Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

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

An image encryption scheme with a pseudorandom permutation based on chaotic maps

Ji Won Yoon [a], Hyoungshick Kim [b,*]

[a] University of Oxford, Parks Road, Oxford, UK
[b] University of Cambridge, Cambridge, UK

ARTICLE INFO

Article history:
Received 4 June 2009
Received in revised form 17 January 2010
Accepted 28 January 2010
Available online 4 February 2010

Keywords:
Chaotic maps
Image encryption
Pseudorandom permutation

ABSTRACT

Many research efforts for image encryption schemes have elaborated for designing nonlinear functions since security of these schemes closely depends on inherent characteristics of nonlinear functions. It is commonly believed that a chaotic map can be used as a good candidate of a nonlinear component for image encryption schemes. We propose a new image encryption algorithm using a large pseudorandom permutation which is combinatorially generated from small permutation matrices based on chaotic maps. The random-like nature of chaos is effectively spread into encrypted images by using the permutation matrix. The experimental results show that the proposed encryption scheme provides comparable security with that of the conventional image encryption schemes based on Baker map or Logistic map.

1. Introduction

With the rapid developments in the multimedia industry and communications, a great deal of concerns have been raised in the security of multimedia data transmitted or stored over open channels. A major challenge is to protect confidentiality for multimedia data in digital distribution networks. The most effective method is to encrypt multimedia data so that the only authorized entities with the key can decrypt them. In practice Advanced Encryption Standard (AES) [1] has been widely recognized as de facto standard in multimedia industry. However, it is known that conventional encryption schemes such as AES have obvious limitations on the multimedia specific requirements [2] as follows:

- *Perceptual quality control:* An encryption algorithm can be used to intentionally degrade the quality of perception, but still keep the multimedia data visually perceivable.
- *Codec/format compliant:* It could be desired that the encryption algorithm preserves compression format of the multimedia data, so that the standard decoders can still decode the encrypted data without crashing.
- *Real-time constraint:* In many multimedia applications, it is important that the encryption and decryption algorithms are very efficient to access multimedia data in real time similar to unprotected data. As one of speed up techniques, some important parts of multimedia data can be selectively encrypted.

To meet these challenges, many multimedia encryption schemes have been proposed [2,3]. A common design principle is to use nonlinear functions as fundamental building blocks for encryption [4]. Nonlinear components are essential to every strong cryptographic primitive. In conventional block cipher algorithms such as AES nonlinear functions are generally implemented as S-box which is a table-driven nonlinear substitution operation [1].

[*] Corresponding author. Address: Statistics Department, Trinity College Dublin, Ireland.
E-mail addresses: jwyoon@robots.ox.ac.uk, yoonj@tcd.ie (J.W. Yoon), hk331@cam.ac.uk (H. Kim).

1007-5704/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cnsns.2010.01.041

For image encryption, chaotic maps can be proper alternatives to S-box since they have shown some interesting properties such as aperiodicity, sensitive dependence on initial conditions and topological transitivity, ergodicity and random-like behaviors [5–14]. Most chaos-based encryption schemes basically produce a long random sequence by using chaotic maps as pseudorandom number generators and encrypt a plaintext image with the random sequence [11].

In particular, many chaotic encryption schemes are based on permutation. This approach first produces a pseudorandom permutation from chaotic maps and permutes plaintext images with the pseudorandom permutation [5,14,6]. A discrete chaotic cat map is discussed to transform pixel positions of the plaintext image into new positions [15]. Fridrich showed that plaintext images can be reasonably shuffled by permutation operations via the construction of Baker map [5]. Alternatively, a chaotic Kolmogorov-flow-based image encryption is developed [14].

However, the traditional pseudorandom permutation should be severally repeated to provide reasonable security since their approaches shuffle image pixels locally. Our main goals are to achieve the high performance of pseudorandom permutation by using a large permutation matrix and high robustness against statistical cryptanalysis. We propose a new multimedia encryption scheme which provides reasonable security even if a generated random sequence from chaotic maps has short periodic orbits. In the proposed algorithm, a pseudorandom sequence S_1 generated from chaotic maps is combinatorially extended to a relatively long pseudorandom sequence S_2 to provide sufficient security of the encrypted image by constructing permutation matrix with the sequence S_1.

The remainder of the paper is organized as follows. In Section 2 we describe overall framework of the proposed image encryption scheme. Section 3 presents how to construct the permutation matrix with chaotic maps in detail. Section 4 evaluates the security of the proposed algorithm via several randomness tests. Finally, in Section 5 we give conclusions of this paper.

2. Proposed algorithm

Our proposed encryption scheme is pictorially shown in Fig. 1. Two sub-figures represent encryption and decryption process. A key is defined as initial conditions for a chaotic map and parameters related to small permutation matrices. In both

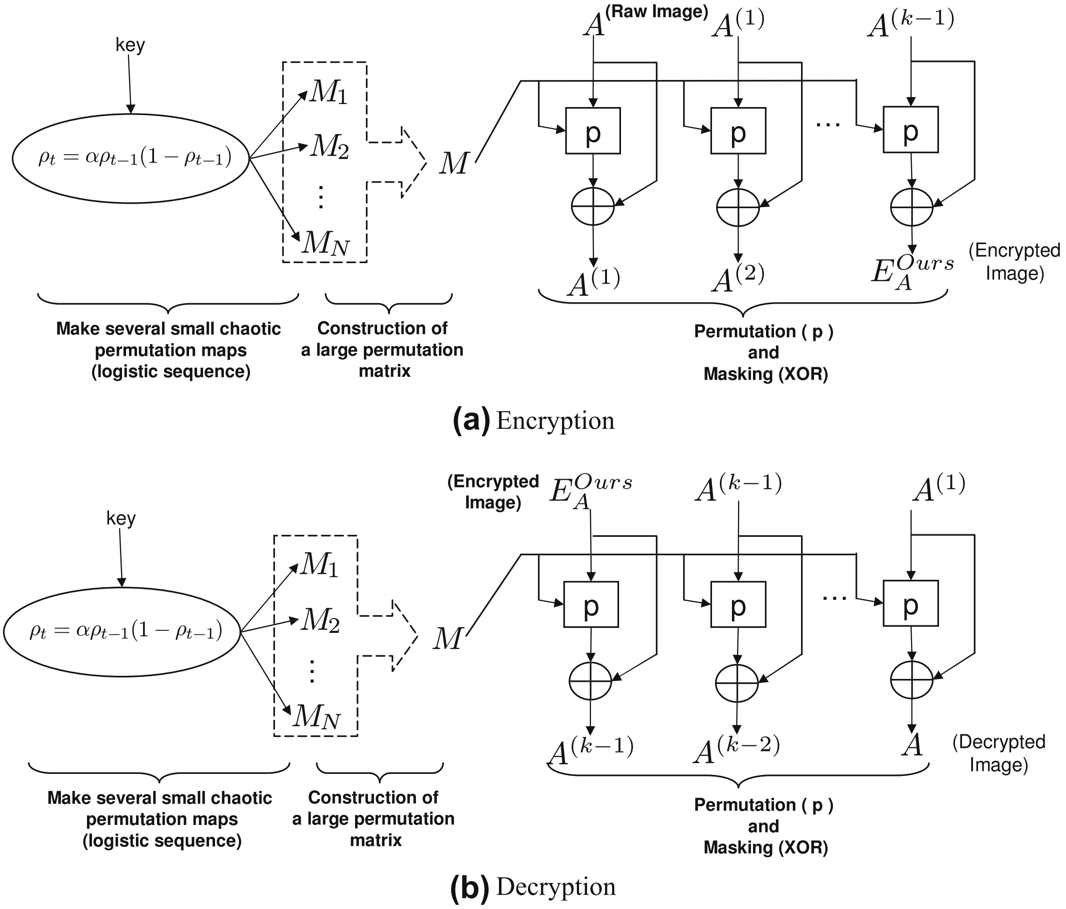

key
$\rho_t = \alpha\rho_{t-1}(1-\rho_{t-1})$
M_1
M_2
M_N
M
$A^{\text{(Raw Image)}}$
$A^{(1)}$
$A^{(k-1)}$
p
$A^{(1)}$
$A^{(2)}$
E_A^{Ours}
(Encrypted Image)
Make several small chaotic permutation maps (logistic sequence)
Construction of a large permutation matrix
Permutation (p) and Masking (XOR)
(a) Encryption
key
$\rho_t = \alpha\rho_{t-1}(1-\rho_{t-1})$
M_1
M_2
M_N
M
(Encrypted Image)
E_A^{Ours}
$A^{(k-1)}$
$A^{(1)}$
p
$A^{(k-1)}$
$A^{(k-2)}$
A
(Decrypted Image)
Make several small chaotic permutation maps (logistic sequence)
Construction of a large permutation matrix
Permutation (p) and Masking (XOR)
(b) Decryption

Fig. 1. Overall framework for encryption and decryption.

encryption and decryption, there is a common process to generate a large permutation matrix M built by combining several small permutation matrices, which are nonlinearly generated with a chaotic map. Basically, implementation of the proposed encryption scheme is freely independent of a specific chaotic map. Any chaotic maps that produce a pseudorandom permutation can be used in the proposed scheme. For a reference implementation, we design the encryption scheme based on a Logistic map [16]. We summarize the process of the proposed encryption algorithm as follows:

1. *Key generation:* Select initial conditions of a Logistic map and the size of each small matrix.
2. *Small matrices generation:* Compute the values of elements in the small matrices with the Logistic map.
3. *Permutation matrix construction:* Construct a large permutation matrix from the small matrices.
4. *Permutation:* Permute a plaintext image with the constructed permutation matrix.
5. *Masking:* Mask the permuted image in the step 4 using the permutation matrix.

For a given security parameter k, the steps 4 and 5 are recursively repeated k times. Let $A^{(i)}$ be the encrypted image for the i round. For each round, $A^{(i-1)}$ is used as the input of the algorithm to generate the encrypted image $A^{(i)}$. $A^{(0)}$ is defined as a plaintext image. This process is recursively performed for k rounds. The more rounds are processed, the more secure the encryption is, but at the expense of computations and time delays. The decryption process operates similarly by applying the inverse of all the transformations described in reverse order as shown in Fig. 1(b).

3. Construction of permutation matrix

3.1. Definitions

PM is an acronym of Permutation Matrix and we use PM in this paper rather than PA (Permutation Array). C is a PM over $\Re$ of size n and then C is $n \times n$ matrix. S_n denotes the set of all $n!$ permutations with n distinct elements of some fixed set R. For example, for $n = 3$ and $R = \{0, 1, 2\}$, $S_3 = \{012, 021, 120, 102, 201, 210\}$. (n,d) PM stands for a subset of S_n with the property that the Hamming distance between any two distinct permutations in the array is at least d. Particularly, $(n, n-1)$ PM is equivalent to circular arrays $(d = n-1)$. C is r-bounded if no element of R appears more than r times in any column of C and C is r-balanced if each element of R appears exactly r times in each column of C. In addition, we say that C is r-separable if it is the disjoint union of $r(n, n-1)$ PM of size n.

3.2. Generation of small matrices with chaotic maps

The size of the small matrices is defined by a key. Initially, each column of the small matrix M_i (size $m_i \times m_i$) is set to $\{1, 2, \cdots, m_i\}$. We randomly shuffled the values of M_i by using a Logistic map, which is similar to Knuth shuffling [17]. This process is written in the following form:

$$\begin{aligned} &\rho_t = \alpha\rho_{t-1}(1-\rho_{t-1}), \\ &M_i(t) \Longleftrightarrow M_i(mod(\rho_t \times 10^3, m_i)), \end{aligned} \tag{1}$$

where α is set to 3.9999 in this paper. Here, M_i is a matrix with size $m_i \times m_i$ but we can access a value of the matrix with a lattice index (an integer) after regarding as a single block. $M_i(a) \Longleftrightarrow M_i(b)$ represents an exchange operation of two values at the ath and the bth positions of the small matrix M_i.

3.3. Construction of permutation matrix with small matrices

r-Bounded $(m_1m_2, m_1m_2 - 1)$ PM is built by combining r-bounded $(m_1, m_1 - 1)$ PM and s-separable $(m_2, m_2 - 1)$ PM. Also, r-balanced $(m_1, m_1 - 1)$ PM and s-separable $(m_2, m_2 - 1)$ PM construct r-balanced $(m_1m_2, m_1m_2 - 1)$ PM [18]. Especially, we use 1-balanced $(m_1, m_1 - 1)$ PM in order that PM is orthogonal and magic square. That is, the matrix combination is defined by

$$C \times T = \begin{bmatrix} T_{f(1,1)} + m_2c_{1,1}J & \dots & T_{f(1,m_1)} + m_2c_{1,m_1}J \\ \vdots & \ddots & \vdots \\ T_{f(m_1,1)} + m_2c_{m_1,1}J & \dots & T_{f(m_1,m_1)} + m_2c_{m_1,m_1}J \end{bmatrix}, \tag{2}$$

where C is an r-balanced $(m_1, m_1 - 1)$ PM and T is a s-separable $(m_2, m_2 - 1)$ PM. Here, J is a matrix of which elements have values 1 and $c_{i,j}$ denotes an element in ith row and jth column of C and when $c_{i,j} = \alpha$ and the α appears t times in the jth column, we have $f(i,j) = t$. For example, let C be an 1-balanced (2,1) PM and T be a 2-separable (3,2) PM such as the following:

$$C=\begin{bmatrix}0&1\\1&0\end{bmatrix},\quad T_1=\begin{bmatrix}0&1&2\\1&2&0\\2&0&1\end{bmatrix},\quad T_2=\begin{bmatrix}0&2&1\\2&1&0\\1&0&2\end{bmatrix}.$$

Here, $T=T_1\cup T_2$ and by Eq. (2) the constructed matrix is

$$C\times T=\begin{bmatrix}T_1 & T_1+3J\\ T_1+3J & T_1\end{bmatrix}=\begin{bmatrix}0&1&2&3&4&5\\1&2&0&4&5&3\\2&0&1&5&3&4\\3&4&5&0&1&2\\4&5&3&1&2&0\\5&3&4&2&0&1\end{bmatrix}.$$

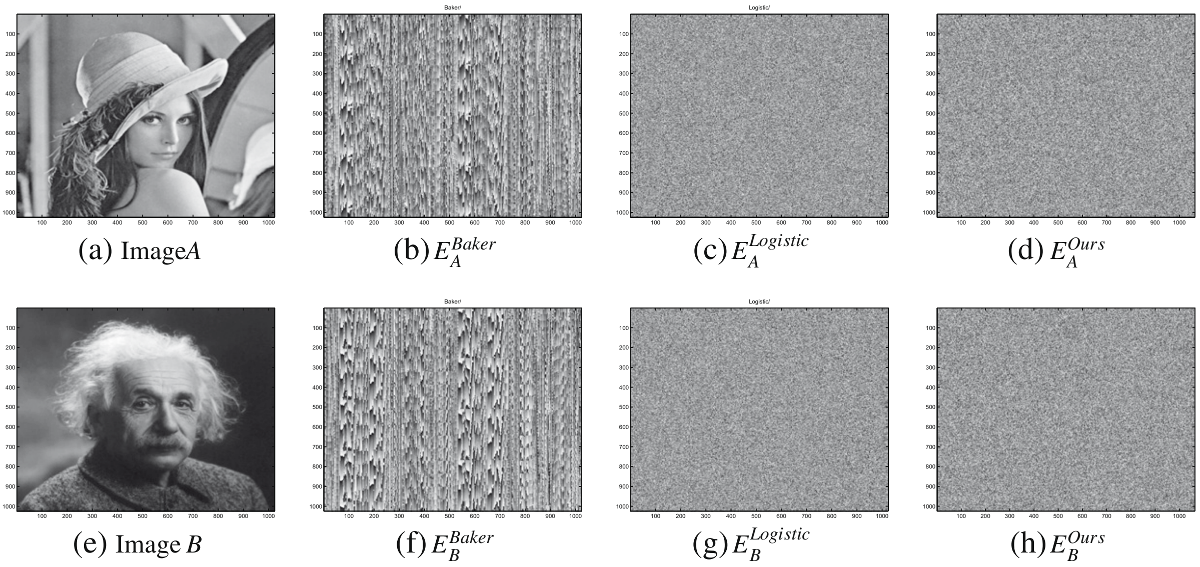
As combining two 1-balanced PMs results in an 1-balanced PM, combination of N 1-balanced PMs can build an 1-balanced PM as described in Eq. (2). Thus, if m_i is the size of an ith small PM, the constructed PM has the size of $m\left(=\prod_{i=1}^{N}m_i\right)$. For example, a large PM of size 1200 may be generated by five small PMs which have sizes of 5, 5, 4, 4, and 3. The sequential combination is described as $(((M_5\times M_4)\times M_3)\times M_2)\times M_1\rightarrow M$ where M is an 1-balanced $(1200,1199)$ PM and M_i is also 1-balanced PM for $i\in\{1,2,\cdots,5\}$.

4. Security analysis

A secure encryption scheme should resist several types of cryptanalysis such as histogram analysis. We discuss the security analysis of the proposed encryption scheme by experimental tests with two gray images. The two images are called A and B. An image A is a jpeg format with 1024×1024 size and the other image B is a png format with 1024×1024 size. In these experiments we evaluate the security measures such as histogram and correlation coefficients. For comparison, we implement two image encryption schemes based on Baker map [5] and Logistic map [10]. We used the security parameter k to represent “the number of rounds” for Baker-based encryption and our encryption.

Fig. 2 shows the original images (a and e) and encrypted images (b, c, d, f, g and h) generated by using the Baker-based encryption ($k=2$), the Logistic-based encryption and our method ($k=2$). We name the encrypted images by E_I^{type} where ‘I’ denotes the image and ‘*type*’ represents the method, i.e. E_A^{Ours} represents an encrypted image of A by using our method. As we can see in the figures, the encrypted images by our proposed methods and the Logistic-based encryption look much more random than those by Baker-based encryption algorithm.

Also, we analyzed the effect of rounds on our encryption. Fig. 3 shows permuted images with several rounds by the Baker-based encryption and our encryption. In this figure, we can see that the Baker-based encryption cannot shuffle the images well in a small rounds.



(a) Image$A$ (b) E_A^{Baker} (c) $E_A^{Logistic}$ (d) E_A^{Ours}

(e) Image B (f) E_B^{Baker} (g) $E_B^{Logistic}$ (h) E_B^{Ours}

Fig. 2. Two images A and B and their encrypted images by using the Baker-based encryption ($k=2$), Logistic-based encryption and our encryption ($k=2$).

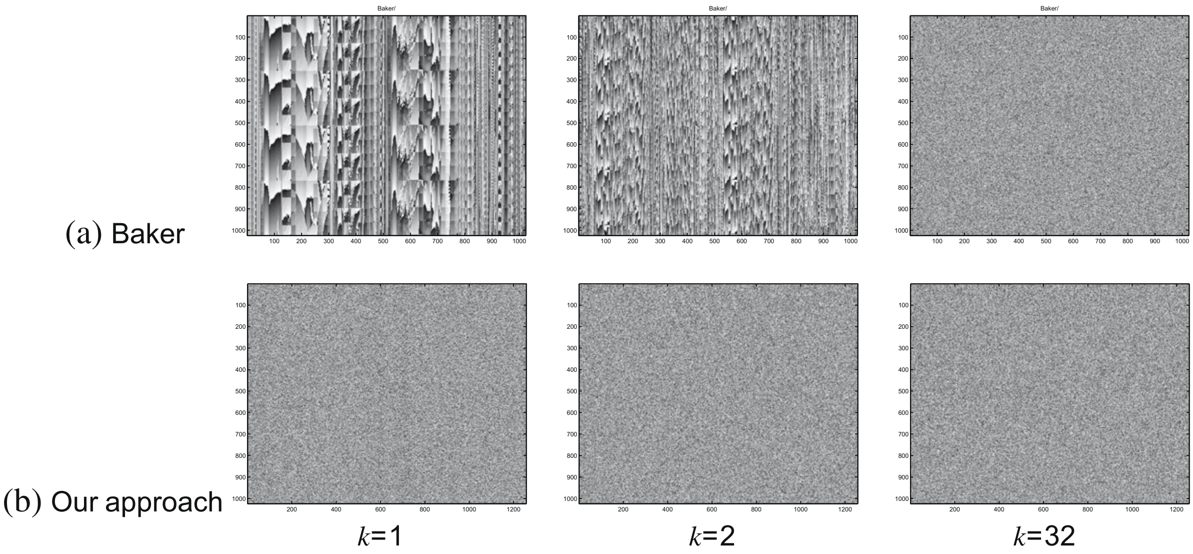


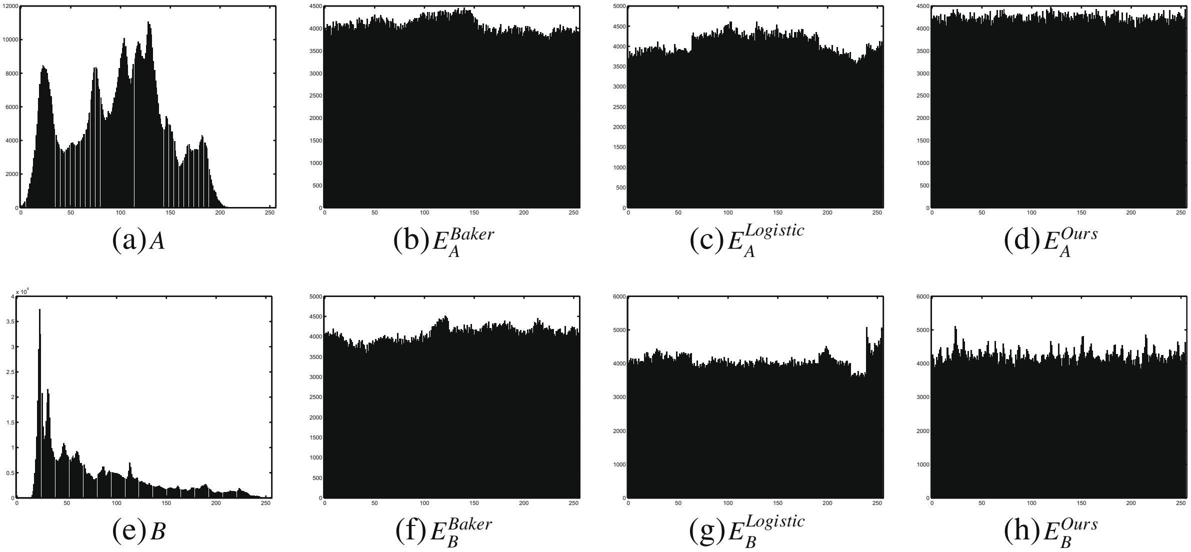
Fig. 3. Encrypted images of *A* with *k* rounds: Baker-based encryption (a), our encryption (b).

4.1. Histograms of encrypted images

Histograms of plaintext images and encrypted images are compared in Fig. 4. As can be seen in Fig. 4(a) and (e), particular values of pixels are dominant in the plaintext images and there are certain patterns. Our ideal goal is that encrypted images have histograms with random behavior. Fig. 4(b) and (f) are the encrypted images of *A* and *B* by using a Baker map. Fig. 4(c) and (g) are the encrypted images of *A* and *B* by using a Logistic map. The intensities of these encrypted images are distributed in all values among 0–255 but it is not close to the ideal uniform distribution. Whereas, the values of pixels of E_A^{Ours} and E_B^{Ours} of Fig. 4(d) and (h) are almost uniformly distributed between 0 and 255. Thus, our approach is more robust against statistical analysis such as histogram information than the Baker-based encryption [5] and the Logistic-based encryption [10].

4.2. Correlation between two adjacent pixels

To show the correlation between two adjacent pixels in encrypted images, we analyze the correlation between two vertically adjacent pixels, two horizontally adjacent pixels, and two diagonally adjacent pixels, respectively.



(a) A (b) E_A^{Baker} (c) $E_A^{Logistic}$ (d) E_A^{Ours}

(e) B (f) E_B^{Baker} (g) $E_B^{Logistic}$ (h) E_B^{Ours}

Fig. 4. Histograms of the original images and the encrypted images: (b and f) by Baker-based encryption ($k = 2$), (c and g) by Logistic-based encryption and (c and f) by our encryption ($k = 2$).

Table 1
Correlation coefficients of the encrypted images by the Baker-based encryption, the Logistic-based encryption and our approach ($\times 10^{-3}$).

Image	Measure	Vertical			Horizontal			Diagonal		
		Baker	Logistic	Ours	Baker	Logistic	Ours	Baker	Logistic	Ours
A	$k = 1$	923	−88	−61	605	−29	9	575	−21	5
	$k = 2$	910		0	233		13	204		−22
	$k = 5$	761		−2	−20		−60	232		−3
	$k = 10$	591		−15	−34		−7	−6		3
	$k = 16$	438		−18	9		−7	−19		15
	$k = 32$	127		−13	−14		−2	−1		−20
	$k = 64$	−16		−16	2		7	−15		17
B	$k = 1$	933	−55	−32	674	−12	3	649	−6	17
	$k = 2$	905		−12	287		−36	303		15
	$k = 5$	815		2	13		−42	21		25
	$k = 10$	647		−9	−19		7	−36		−7
	$k = 16$	463		−5	10		−9	1		−9
	$k = 32$	130		−5	−23		−7	2		−23
	$k = 64$	−13		−18	12		−4	13		18

Table 1 shows the correlation coefficients of the encrypted images of *A* and *B* by the Baker-based encryption, the Logistic-based encryption and our approach. For a reference data, we also generated a random image and we calculated correlation coefficients of the random image: 10×10^{-3} for vertical adjacency, -25×10^{-3} for horizontal adjacency and -14×10^{-3} for diagonal adjacency. For the Baker-based encryption and our approach, we analyzed changes of the correlation with variant rounds from $k = 1$ to $k = 64$. In this table, we can see that our approach and the Logistic-based encryption generate random-like encrypted images. However, the Baker-based encryption requires many rounds (such as over 64 rounds) to obtain sufficient random images as shown in Table 1.

We can represent these results visually in Fig. 5.[1] The Baker-based encryption cannot give sufficient randomness (shuffling) even with 32 rounds and the Logistic-based encryption also seems that the encrypted image has a sort of patterns in diagonal. Similarly to the random image, we cannot see any patterns in the encrypted images via our approach. Thus, our approach has superior performance of shuffling in all rounds compared to other two approaches.

4.3. Key sensitivity

In secure encryption schemes, high key sensitivity is generally required since they should prevent adaptive chosen-plaintext attacks, including differential and linear cryptanalysis. For key sensitivity tests, we partially change initial conditions of chaotic maps and analyze the effect of the difference of the ciphertext image. We used two different keys of which the third bit is different, *OXCAE7E4* and *OXAAE7E4*. We compared the key sensitivity of the proposed algorithm with the Baker-based and the Logistic-based encryption as shown in Table 2.

The Baker-based encryption scheme has poor performance on the key sensitivity as already referred to in [5]. However, Logistic-based encryption and our proposed approach have high key sensitivity.

4.4. FIPS 140 testing

We also show that our proposed algorithm pass the FIPS 140-2 randomness tests [19].

There are four tests: Monobit, Poker, Runs tests and Long run tests. Each of the tests were designed to test the randomness of a sample sequence length of 20,000 bits as follows:

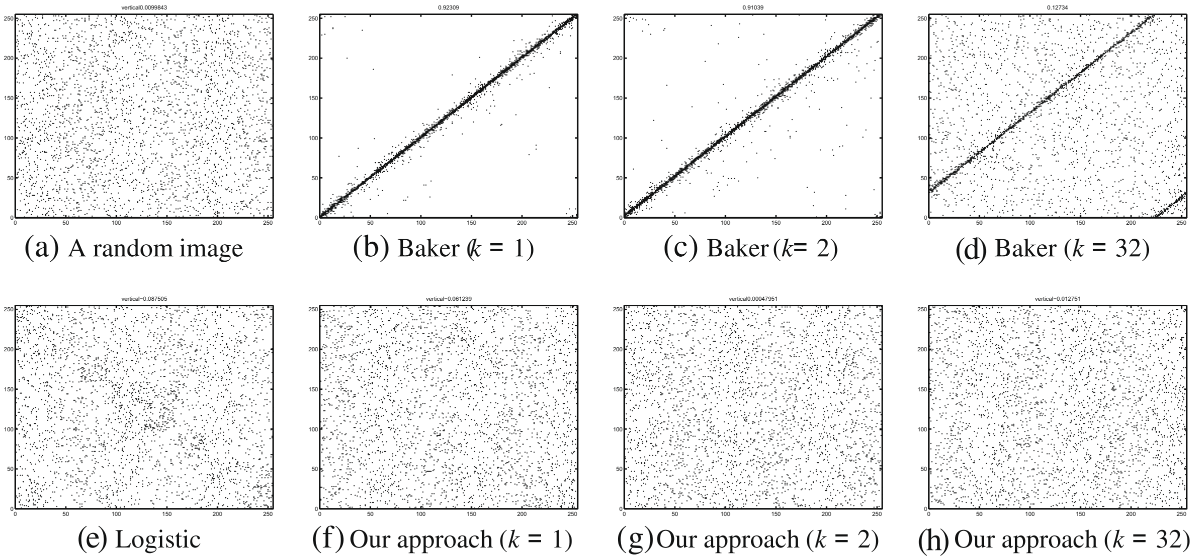
- The Monobit test
  1. Calculate x which is the number of ones in the 20,000 bit stream.
  2. The test is passed if $9725 < x < 10{,}275$.

- The Poker test
  1. Divide the 20,000 bit stream into 5000 contiguous 4 bit segments. Count and store the number of occurrences of each of the 16 possible 4 bit values. Denote $g(i)$ as the number of each 4 bit value i where 0–15.
  2. Calculate x by

$$x = \frac{16}{5000} \sum_{i=0}^{15} g(i)^2 - 5000.$$

  3. The test is passed if $2.16 < x < 46.17$.

[1] Given the space available, we only demonstrate the correlation between two vertically adjacent pixels.



(a) A random image (b) Baker (*k* = 1) (c) Baker (*k*= 2) (d) Baker (*k* = 32)

(e) Logistic (f) Our approach (*k* = 1) (g) Our approach (*k* = 2) (h) Our approach (*k* = 32)

Fig. 5. Correlation of the encrypted images of image *A*.

Table 2
Comparison of correlation coefficients for key sensitivity ($\times 10^{-3}$).

Image	Measure	Baker	Logistic	Ours
A	$k = 1$	86	14	−13
	$k = 2$	42		29
	$k = 5$	84		−23
	$k = 10$	103		−6
	$k = 16$	100		−26
	$k = 32$	70		−1
	$k = 64$	55		−11
B	$k = 1$	991	−1	−19
	$k = 2$	990		17
	$k = 5$	979		−19
	$k = 10$	963		15
	$k = 16$	938		−2
	$k = 32$	889		−8
	$k = 64$	770		−20

- The Runs test
  1. A run represents a maximal sequence of consecutive bits of either all ones or all zeros. The incidences of runs of all lengths in the sample stream should be counted and stored.
  2. The test is passed if the number of runs is each within the corresponding interval specified below.

Length of the run	1	2	3	4	5	⩾6
Required interval	2315–2685	1114–1386	527–723	240–384	103–209	103–209

- The Long run test
  1. Find the longest run in the 20,000 bits.
  2. If the length of the longest run in the bit stream of 20,000 bit (both of one and zero) is smaller than 26, the test is passed.

We need, however, to change the testing algorithm to suit to image data so we randomly chose 100 streams of 20,000 consecutive bits from the encrypted images of image *A*. Then we calculated statistics of the randomly chosen 100 streams for each test and compared them to the acceptance ranges.

Tables 3 and 4 show the numbers of the samples among 100 randomly chosen samples, which passed the Monobit, Poker, Long run tests and run tests. While almost all samples from the Logistic-based encryption and our approach pass the

Table 3
The number of samples passed by Monobit, Poker tests and Long run tests.

Methods	Number of rounds	Monobit test	Poker test	Long run test
Baker	$k = 1$	36	3	0
	$k = 2$	60	29	0
	$k = 10$	100	100	12
	$k = 16$	100	100	19
	$k = 32$	100	100	68
Logistic		100	100	100
Ours	$k = 1$	95	61	100
	$k = 2$	100	100	100
	$k = 10$	100	100	100
	$k = 16$	100	100	100
	$k = 32$	100	100	100

Table 4
The number of samples passed by Runs tests and the probability to pass the test for image *A*.

Method	Type of runs		The number of runs					
			1	2	3	4	5	⩾6
Baker	$k = 1$	0	0	0	0	2	40	0
		1	0	0	0	1	48	1
	$k = 2$	0	0	0	0	3	69	0
		1	0	0	1	2	71	0
	$k = 10$	0	0	100	56	64	90	1
		1	0	100	58	73	88	1
	$k = 16$	0	0	18	11	100	100	2
		1	0	23	10	100	100	2
	$k = 32$	0	2	82	84	100	100	7
		1	2	81	82	100	100	4
Logistic	0		99	100	100	100	100	99
	1		100	97	99	100	100	100
Ours	$k = 1$	0	100	100	100	100	100	100
		1	100	100	100	100	100	100
	$k = 2$	0	100	100	100	100	100	100
		1	100	100	100	100	100	100
	$k = 10$	0	100	100	100	100	100	100
		1	100	100	100	100	100	100
	$k = 16$	0	100	100	100	100	100	100
		1	100	100	100	100	100	100
	$k = 32$	0	100	100	100	100	100	100
		1	100	100	100	100	100	100

Monobit test and Poker test even in very small number of rounds, many samples from the Baker-based encryption cannot pass them with $k = 2$. In addition, the Baker-based encryption cannot pass the Long run test even in 32 rounds $(k = 32)$ but the Logistic-based encryption and our approach pass the test in all cases.

Table 4 shows the performance of the Runs test with the different length of the runs. As we can see in the table, our proposed encryption exhibits better randomness than other two approaches since our encryption passed all tests $(k = 1, k = 2, k = 10, k = 16 \text{ and } k = 32)$.

5. Conclusion

We proposed a new image encryption algorithm with a large pseudorandom permutation which is computed from chaotic maps combinatorially. Since a pseudorandom sequence is securely extended by the permutation matrix, the proposed encryption algorithm shows secure statistical information with relatively short pseudorandom sequences compared to other encryption algorithms. Therefore, we expect that initial conditions or parameters of chaos maps can be chosen with inexpensive cost unlike other chaos-based encryption schemes since the randomness properties of chaotic maps can be effectively spread into encrypted images by using the permutation matrix. By analyzing the statistical information of encrypted images in the experimental tests, we show that the proposed algorithm provides reasonable security against statistical cryptanalysis.

References

[1] National Institute of Standards and Technology (NIST). FIPS-197: Advanced encryption standard; November 2001. Available from: http://csrc.nist.gov/publications/PubsFIPS.html.

[2] Socek D, Magliveras S, C'ulibrk D, Marques O, Kalva H, Furht B. Digital video encryption algorithms based on correlation-preserving permutations. EURASIP J Inform Security 2007.
[3] Chang C, Hwang M, Chen T. A new encryption algorithm for image cryptosystems. J Syst Software 2001;58:83–91.
[4] Preneel B. Design principles for dedicated hash functions. In: Fast software encryption, Cambridge security workshop, Lecture notes in computer science, vol. 809, Springer, Berlin; 1993. p. 71–82.
[5] Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurcat Chaos 1998;8:1259–84.
[6] Zhang L, Liao X, Wang X. An image encryption approach based on chaotic maps. Chaos Soliton Fract 2005;24:759–65.
[7] Tong X, Cui M. Image encryption scheme based on 3D baker with dynamical compound chaotic sequence cipher generator. Signal Process 2009;89:480–91.
[8] Gao T, Chen Z. Image encryption based on a new total shuffling algorithm. Chaos Soliton Fract 2008;38:213–20.
[9] Chen G, Mao Y, Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Soliton Fract 2004;21(3):749–61.
[10] Kocarev L, Jakimoski G. Logistic map as a block encryption algorithm. Phys Lett A 2001;289(4–5):199–206.
[11] Pisarchik AN, Flores-Carmona NJ, Carpio-Valadez M. Encryption and decryption of images with chaotic map lattices. Chaos: Interdiscipl J Nonlinear Sci 2006;16(3):033118.
[12] Li P, Li Z, Halang WA, Chen G. A stream cipher based on a spatiotemporal chaotic system. Chaos Soliton Fract 2007;32(5):1867–76.
[13] Sun F, Liu S, Li Z, Lu Z. A novel image encryption scheme based on spatial chaos map. Chaos Soliton Fract 2008;38(3):631–40.
[14] Scharinger J. Fast encryption of image data using chaotic Kolmogrov flow. J Electron Eng 1998;7(2):318–25.
[15] Dyson FJ, Falk H. Period of a discrete cat mapping. Am Math Month 1992;99(7):603–14.
[16] May RM. Simple mathematical models with very complicated dynamics. Nature 1976;261.
[17] Knuth DE. The art of computer programming, third ed., vol. 2. Reading, MA: Addison-Wesley; 1997.
[18] Ding C, Fu F, Klove T, Wei VKW. Constructions of permutation arrays. IEEE Trans Inform Theory 2002;48:977–80.
[19] National Institute of Standards and Technology (NIST). FIPS pub 140-2: Security requirements for cryptographic modules; May 2001. Available from: http://csrc.nist.gov/publications/PubsFIPS.html.