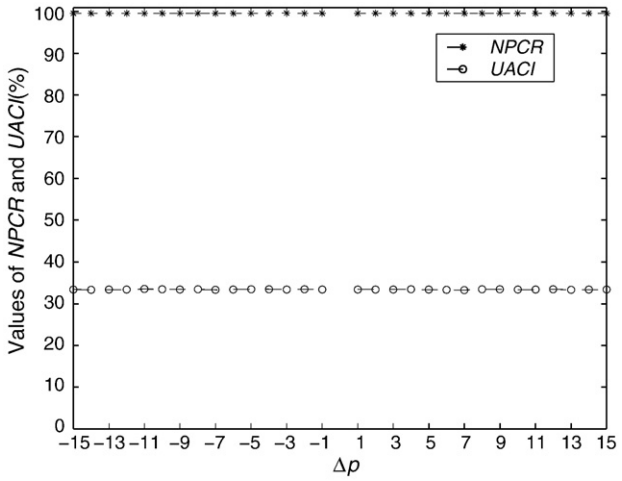

Fig. 9. *NPCR* and *UACI* between cipher-image with key $(x_0 = 0.123456789, p = 0.23)$ and other cipher-images with slightly different keys $(x_0, p + \Delta p)$.

4.5. Resistance to known-plaintext and chosen-plaintext attacks

In the diffusion step, a feedback from the cipher-image is employed to change the number of iterations of the skew tent map. In Eq. (3), d_i, extracted from the skew tent map, depends on the permuted plain-image. When different plain-images are encrypted, the corresponding keystream is not the same. The attacker cannot obtain useful information by encrypting some special images since the resultant information is related to those chosen-images. Therefore, the attacks proposed in Refs. [7,9,11] become ineffective on this new scheme. The proposed scheme can well resist the known-plaintext and the chosen-plaintext attacks.

5. Conclusions

In this paper, a novel chaos-based image encryption with a permutation–diffusion architecture is proposed. In the permutation step, the scheme generates a P-box with the same size of plain-image by a Skew tent chaos map. The keystream in the diffusion step depends on both the key (the initial value and the control parameters of the skew tent map) and the plain-image. The key space is large enough to resist brute-force attacks. Statistical analysis shows that the scheme can well protect the image from the statistical attack. The scheme possesses high key sensitivity and gets a good ability to anti differential attack. It can

Table 5
NPCR and *UACI* between cipher-images with slightly different plain-images.

Plain-image	*NPCR* (%)	*UACI* (%)
Lenna	37.6389	12.7034
Airplane	84.1213	28.275
Baboon	84.1255	28.1799

Table 6
NPCR and *UACI* between (2 rounds) cipher-images with slightly different plain-image.

Plain-image	NPCR (%)	UACI (%)
Lenna	99.6063	33.4758
Airplane	99.6246	33.4723
Baboon	99.6048	33.4554

avoid the known-plaintext and chosen-plaintext attack. With a high encryption speed, it can be used in internet applications.

Acknowledgements

This work was supported by the Natural Science Foundation of Guangdong Province (Grant no. 8151064101000033).

References

[1] S. Li, G. Chen, A. Cheung, B. Bhargava, K.-T. Lo, On the Design of Perceptual MPEG-video Encryption Algorithms, CoRR abs/cs/0501014, 2005.
[2] J. Fridrich, International Journal of Bifurcation and Chaos 8 (1998) 1259.
[3] F. Sun, S. Liu, Z. Li, Chaos, Solitons & Fractals 38 (2008) 631.
[4] Z. Liu, Q. Guo, L. Xu, M.A. Ahmad, S. Liu, Optics Express 18 (2010) 12033.
[5] G. Chen, Y. Mao, C.K. Chui, Chaos, Solitons & Fractals 21 (2004) 749.
[6] Z.-H. Guan, F. Huang, W. Guan, Physics Letters A 346 (2005) 153.
[7] D. Xiao, X. Liao, P. Wei, Chaos, Solitons & Fractals 40 (2009) 2191.
[8] X. Tong, M. Cui, Image and Vision Computing 26 (2008) 843.
[9] C. Li, S. Li, G. Chen, W.A. Halang, Image and Vision Computing 27 (2009) 1035.
[10] V. Patidar, N. Pareek, K. Sud, Communications in Nonlinear Science and Numerical Simulation 14 (2009) 3056.
[11] R. Rhouma, E. Solak, S. Belghith, Communications in Nonlinear Science and Numerical Simulation 15 (2010) 1887.
[12] C. Li, K.-T. Lo, S. Belghith, Optimal quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks, Signal Processing 91 (2011) 949.
[13] Y. Wang, K.-W. Wong, X. Liao, T. Xiang, G. Chen, Chaos, Solitons & Fractals 41 (2009) 1773.
[14] L. Billings, E. Bollt, Chaos, Solitons & Fractals 12 (2001) 365, Chaos in Ecology.
[15] G. Alvarez, S. Li, International Journal of Bifurcation and Chaos 16 (2006) 2129.
[16] IEEE Task P754, IEEE 754–2008, Standard for Floating-Point Arithmetic, 2008.
[17] D. Stinson, Cryptography: Theory and Practice, Second ed.CRC/C&H, 2002.
[18] F. Sun, Z.L.S. Liu, Optics Communications 283 (2010) 2066.
[19] C.E. Shannon, Bell Systems Technical Journal 28 (1949) 656.
[20] A. Akhshani, S. Behnia, A. Akhavan, H.A. Hassan, Z. Hassan, Optics Communications 283 (2010) 3259.
[21] A.D. Santis, A.L. Ferrara, B. Masucci, Discrete Applied Mathematics 154 (2006) 234, Coding and Cryptography.
[22] A.G. Bluman, Elementary Statistics: A Step by Step Approach, WCB/McGraw-Hill, 1997.
[23] Y. Mao, G. Chen, S. Lian, International Journal of Bifurcation and Chaos 14 (2004) 3613.