Commun Nonlinear Sci Numer Simulat 15 (2010) 3507–3517

Contents lists available at ScienceDirect

Commun Nonlinear Sci Numer Simulat

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

A fast image encryption and authentication scheme based on chaotic maps

Huaqian Yang [a,b,*], Kwok-Wo Wong [b], Xiaofeng Liao [c], Wei Zhang [a], Pengcheng Wei [a]

[a] Department of Computer & Modern Education Technology, Chongqing Education of College, Chongqing 400067, China
[b] Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong
[c] College of Computer Science and Engineering, Chongqing University, Chongqing 400044, China

ARTICLE INFO

Article history:
Received 7 June 2009
Received in revised form 3 January 2010
Accepted 4 January 2010
Available online 11 January 2010

Keywords:
Image encryption
Chaotic map
Hash function
Message authentication code

ABSTRACT

In recent years, a variety of chaos-based image cryptosystems have been proposed. The key used for encryption/decryption is usually independent of the plain-image. To achieve a satisfactory level of security, at least two overall rounds of the substitution-diffusion process are required so that a change in any pixels of the plain-image spreads over the whole cipher-image. Moreover, the receiver is not able to determine whether the decrypted image is exactly the one sent. In this paper, a fast image encryption and authentication scheme is proposed. In particular, a keyed hash function is introduced to generate a 128-bit hash value from both the plain-image and the secret hash keys. The hash value plays the role of the key for encryption and decryption while the secret hash keys are used to authenticate the decrypted image. Simulation results show that satisfactory security performance is achieved in only one overall round. The speed efficiency is thus improved.

1. Introduction

In recent years, the number of image files transmitted through the Internet keeps increasing. As a result, the secure transmission of confidential digital images over public channels has become a common interest in both research and application fields. Traditional symmetric ciphers such as DES and AES are designed with good confusion and diffusion properties [1]. However, image encryption is different from text encryption due to some intrinsic properties of images such as bulk data volume and high pixel correlation. The major challenge in designing effective image encryption algorithms is that it is rather difficult to shuffle and diffuse image data efficiently by traditional cryptographic means.

In [2], Fridrich suggested that a chaos-based image encryption scheme should compose of the iteration of two processes: substitution and diffusion, which are found in traditional symmetric ciphers such as DES and AES. The substitution stage permutes all the pixels as a whole, without changing their values. In the diffusion stage, the pixel values are modified sequentially so that a tiny change in a pixel spreads out to as many pixels in the cipher-image as possible. To eliminate the correlation between adjacent pixels, multiple permutation rounds are performed in the substitution stage. The whole substitution–diffusion process repeats for a number of times in order to achieve a satisfactory level of security. To further enhance the security, the parameters of the chaotic maps governing the permutation and the diffusion should better be distinct in different rounds. Several chaos-based image cryptosystems were designed under this architecture. For example, Chen and his research group employed a three-dimensional (3D) cat map [3] and a 3D baker map [4] in the substitution stage. Guan et al. used a 2D cat map for pixel relocating and the discretized Chen's chaotic system for pixel value masking [5]. Lian et al. [6] used a chaotic standard map in the substitution stage and a quantized logistic map in the diffusion stage.

* Corresponding author. Address: Department of Computer & Modern Education Technology, Chongqing Education of College, Chongqing 400067, China. Tel.: +86 23 86059101.
E-mail address: mailtostorm@163.com (H. Yang).

1007-5704/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.cnsns.2010.01.004

In the above-mentioned image cryptosystems, the effect of confusion is contributed solely by the substitution stage while that of diffusion is only achieved in the diffusion stage. Wong et al. proposed a modified architecture for fast image encryption [7], as shown in Fig. 1. In this architecture, certain diffusion effect is introduced in the substitution stage by a simple sequential add-and-shift operation. As a result, satisfactory security performance is achieved in fewer overall rounds and the total encryption time is shortened [7].

Recently, some chaos-based image encryption algorithms with substitution-diffusion structure are cryptanalyzed [8–11]. The common flaws or deficiencies of these algorithms are summarized as follows:

(1) The key for encryption/decryption is independent of the plain-image and this favors known-plaintext and chosen-plaintext attacks.
(2) In the diffusion process, the change in the current pixel value only affects a few cipher-image pixels in one round and so at least two overall rounds of substitution-diffusion are required.
(3) It is difficult for the receiver to determine whether the decrypted image is exactly the one sent.

Usually, the common meaning of efficiency is denoted that the encryption/decryption speed is fast as soon as possible under keeping security of algorithm condition. To further enhance the security and the efficiency of chaos-based image cryptosystems, we propose a novel scheme for both encryption and authentication. A keyed hash function constructed by chaotic maps is used to generate a 128-bit hash value from the plain-image and the secret hash keys. This hash value is the encryption key. At the receiver, the same hash value is employed to decrypt the cipher-image while the hash keys are then used to authenticate the decrypted image. A tiny change in any plain-image pixels or the hash keys leads to a totally different encryption key. As a result, satisfactory security performance is achieved in only one encryption round and the operation efficiency is much improved. Experimental results indicate that the encryption/decryption key cannot be recovered if the cryptanalyst only possesses some portion of the plaintext but not the secret hash keys. Therefore, the proposed algorithm is secure and efficient.

The rest of this paper is organized as follows. In Section 2, the keyed hash function based on chaotic maps is introduced and its performance is analyzed. Section 3 is devoted to the description of the proposed fast image encryption and authentication algorithm. Performance and security of this scheme are analyzed in Section 4, together with some discussions. In Section 5, a conclusion is drawn.

2. Keyed hash function

2.1. Architecture

A novel keyed hash function shown in Fig. 2 is employed to generate the encryption/decryption key from the plain-image and the secret hash keys. Each block T_α is a 1D chaotic tent map defined by the following equation:

$$T_\alpha : x_j = \begin{cases} \frac{x_{j-1}}{\alpha}, & \text{if } 0 \leqslant x_{j-1} \leqslant \alpha \\ \frac{1-x_{j-1}}{1-\alpha}, & \text{if } \alpha < x_{j-1} \leqslant 1 \end{cases} \tag{1}$$

where $0 < \alpha < 1$. This function maps the interval $[0,1]$ onto itself with only one parameter α. A sequence formed by iterating T_α from an arbitrary initial point in $(0,1)$ exhibits chaotic properties [12–15] as T_α is expanding everywhere in the interval $(0,1)$.

The plain-image to be encrypted is a gray-scale one with each pixel represented by an 8-bit value. It is scanned by rows to form a 1D sequence P with length equal to the number of pixels in the plain-image. The sequence is first padded with zeros to make its length a multiple of 4. It is then divided into a number of blocks $P_0, P_1, P_2, \ldots, P_r \ldots$, each consists of 4 pixels. The hash value is generated as follows:

(1) Randomly choose the four secret hash keys within the range $(0,1)$. They become the initial values for x_0, x_1, x_2 and x_3.

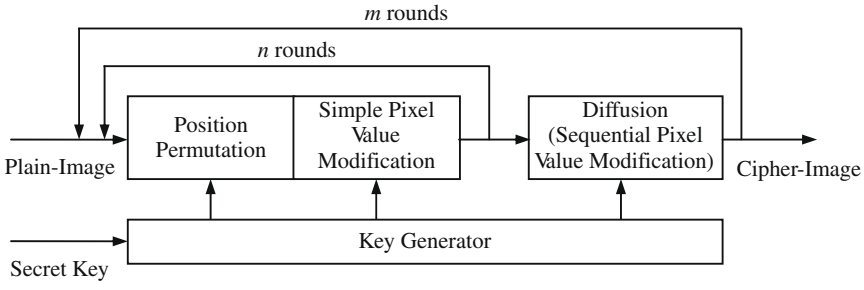

Fig. 1. Architecture of the chaos-based image encryption proposed by Wong et al.

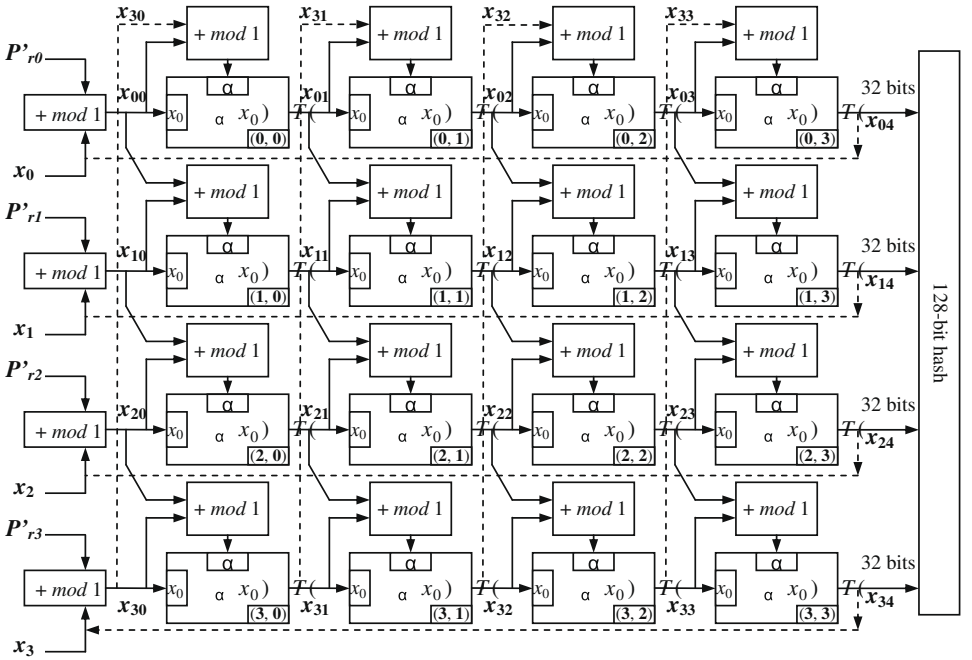

Fig. 2. Structure of the proposed hash function.

(2) Mix the pixels using the hash keys.

Firstly, each pixel value is mapped to a decimal number in the interval (0,1) by means of a linear transformation. Mathematically, for each pixel P_{ri} $(i = 0, 1, 2, 3)$ in the block P_r $(r = 0, 1, 2, \ldots)$,

$$P'_{ri} = (P_{ri} + 0.8)/256, \quad i = 0, 1, 2, 3 \tag{2}$$

Then each P'_{ri} is mixed with the hash keys according to the following equation:

$$x_{i0} \leftarrow (P'_{ri} + x_i) \bmod 1, \quad i = 0, 1, 2, 3 \tag{3}$$

(3) Iterate the tent maps

For the convenience of description, the tent map located at the ith row and jth column of the architecture shown in Fig. 2 is denoted as $T_\alpha^{(i,j)}(x_0)$, where $0 \leqslant i, j \leqslant 3$. The initial value x_0 for iterating $T_\alpha^{(i,j)}(x_0)$ is equal to x_{ij}, i.e., output of the previous map. The tent map parameter α is determined by the outputs of the two neighboring maps using the formula $\alpha = (x_{ij} + x_{(i-1)j}) \bmod 1$. For $i = 0, \alpha = (x_{0j} + x_{3j}) \bmod 1$. The output of the map $T_\alpha^{(i,j)}(x_0)$ is $x_{i(j+1)}$. For each block P_r $(r = 0, 1, 2, \ldots)$, iterate the maps from top to bottom and then left to right. If P_r is not the last block of the sequence P, the output x_{i4} of the map $T_\alpha^{(i,3)}(x_0)$ in the last column will be feedback to become x_i. Then return to Step (2) for processing the next block P_{r+1}.

The cipher-block chaining (CBC) operation mode of the proposed keyed hash function causes x_{i4}, the output of the map $T_\alpha^{(i,3)}(x_0)$ in the last column, depending on the initial values, the current and all preceding plain-image pixels.

(4) Extract bits to form the final hash value

After all the blocks in P have been processed, convert the final values x_{04}, x_{14}, x_{24} and x_{34} to binary format and extract the first 32 bits after the decimal point [16,19] from each value to form four binary sequences h_0, h_1, h_2 and h_3. The 128-bit hash value h is obtained by concatenating them, i.e., $h = (h_0 h_1 h_2 h_3)$.

2.2. Performance analysis

In this section, the performance of the proposed hashing scheme is analyzed. The secret hash keys are randomly selected as $x_{h0} = 0.2467397419$, $x_{h1} = 0.4567943241$, $x_{h2} = 0.6349032456$ and $x_{h3} = 0.8325096845$. The initialization process is performed by setting $x_0 \leftarrow x_{h0}$, $x_1 \leftarrow x_{h1}$, $x_2 \leftarrow x_{h2}$ and $x_3 \leftarrow x_{h3}$. Then the proposed hashing scheme is applied to the following test cases:

Scenario 1: The 512×512 Lena image with 8-bit gray-scale, as shown in Fig. 3, is used.
Scenario 2: Add 1 to the value of the pixel located at the lower right corner.
Scenario 3: Subtract 1 from the value of the pixel located at the upper left corner.

Fig. 3. Original image Lena.

Scenario 4: Change the secret key $x_3 = 0.8325096845$ to 0.8325096846 and use the original image.

The corresponding hash values in hexadecimal form are:

Scenario 1: 2FE487D0C2CA919C7975333ADAC7FC15.
Scenario 2: A0D9EBC6F3F00F3CC7693DC799574020.
Scenario 3: E8C930AB8E31B9955AB1430076BB9B0F.
Scenario 4: 648C439FECAB9FA51933EF017E934F48.

If the hash value is expressed in binary format, each bit is either 0 or 1. Therefore, the ideal case is that any tiny changes in the initial condition of the control parameter or the plain-image lead to a 50% changing probability in each bit. Six statistical tests are widely used to evaluate the performance of the hash function quantitatively. They are minimum changed bit number $B_{\min}$, maximum changed bit number $B_{\max}$, mean changed bit number $\overline{B}$, mean changed probability P, standard variance of the changed bit number ΔB and standard variance of the change probability ΔP [17–19].

We have performed the following tests. Firstly, the 128-bit hash value of the standard Lena image is generated. Then a pixel is randomly selected and its value is incremented by 1. A new hash value is computed to compare with the original one. The results of M = 512, 2048, 10,000, respectively, are listed in Table 1, where M is the numbers of test cases. They show that both the mean changed bit number $\overline{B}$ and the mean changed probability P are very close to the ideal values (64-bit and 50%, respectively). Moreover, the small values of ΔB and ΔP indicate that the performance of the hashing scheme is very stable. As a slight difference in the plain-image causes substantial changes in the hash value, the latter can be used in the design of an image encryption scheme that is very sensitive to the plain-image.

Collision resistance is essentially a measure of the probability that two random inputs hash to the same value. In the proposed keyed hash function, CBC mode is introduced to ensure that each tent map is influenced by others. This structure, together with the inherent sensitivity to initial value and the mixing property of chaotic map, expedites the avalanche effect in the hashing process and guarantees that each bit of the final hash value is related to all the image pixels.

The performance on collision resistance of our keyed hash function can be measured quantitatively by the following method. Firstly, the 128-bit hash value of the standard Lena image is computed and is expressed as a 16-byte sequence. Then a pixel is randomly selected and its value is incremented by 1. A new hash value is computed to compare with the original one. The number of identical bytes w at the same position of the two hash values is counted. This kind of measurement is performed independently for 10,000 times. It is found that there are 9390 cases have zero identical byte at the same position, denoted as $W_{10000}(0) = 9390$. The other results are $W_{10000}(1) = 590$, $W_{10000}(2) = 19$, $W_{10000}(3) = 1$, $W_{10000}(w) = 0$ for w = 4,5,...,16, as listed in Table 2. The theoretical number of identical bytes at the same position of the hash value in M independent tests is given by:

$$W_M(w) = M \times \text{Prob}\{w\} = M\frac{s!}{w!(s-w)!}\left(\frac{1}{2^k}\right)^w\left(1-\frac{1}{2^k}\right)^{s-w} \tag{4}$$

Table 1
Statistical performance of the proposed hash function.

	M = 512	M = 2048	M = 10,000
$B_{\min}$	59	44	42
$B_{\max}$	80	83	83
$\overline{B}$	64.06	64.01	63.94
P (%)	50.05	50.00	49.95
ΔB	5.93	5.66	5.68
ΔP (%)	4.63	4.43	4.43

Table 2
Distribution of the number of identical bytes at the same location in 10,000 hash values.

w	0	1	2	3	>3
$W_{10000}(w)$	9390	590	19	1	0

where $k = 8$, $s = 128/k = 16$, $w = 0, 1, \ldots, s$. It can be observed that our results are very close to the theoretical values [18], i.e., $W_M(0) = 9392.98$, $W_M(1) = 589.36$, $W_M(2) = 17.33$, $W_M(3) = 0.32$, $W_M(4) = 0.0041, \ldots, W_M(16) = 2.94 \times 10^{-39}$ when M = 10,000. This shows that the proposed keyed hash scheme has good statistical properties and strong collision resistance.

The performance of some chaos-based hash algorithms proposed recently is tabulated in Table 3 [18,19], together with that of MD5. As SHA-1 produces a 160-bit hash value, it is not compared with these algorithms. The results show that the hash algorithms studied in [18,19] and the proposed one have similar performance, which is close to ideal and can resist statistical attacks. However, their computational complexities are different. The algorithm proposed in [18] is based on spatiotemporal chaos which requires about 15L iterations, where L is the message length, in software implementation. To maintain the independence of the distribution of each lattice value, 45 iterations are required in each step of the scheme investigated in [19]. This makes it the slowest. Our scheme only needs 4L iterations and so is the most efficient.

3. The proposed image encryption and authentication scheme

3.1. Permutation

In [2], three 2D chaotic maps, namely, the standard map, the cat map and the baker map, are recommended to shuffle the pixels for confusion purpose. However, Lian et al. [6] pointed out that there exist some weak keys for ciphers employing the cat and the baker maps. Moreover, the key space of these two maps is not as large as that of the standard map. Therefore, they suggested using a standard map for permutation while keeping the logistic map for diffusion. In our scheme, the standard map is employed to permute the image pixels.

As the corner pixel (s = 0, t = 0) is not permuted at all under the standard map, a random scan couple (r_s, r_t) is included to move this corner pixel together with other pixels. The modified standard map equations are given by the following equation:

$$\begin{cases} s_{k+1} = (s_k + t_k + r_s + r_t) \bmod N \\ t_{k+1} = \left(t_k + r_t + K_c \sin\frac{2\pi \cdot s_{k+1}}{N}\right) \bmod N \end{cases} \tag{5}$$

where (s_k, t_k) and (s_{k+1}, t_{k+1}) are the original and the permuted pixel position of an $N \times N$ image, respectively. The standard map parameter K_c is a positive integer.

3.2. Diffusion

To avoid known-plaintext and chosen-plaintext attacks, the pixel values are altered sequentially in the diffusion process so that the change made to a particular pixel depends on the accumulated effect of all the previous pixel values. The logistic map employed in this stage is defined as

$$l_{k+1} = \mu l_k (1 - l_k) \tag{6}$$

where $k = 0, 1, 2, \ldots$; μ and l_0 are the control parameter and the initial value, respectively. In particular, μ should be in the range of [3.9,4] to ensure good chaotic properties.

Details of the diffusion operation are described below:

(1) Iterate Eq. (6) to obtain the state value l_k at discrete time k. If it is within the interval (0.2, 0.8), go to the next step; otherwise, continue the iteration until this condition is satisfied.

Table 3
Statistical performance of the algorithm in [18,19] and MD5 [19].

	M = 512			M = 2048			M = 10,000		
	[18]	[19]	MD5	[18]	[19]	MD5	[18]	[19]	MD5
$B_{\min}$	44	44	49	46	47	44	44	45	42
$B_{\max}$	81	81	82	80	83	86	85	83	83
$\overline{B}$	63.98	63.92	63.92	63.91	63.98	64.03	63.96	63.90	64.05
P (%)	49.98	49.94	49.93	49.92	49.98	50.02	49.97	49.91	50.04
ΔB	5.53	5.62	5.78	5.58	5.53	5.66	5.52	5.58	5.68
ΔP (%)	4.33	4.39	4.36	4.36	4.33	4.42	4.32	4.36	4.44

(2) Generate $\Phi(k)$ from l_k using the following equation:

$$\Phi(k) \leftarrow Bin2Int(b_{11}b_{12\ldots}b_{18}) \tag{7}$$

where l_k is represented in binary format $0.b_1b_2b_{3\ldots}b_{51}b_{52}, b_i \in \{0,1\}$. $b_{11}b_{12,\ldots}b_{18}$ represent the 11th to 18th bits after the decimal point. The IEEE 754 double precision floating-point format possesses 64-bit word length with a 52-bit fraction part, but only the 11th to 18th bits after the decimal point are chosen.

(3) The cipher-pixel value is calculated from the value of the currently operated and the previously operated pixels, according to Eq. (8):

$$C(k) = \Phi(k) \oplus \{(P(k) + 2 \cdot \Phi(k)) \bmod G\} \oplus C(k-1) \tag{8}$$

where $P(k)$ and $C(k)$ are the currently operated plain-image pixel and the cipher-image pixel, respectively. $C(k-1)$ is the previous cipher-image pixel. $C(0)$ is a secret initial value derived from the key, as defined in Section 3.3, G is the total number of possible gray scales in the plain-image. The inverse form of Eq. (8) for decryption is given by

$$P(k) = \{\Phi(k) \oplus C(k) \oplus C(k-1) + G - 2 \cdot \Phi(k)\} \bmod G \tag{9}$$

(4) Return to Step (1) until all plain-image pixels are processed.

3.3. Encryption

The architecture of the proposed cipher is shown in Fig. 4.

Step 1: Choose the secret keys x_{h0}, x_{h1}, x_{h2} and x_{h3} randomly to generate the 128-bit hash according to the approach described in Section 2.1.

Step 2: Determine the values of the secret parameters K_c, r_s, r_t, l_0 and $C(0)$ from the 128-bit hash code $(b_1b_{2\ldots}b_{24\ldots}b_{128})$ using the following bit assignment:

$$K_c \leftarrow Bin2Int(b_1b_{2\ldots}b_{24});$$
$$r_s \leftarrow Bin2Int(b_{25}b_{26\ldots}b_{40});$$
$$r_t \leftarrow Bin2Int(b_{41}b_{42\ldots}b_{56});$$
$$x_{01} \leftarrow (0.b_{57}b_{58\ldots}b_{88})_2;$$
$$x_{02} \leftarrow (0.b_{89}b_{90\ldots}b_{120})_2;$$
$$l_0 \leftarrow Bin2Dec((x_{01} + x_{02}) \bmod 1);$$
$$C(0) \leftarrow Bin2Int(b_{121}b_{122\ldots}b_{128}).$$

The function $Bin2Int(\cdot)$ transforms a binary number to an integer while $Bin2Dec(\cdot)$ transforms a binary number to a decimal value.

Step 3: Permute the plain-image pixels using the modified standard map given by Eq. (5) and diffuse the shuffled pixels using the scheme described in Section 3.2. It should be noticed that the permutation and diffusion processes are performed simultaneously in a single scan of plain-image pixels. The value is altered while relocating a pixel [7].

3.4. Decryption

As the permutation and diffusion are performed simultaneously in a single scan of plain-image pixels, the decryption procedure is slightly different from the encryption one [7]. Details are described as follows:

Step 1: From the 128-bit hash received secretly from the sender, determine the values of the parameters K_c, r_s, r_t, l_0 and $C(0)$ using the same bit assignment stated in Section 3.3.

Step 2: Permute the pixels of the cipher-image reversely to obtain an intermediate image.

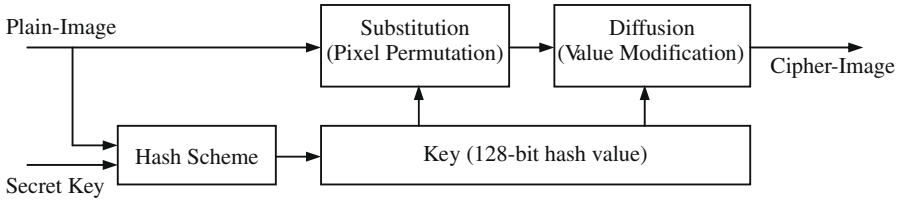

Fig. 4. Architecture of the proposed chaos-based image cryptosystem.

Step 3: Perform the reverse operations in the intermediate image to remove the effect of diffusion. The detail operations are the same as those described in Section 3.3, except that Eq. (8) is replaced by Eq. (9).

4. Performance and security analysis

As the chaotic region of the logistic map parameter μ is quite narrow, it is not selected as a key and is fixed at 4. The image for testing is the standard 512×512 Lena image with 8-bit gray-scale. The initial hash keys are chosen randomly as $x_{h0} = 0.2467397419$, $x_{h1} = 0.4567943241$, $x_{h2} = 0.6349032456$ and $x_{h3} = 0.8325096845$.

4.1. Key space analysis

The key space of any encryption algorithms should be sufficiently large to make brute-force search infeasible. In the proposed image encryption and authentication algorithm, the hash keys x_{h0}, x_{h1}, x_{h2} and x_{h3} are solely used to generate the 128-bit hash which is then employed for encryption and decryption. Hence the key space is $2^{128} \approx 3.4028 \times 10^{38}$.

Besides encryption, the proposed image encryption and authentication scheme possesses the authentication function as well. If this feature is required at the receiver, the key space is composed of x_{h0}, x_{h1}, x_{h2} and x_{h3}, together with the 128-bit hash code. If the state of all chaotic maps is represented by the IEEE 754 double precision floating-point standard, the key space is much larger than 2^{128}.

4.2. Key sensitivity test

To evaluate the key sensitivity of our algorithm, the secret value x_{h3} is changed from 0.8325096845 to 0.8325096846, denoted as x'_{h3}, and the encryption is repeated. The two corresponding cipher-images are compared and a 99.62% difference in pixel values is found. The results are depicted in Fig. 5, which show that our proposed algorithm is sensitive to the key even for a difference as small as 10^{-10}. In [21], the object can still be recognized in the encrypted image if the change in the initial condition is $\Delta y_0 = 10^{-10}$. It cannot be distinguished only when $\Delta y_0 = 10^{-9}$ and three encryption rounds are performed. The key sensitivity of our algorithm is higher than that of the cryptosystem proposed in [21].

4.3. Differential attack

Differential attack would become ineffective if a tiny change in the plain-image causes a significant difference in the cipher-image. To measure this capability quantitatively, the following measures are usually used: number of pixels change rate (NPCR) and unified average changing intensity (UACI). They are defined as follows [3]:

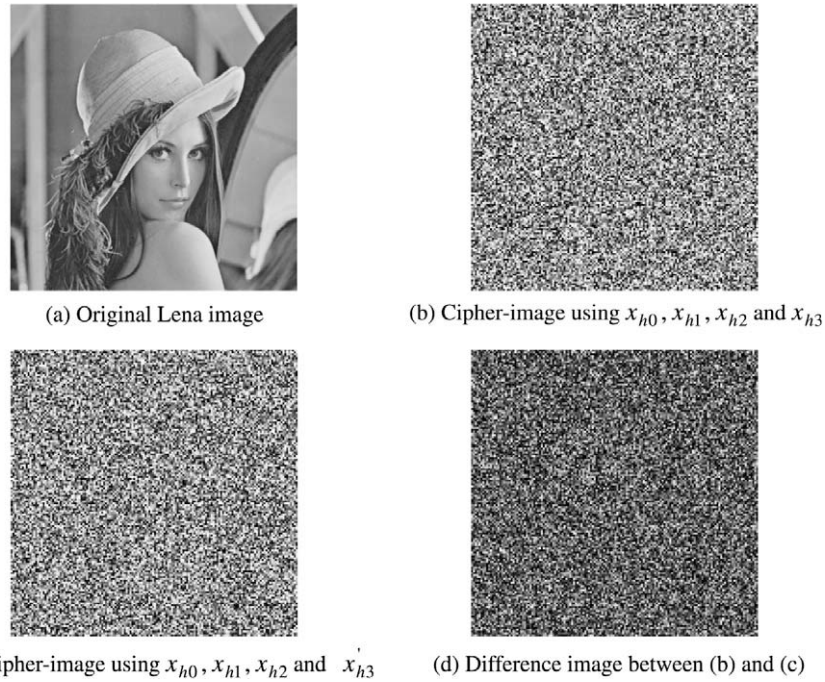

Fig. 5. Key sensitivity test.

$$\mathrm{NPCR} = \frac{\sum_{r,c} D(r,c)}{W \times H} \times 100\% \tag{10}$$

$$\mathrm{UACI} = \frac{1}{W \times H}\left[\sum_{r,c} \frac{|C_1(r,c) - C_2(r,c)|}{255}\right] \times 100\% \tag{11}$$

where C_1 and C_2 are the two cipher-images whose corresponding plain-images have only one-pixel difference, the gray-scale values of the pixels at position (r,c) of C_1 and C_2 are denoted as $C_1(r,c)$ and $C_2(r,c)$, respectively. W and H are the width and height of the cipher-image, respectively. The difference array $D(r, c)$ is determined by the following rule: if $C_1(r,c) = C_2(r,c)$ then $D(r,c) = 0$; otherwise it is 1. The values of these two quantitative measures (NPCR and UACI) for our algorithm are listed in Table 4.

Simulation results show that to achieve a similar performance of Lian et al.'s recommended cryptosystem [6], the proposed scheme only requires one overall round with one substitution and one diffusion, i.e., $m = 1$ and $n = 1$. Moreover, the substitution and diffusion processes are performed simultaneously. For comparison, Wong et al.'s scheme [7] needs one overall round with three permutation rounds in the confusion stage, i.e., $m = 1$ and $n = 3$, Lian et al.'s algorithm requires four overall rounds, i.e., $m = 4$ and $n = 4$. To achieve a higher performance such as NPCR > 0.996 and UACI > 0.334, Wong et al.'s and Lian et al.'s requirements are $m = 2$, $n = 2$ and $m = 6$, $n = 3$, respectively. However, $m = 1$ and $n = 1$ is sufficient in our cryptosystem.

4.4. Statistical analysis

According to Shannon's theory, a secure cryptographic scheme should be strong enough to resist statistical attack. To frustrate the powerful attacks based on statistical analysis, he suggested that diffusion and confusion should be employed in the cryptosystem [20]. In the proposed cryptosystem, a standard map is used to shuffle the image pixels, which can be considered as a confusion process. A logistic map is employed to generate a chaotic sequence with pseudorandom properties. Then this chaotic sequence is used to change the image pixel values sequentially. Thus, the diffusion effect is introduced.

The following measures [3] show that the confusion and diffusion properties are preserved in the proposed image encryption and authentication scheme.

(1) Histograms of the plain-image and the cipher-image

Histograms of the plain-image and the cipher-image are shown in Fig. 6. The latter histogram is fairly uniform and does not reveal any statistical information of the former.

(2) Correlation of two adjacent pixels

The high correlation of adjacent pixels can be utilized to carry out cryptanalysis. Therefore, a secure encryption scheme should remove the correlation between adjacent image pixels in order to improve the resistance against statistical analysis. The following procedures are carried out to test the correlation between two adjacent pixels in vertical, horizontal and diag-

Table 4
NPCR and UACI of the proposed, Wong et al.'s and Lian et al.'s cryptosystems (test image: Lena, 512 × 512).

(m,n)	NPCR			UACI		
	Proposed	Wong et al. [7]	Lian et al. [6]	Proposed	Wong et al. [7]	Lian et al. [6]
(1,1)	**0.996185**	–	–	**0.334795**	–	–
(1,2)	–	0.686642	0.000179	–	0.208793	0.000040
(1,3)	–	0.994370	0.000252	–	0.328191	0.000061
(2,2)	–	**0.996086**	0.009903	–	**0.334273**	0.002623
(4,4)	–	0.996143	0.992676	–	0.334972	0.317068
(6,3)	–	0.996181	**0.995865**	–	0.334865	**0.334197**

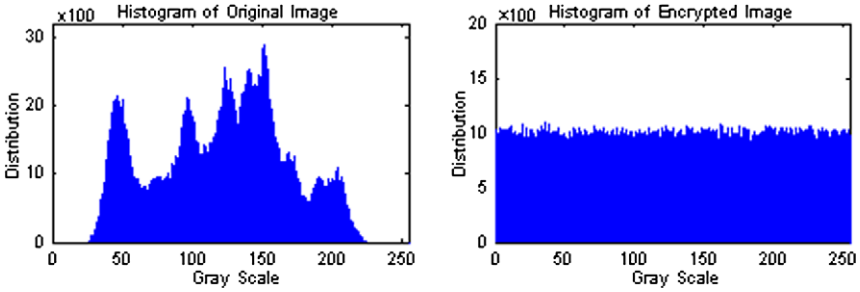

Fig. 6. Histograms of plain-image and cipher-image.

onal directions of a cipher-image, respectively. First, randomly select R pairs of two adjacent image pixels in the corresponding direction. Then, calculate the correlation coefficient of each pair using the following formula [3]:

$$\mathrm{cov}(x,y) = E[(x - E(x))(y - E(y))] = \frac{1}{R}\sum_{i=1}^{R}[(x_i - E(x))(y_i - E(y))] \tag{12}$$

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\cdot\sqrt{D(y)}} \tag{13}$$

where x and y are gray-scale values of two adjacent pixels in the image. $E(x) = \frac{1}{R}\sum_{i=1}^{R} x_i$ and $D(x) = \frac{1}{R}\sum_{i=1}^{R}(x_i - E(x))^2$. In each test, 10,000 pairs of adjacent pixels are selected, i.e., R = 10,000. The correlation distributions of two horizontally adjacent pixels in the plain-image and the cipher-image are shown in Fig. 7. The correlation coefficients are 0.980223 and −0.002097, respectively. Other test results are also listed in Table 5.

The measured correlation coefficients of the plain-image are close to 1 while those of the cipher-image are nearly 0. This indicates that the proposed algorithm has successfully removed the correlation of adjacent pixels in the plain-image so that neighbor pixels in the cipher-image virtually have no correlation.

4.5. Computational complexity and execution time

Apart from security consideration, other issues of an image cryptosystem such as the operation speed are significant, especially for real-time applications. The actual execution time of an algorithm is determined by many factors such as programming skill, programming language, execution environment, etc. Therefore, we discuss the performance of the proposed scheme from both the computational complexity and the measured execution time.

(1) Computational complexity

It is found from Table 4 that, to achieve the satisfactory performance of NPCR > 0.996 and UACI > 0.334, the proposed scheme only requires one overall round, i.e., m = 1, n = 1, Lian et al.'s requirement is m = 6, n = 3, and Wong et al.'s needs m = 2, n = 2. The numbers of operations, in terms of the image width N, required by different schemes are listed in Table 6. They indicate that the proposed algorithm and Wong et al.'s [7] have the same computational complexity while Lian

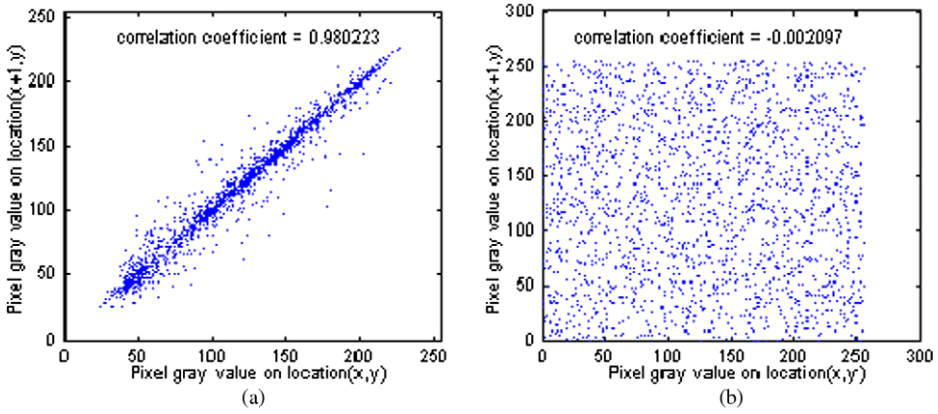

Fig. 7. Correlation of two horizontally adjacent pixels in (a) the plain-image and (b) the cipher-image.

Table 5
Correlation coefficients of two adjacent pixels in the image.

	Plain-image	Cipher-image
Horizontal	0.980223	−0.002097
Vertical	0.986639	−0.016187
Diagonal	0.964683	0.017805

Table 6
The required number of operations to achieve NPCR > 0.996 and UACI > 0.334.

	Hashing (tent map)	Substitution (standard map)	Diffusion (logistic map)	Total
Proposed	$4N^2$	N^2	N^2	$6N^2$
Lian et al. [6]	–	$18N^2$	$6N^2$	$24N^2$
Wong et al. [7]	–	$4N^2$	$2N^2$	$6N^2$

Table 7
Execution time of the proposed, Wong et al.'s and Lian et al.'s cryptosystems for some selected combinations of m and n (test image: Lena, 512×512).

(m,n)	Encryption time (ms)			Decryption time (ms)		
	Proposed	Wong et al. [7]	Lian et al. [6]	Proposed	Wong et al. [7]	Lian et al. [6]
(1,1)	**31.27**	–	–	**34.19**	–	–
(1,2)	–	15.65	14.80	–	16.95	15.35
(1,3)	–	20.79	19.39	–	22.37	19.78
(2,2)	–	**31.86**	30.12	–	**34.77**	30.77
(4,4)	–	102.85	95.81	–	108.65	97.90
(6,3)	–	124.40	**116.33**	–	134.18	**119.53**

et al.'s [6] is much higher. Some simple operations such as bit-extraction only consume a negligible amount of time and so are ignored in the calculation.

(2) Execution time

We compare the speed of the proposed cryptosystem with other ciphers [6,7,21] in encrypting the same image. The computer used for the test possesses a 2.1 GHz Intel Centrino Pro processor with 2 GB memory and 160 GB hard-disk capacity, running Microsoft Visual C++ 6.0 software. The measured timing data listed in Table 7 indicate that the proposed method is indeed the fastest among the three cryptosystems. It is also sufficiently fast for real-time applications.

4.6. Image authentication

For the convenience of description, the hashing process is denoted as $H_{x_{h0},x_{h1},x_{h2},x_{h3}}(\cdot)$, the encryption process as $E_{k_1}(\cdot)$ and the decryption process as $D_{k_2}(\cdot)$. The plain-image and the cipher-image are represented by P and C, respectively.

In the proposed scheme, the following information needs to be transmitted to the receiver for decryption: the cipher-image C; the 128-bit hash value h generated from the four hash keys and the plain-image by the sender. If authentication on the decrypted image is also required, the hash keys x_{h0}, x_{h1}, x_{h2} and x_{h3} should also be sent to the receiver secretly. When the decryption process is completed, a decrypted image $P' = D_h(C)$ is obtained. If authentication is needed, the receiver computes $h' = H_{x_{h0},x_{h1},x_{h2},x_{h3}}(P')$ using the four secret hash keys and the decrypted image. If h' is equal to h, the decrypted image is verified as identical to the original plain-image. Otherwise, the authentication process is not passed.

5. Conclusion

A fast and secure image encryption and authentication scheme is proposed. A keyed hash scheme constructed by chaotic maps is employed to generate a 128-bit hash value from the plain-image and the secret hash keys. This hash value plays the role of the encryption/decryption key. After decryption, the secret hash keys are used to authenticate the decrypted image. Simulation results show that satisfactory security performance is achieved in only one overall encryption round and so the speed efficiency is improved. Moreover, the security of the proposed scheme is verified by the analyses on its size of key space, key sensitivity, statistical and differential properties.

Acknowledgements

The work described in this paper was supported by a Grant from CityU (No. 7002216), the Chongqing Education Committee (No. kj091502) and the Natural Science Foundation of CQ CSTC (No. 2009BB2282).

References

[1] Schneier B. Cryptography: theory and practice. Boca Raton (FL): CRC Press; 1995.
[2] Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifur Chaos 1998;8(6):1259–84.
[3] Chen Guanrong, Mao Yaobin, Chui Charles K. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Soliton Fract 2004;21:749–61.
[4] Mao YB, Chen G, Lian SG. A novel fast image encryption scheme based on 3D chaotic baker maps. Int J Bifurcat Chaos 2004;14(10):3613–24.
[5] Guan Z, Huang F, Guan W. Chaos-based image encryption algorithm. Phys Lett A 2005;346:153–7.
[6] Lian SG, Sun J, Wang Z. A block cipher based on a suitable use of the chaotic standard map. Chaos Soliton Fract 2005;26:117–29.
[7] Wong KW, Kwok Bernie SH, Law WS. A fast image encryption scheme based on chaotic standard map. Phys Lett A 2008;372:2645–52.
[8] Wang Kai, Pei Wenjiang, Zou Liuhua, Song Aiguo, He Zhenya. On the security of 3D cat map based symmetric image encryption scheme. Phys Lett A 2005;343:432–9.
[9] Xiao Di, Liao Xiaofeng, Wei Pengcheng. Analysis and improvement of a chaos-based image encryption algorithm. Chaos Soliton Fract 2007; December 2007. doi:10.1016/j.chaos.2007.10.009.
[10] Wang Yong, Liao Xiaofeng, Xiang Tao, Wong Kwok-Wo, Yang Degang. Cryptanalysis and improvement on a block cryptosystem based on iteration a chaotic map. Phys Lett A 2007;363:277–81.
[11] Wei Jun, Liao Xiaofeng, Wong Kwok-wo, Zhou Tsing. Cryptanalysis of a cryptosystem using multiple one-dimensional chaotic maps. Commun Nonlinear Sci Numer Simulat 2007;12:814–22.
[12] Xun Yi, How Tan Chik, Kheong Siew Chee. A new block cipher based on chaotic tent maps. IEEE Trans Circuits Syst I 2002;49(12):1826–9.
[13] Jakimoski G, Kocarev L. Chaos and cryptography: block encryption ciphers based on chaotic maps. IEEE Trans Circuits Syst I 2001;48(2):163–9.

[14] Stojanovski T, Kocarev L. Chaos-based random number generators – part I: analysis. IEEE Trans Circuits Syst I 2001;48(3):281–8.
[15] Stojanovski T, Kocarev L. Chaos-based random number generators – part II: practical realization. IEEE Trans Circuits Syst I 2001;48(3):382–5.
[16] Xiao D, Liao X, Deng S. One-way hash function construction based on the chaotic map with changeable-parameter. Chaos Soliton Fract 2005;24:65–71.
[17] Wong KW. A combined chaotic cryptographic and hashing scheme. Phys Lett A 2003;307:292–8.
[18] Zhang J, Wang X, Zhang W. Chaotic keyed hash function based on feedforward–feedback nonlinear digital filter. Phys Lett A 2007;362:439–48.
[19] Wang Yong, Liao Xiaofeng, Xiao Di, Wong Kwok-Wo. One-way hash function construction based on 2D coupled map lattices. Inform Sciences 2008;178:1391–406.
[20] Shannon CE. Communication theory of secrecy system. Bell Syst Tech J 1949;28:656–715.
[21] Pisarchik AN, Zanin M. Image encryption with chaotically coupled chaotic maps. Physica D 2008;237:2638–48.