Signal Processing 93 (2013) 3039–3052

Contents lists available at SciVerse ScienceDirect

Signal Processing

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

Image encryption using a new parametric switching chaotic system

Yicong Zhou*, Long Bao, C.L. Philip Chen

Department of Computer and Information Science, University of Macau, Macau, China

ARTICLE INFO

Article history:
Received 29 January 2013
Received in revised form
17 April 2013
Accepted 22 April 2013
Available online 29 April 2013

Keywords:
Parametric switching chaotic system
Image encryption
Security analysis

ABSTRACT

This paper introduces a new parametric switching chaotic system (PSCS) and its corresponding transforms for image encryption. The proposed PSCS has a simple structure and integrates the Logistic, Sine and Tent maps into one single system. The PSCS shows more general properties, including the Sine and Tent maps as special instances. It has complex chaotic behaviors. A novel image encryption algorithm is introduced using the proposed PSCS and its transforms. Simulation results and security analysis are given to demonstrate that the proposed algorithm can encrypt different types of images with a high level of security.

1. Introduction

Nowadays, information, particularly in the format of images/videos, plays an increasingly important role in digital communications and networks. Benefited from the great advances of network technologies, people all over the world can easily search information, work on projects, and communicate with friends via the Internet. Every single minute, billions and tons of images and videos are created and transmitted over networks. These images and videos may contain private or sensitive information such as personal information, medical record, commercial designs or secret messages. Improper distribution of them may cause serious problems to individuals and organizations. As a result, the increasing demand of providing security protection to these images and videos has become an urgent and imperative issue for both individuals and organizations. Image encryption as an effective tool is used to enhance security of images and videos by transforming them into an unrecognized format. In such a way, protected images and videos can be safely transmitted over public channels and networks, without worrying about being intercepted and captured.

Based on different technologies, such as chaos [1–3], wave transmission [4], fractional Mellin transform [5,6], p-Fibonacci transform [7], visual cryptography [8], elliptic curve ElGamal [9], gray code [10], gyrator transform [11] and so on, many image encryption algorithms have been developed in recent years. Traditional Advanced Encryption Standard (AES) originally developed for data encryption can also be used to encrypt images. However, because of the lack of consideration about the redundancy property of images, AES does not show a good performance in image encryption, especially in encrypting image with block contents. Due to the fact that chaotic maps show excellent random behaviors, state ergodicity and sensitivity to the initial values and system parameters, chaotic maps have been utilized for image encryption [12–24]. Pareek et al. proposed image encryption algorithms using chaotic Logistic map in 2006 [14] and 2009 [15], respectively. Singh and Sinha combined the Hartley transform with the Logistic map for image encryption [17]. Recently, Wang et al. used the Logistic map to encrypt color images [1]. However, the variant density function of the Logistic map has been found to be not uniform [25]. Image encryption algorithms using one-dimensional (1D) chaotic

* Corresponding author. Tel.: +853 83978458; fax: +853 28838314.
E-mail address: yicongzhou@umac.mo (Y. Zhou).

0165-1684/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.sigpro.2013.04.021

map have been shown to be vulnerable to low-computation-cost analysis using iteration and correlation functions [26].

To overcome the security weakness of existing 1D chaotic maps, this paper introduces a new parametric switching chaotic system (PSCS) by embedding three existing chaotic maps, including the Logistic, Sine and Tent maps. Under different conditions, the proposed PSCS can not only revert back into traditional Sine and Tent maps, but also create more new chaotic sequences. It shows more general and robust random properties and complex chaotic behaviors. The corresponding PSCS transforms are also presented. Utilizing the proposed PSCS and its transforms, we introduce a new image encryption algorithm to satisfy the Shannon's confusion and diffusion properties. Computer simulations and security analysis are provided to demonstrate the algorithm's performance with respect to encryption and security.

This paper is organized as follows. Section 2 designs the new PSCS and discusses its properties. Section 3 presents two corresponding PSCS transforms. Section 4 introduces the novel image encryption algorithm. Computer simulation results and comparisons are presented in Section 5. Section 6 provides detailed security study and various attacks to the proposed image encryption algorithm. Section 7 reaches a conclusion.

2. The parametric switching chaotic system

This section introduces a new chaotic system called the parametric switching chaotic system (PSCS). Its properties are also discussed.

2.1. The PSCS

The new PSCS has a simple structure as shown in Fig. 1. It is a combination of three 1D chaotic maps: the Logistic, Sine and Tent maps. The output of the Logistic map controls a switch to select either the Sine map or the Tent map as a generator to produce the PSCS's output sequence.

The new PSCS is defined by

$$X_{i+1}=\begin{cases}\mathcal{T}(X_i) & C_i\geq 0.5\\ \mathcal{S}(X_i) & C_i<0.5\end{cases} \tag{1}$$

where X_{i+1} and $X_i(i=0,1,2,\ldots)$ are the $(i+1)$th and ith state values of the PSCS, respectively; and $\mathcal{T}(X_i)$ is the output of the Tent map in Eq. (2) where u is a positive real constant, $u\in[0,2]$;

$$\mathcal{T}(X_i)=\begin{cases}uX_i & X_i<0.5\\ u(1-X_i) & X_i\geq 0.5\end{cases} \tag{2}$$

and $\mathcal{S}(X_i)$ is the output of the Sine map in Eq. (3) where the control parameter $a\in[0,1]$:

$$\mathcal{S}(X_i)=a\sin(\pi X_i) \tag{3}$$

and C_i is the output of the Logistic map in Eq. (4) where the parameter $r\in[0,4]$:

$$C_i=\mathcal{L}(C_{i-1})=rC_{i-1}(1-C_{i-1}) \tag{4}$$

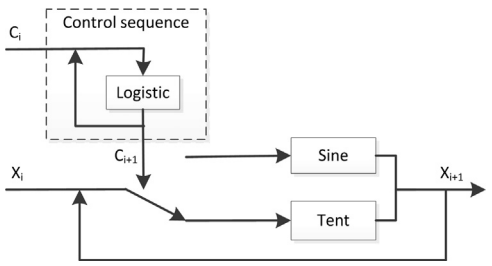

Fig. 1. The structure of the proposed PSCS.

2.2. Discussion

As shown in Fig. 1, the proposed PSCS has a simple structure that combines three traditional 1D chaotic maps. This provides users the convenience and simplicity for implementation in both hardware and software.

The PSCS shows more general properties than these existing ones. Depending on the output values of the Logistic map, the output sequence of the proposed PSCS is a mixed series obtained from either the Tent map or the Since map. The PSCS output is specified by three parameters (u, a and r) and two initial values (X_0 for the PSCS in Eq. (1) and C_0 for the Logistic map in Eq. (4)). When the parameter r in the Logistic map varies within the following ranges, the PSCS will revert back to traditional Tent and Sine maps.

- If $r\in[0,2)$ in Eq. (4), $0\leq C_i<0.5$, then the PSCS reverts back to the Sine map.
- If $r\in[2,3]$, $0.5\leq C_i\leq 1$, then the PSCS becomes the Tent map.
- If $r\in(3,4]$, $0\leq C_i\leq 1$, the PSCS is a completely new chaotic system with a mixture output either from the Sine map or from the Tent maps.

By embedding three existing 1D chaotic maps, the PSCS shows more complex chaotic behaviors than these existing 1D ones. To quantitatively evaluate their chaotic behaviors, we use Information Entropy [27] to measure the randomness of their output sequences. It is defined by

$$H(R)=-\sum_{i=0}^{F-1}P(R=i)\log_2 P(R=i) \tag{5}$$

where F is the number of bins; $P(\cdot)$ is the discrete probability density function. Information Entropy reaches the maximum value when all signal values are randomly distributed.

In this experiment, we change values of the parameters (u, a and r) with the initial values (X_0 and C_0) fixed. We then measure the output sequences of the proposed PSCS and three 1D chaotic maps using Information Entropy in Eq. (5). The results are shown in Table 1 where we choose $F=256$ and parameters of the Logistic, Tent and Sine maps to be the same as that of the PSCS. As can be seen, the PSCS has larger Information Entropy values compared to three existing ones under the same parameter settings. The PSCS output is more randomly distributed.

Due to the fact that traditional 1D chaotic maps show chaotic behaviors only when their parameters are limited within a specific range, the proposed PSCS, however, has chaotic behaviors with parameters being selected from a much larger dynamic range. Even in the worst situation in which the Logistic map loses its chaotic behaviors (e.g. its parameter is out of the specific range where the Logistic map has chaotic behavior), the proposed PSCS reverts back to the Tent map or the Sine map and still keeps good chaotic behaviors.

Based on Eqs. (1)–(4), the PSCS contains three parameters (u, a and r) and two initial values (X_0 and C_0). Similar to other chaotic systems, the proposed PSCS is extremely sensitive to its parameters and initial values. To verify this, a set of tests has been done and the results are shown in Fig. 2. Fig. 2(a)–(c) plots the correlation of two output sequences of the proposed PSCS when a tiny change, such as 10^{-14}, is made to one of its parameters. Fig. 2(d)–(e) shows the cases when the PSCS's initial values have a tiny alteration (10^{-14}). For example, Fig. 2(a) plots the correlation of two output sequences, S_1 and S_2. S_1 is obtained by the PSCS with parameter $a=1$ while S_2 is generated by the PSCS with the same parameters except for setting $a=0.99999999999999$. If S_1 and S_2 are similar (or highly correlated), all points will be located in or close to the diagonal line. As can be seen in Fig. 2(a), all points spread over the entire data range. This means that a tiny change (10^{-14}) of the parameter a leads to completely different output sequences of the PSCS. Similar results are obtained by changing other parameters and initial values as shown in Fig. 2(b)–(e). These results demonstrate that the proposed PSCS has an excellent chaotic property, namely a strong sensitivity to its parameters and initial values.

Table 1
Information Entropy test.

Parameter setting			Logistic map (r)	Tent map (u)	Sine map (a)	PSCS (r, u, a)
r	u	a				
3.5	1.5	0.90	2.0202	6.5296	6.9781	**7.3082**
3.6	1.6	0.92	6.3222	6.8890	7.2188	**7.5917**
3.7	1.7	0.94	7.0919	7.2205	1.6378	**7.6292**
3.8	1.8	0.96	7.3686	7.4872	7.4558	**7.7658**
3.9	1.9	0.98	7.4485	7.7420	7.6723	**7.8461**
4.0	2.0	1.00	7.6864	0.0589	7.6527	**7.7578**

Chaotic maps are frequently used for image encryption due to their excellent chaotic behaviors and sensitivity to their initial values and parameters. The PSCS contains more parameters and initial values than existing 1D chaotic maps. From the security point of view, it ensures more difficulty for the unauthorized users to predict the PSCS's output. This makes the PSCS more suitable for security applications.

In summary, the proposed PSCS has at least the following excellent properties: The PSCS

(1) uses a simple structure to integrate three traditional 1D chaotic maps into one single system,
(2) generates different chaotic sequences while including the Tent and Sine maps as its special instances,

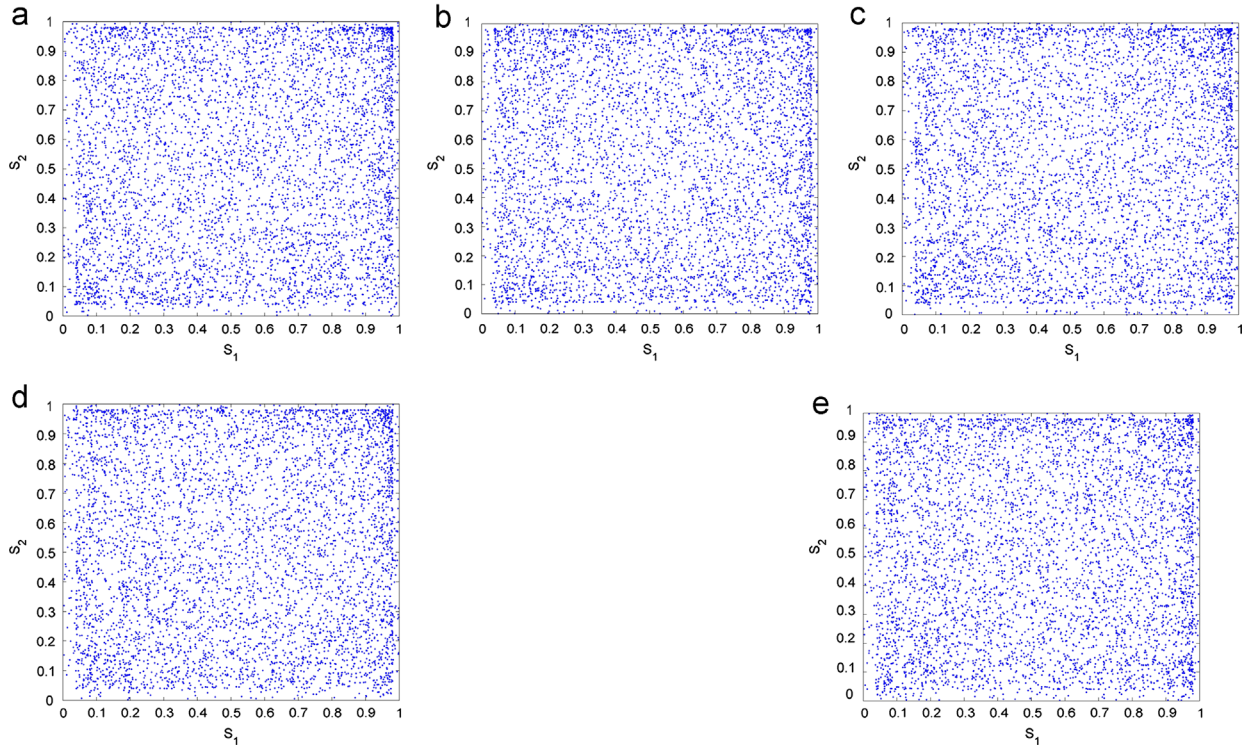

Fig. 2. Correlations of two PSCS's output sequences when slightly changing parameters and initial values only 10^{-14}. Their initial settings are $u=2$, $r=3.99$, $a=0.98$, $C_0=0.3$, $X_0=0.1$. (a) a; (b) r; (c) u; (d) X_0; and (e) the initial value of the Logistic map, C_0.

(3) is strongly sensitive to its parameters and initial states,
(4) has more complex chaotic behaviors,
(5) has chaotic behaviors within much larger ranges of parameter selection than those existing maps,
(6) is well suitable for security applications, such as image encryption.

3. The PSCS transforms

This section introduces the new one-dimensional (1D) and two-dimensional (2D) PSCS transforms. The 1D-PSCS transform is to transform a PSCS chaotic sequence into an integer sequence. The 2D-PSCS transform will be used for pixel permutation in the new image encryption algorithm proposed in Section 4.

3.1. The 1D-PSCS transform

Definition 1. Let $X_i (i=1,\ldots,N)$ be the PSCS chaotic sequence with length of N generated by Eq. (1), T_i be an integer sequence containing N non-repeat integers and $1 \le T_i \le N$. The following transformation is called the 1D-PSCS transform:

$$T_i = 1 + \lfloor (X_i + \varepsilon) N \rfloor \bmod N \tag{6}$$

where $\lfloor \cdot \rfloor$ denotes the floor function, and ε is an offset constant, $0 \le \varepsilon < 1$.

The 1D-PSCS transform is to map a PSCS chaotic sequence $(X_1, X_2, \ldots, X_N)$ into an integer sequence $(T_1, T_2, \ldots, T_N)$, which is actually a permutation of the integer sequence $(1, 2, \ldots, N)$. Varying the parameters or initial values of the PSCS will obtain a different PSCS sequences in Eq. (1) and a corresponding different integer sequences in Eq. (6). Then we obtain different permutations of $(1, 2, \ldots, N)$.

There may be some elements with the same values in the PSCS sequence $(X_1, X_2, \ldots, X_N)$. The offset constant ε is designed to deal with this situation. Its default setting is $\varepsilon = 0$. A different value will be set to ε when there exists duplicate values in the PSCS sequence. In this manner, elements with duplicate values at different locations in the PSCS sequence $(X_1, X_2, \ldots, X_N)$ will generate different non-repeat integers in the output sequence $(T_1, T_2, \ldots, T_N)$. For example, assume that the PSCS sequence is (0.1, 0.3, 0.2, 0.5, 0.7, 0.9, 0.4, 0.8, 0.1, 0.5), thus $N=10$. ε will be set to 0 except for $\varepsilon = 0.5$ when $X_9 = 0.1$ and $X_{10} = 0.5$. Thus, the output integer sequence will be (2, 4, 3, 6, 8, 10, 5, 9, 7, 1) which is a permutation of (1, 2, 3, 4, 5, 6, 7, 8, 9, 10).

The 1D-PSCS transform is excellent to be used for scrambling a data stream such as a text string or audio signal. It can also be applied to change pixel locations within an image. However, this has to be accomplished line by line, requiring a high computation cost. To overcome this problem, a more efficient 2D-PSCS transform is then introduced for image scrambling in Section 3.2.

3.2. The 2D-PSCS transform

Definition 2. Let I be an original image with size of $M \times N$, T_r and T_c be integer sequences generated by Eq. (6) with length of M and N, W_r and W_c be the row and column matrixes, respectively. The 2D-PSCS transform is defined as

$$S = W_r I W_c \tag{7}$$

where S is the scrambled image and

$$W_c(i,j) = \begin{cases} 1 & \text{for } (T_c(j), j) \\ 0 & \text{others} \end{cases} \tag{8}$$

$$W_r(k,l) = \begin{cases} 1 & \text{for } (k, T_r(k)) \\ 0 & \text{others} \end{cases} \tag{9}$$

where $i,\ j,\ k,\ l$ are integers, $1 \le i,\ j \le N$, and $1 \le k,\ l \le M$.

The 2D-PSCS transform is an effective and robust method to change all data locations within a 2D data matrix such as a grayscale image. Applying the 2D-PSCS transform only one time to an image can completely change all its pixel locations, achieving excellent diffusion property. The decryption process requires only one-time applying of corresponding reverse transform [7] to reconstruct the original data matrix. The inverse 2D-PSCS transform is defined in

$$I = W_r^{-1} S W_c^{-1} \tag{10}$$

4. The new PSCS-based image encryption algorithm

This section introduces a new simple and effective algorithm for image encryption using the proposed PSCS and its transforms. The algorithm is called the PSCS-IE algorithm. It encrypts the original images using a set of substitution and permutation (SP) processes as shown in Fig. 3. The encrypted image and decryption key are obtained after Q iterations of SP processes. A pseudo code of this algorithm is provided in Algorithm 1.

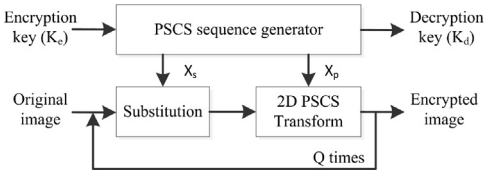

Fig. 3. The flowchart of the PSCS-IE algorithm.

Algorithm 1. The proposed PSCS-IE algorithm.

Input: The encryption key $K_e = (Q,\ u,\ r,\ a,\ C_0,\ X_0)$ and original image with size of $M \times N$

1. Set the length of the PSCS sequence: $L \leftarrow L_s + L_p$, where $L_s = M \times N$ for substitution and $L_p = M + N$ for permutation
2. **for** $i = 1$ to Q **do**
3. Update the initial value of the Logistic map $C_0^{(i)}$ based on Eq. (11)
4. Generate the PSCS sequence X with length of L using Eq. (1) and K_e
5. $X_s \leftarrow X(1 : L_s)$ for substitution, $X_p \leftarrow X(L_s + 1 : L)$ for permutation
6. Perform substitution to the input image using Eq. (15) and X_s
7. Apply 2D-PSCS transform to the image using X_p and Eqs. (6) and (7)
8. **end for**

Output: The encrypted image and decryption key $K_d = (Q, u, r, a, C_0^{(Q)}, X_0)$

The encryption key K_e consists of the iterations (Q) of the SP processes, the PSCS parameters ($u,\ r,\ a$), initial values of

the PSCS (X_0) and Logistic map (C_0). The initial value of the Logistic map (C_0) keeps updated in the beginning of each SP process based on Eq. (11). C_0 is added with information of the original image in the first SP process, and then keeps updating in each iteration by adding the initial PSCS value (X_0) in the previous SP process:

$$C_0^{(i)} = \begin{cases} \frac{1}{2}(C_0 + D) & \text{for } i = 1 \\ \frac{1}{2}(C_0^{(i-1)} + X_0) & \text{for } i > 1 \end{cases} \tag{11}$$

where $C_0^{(i)}$ and $C_0^{(i-1)}$ are initial values of the Logistic map in the ith and $(i-1)$th SP processes, respectively; and D collects information of the original image $I(m,n)$ with size of $M \times N$ as defined by

$$D = \begin{cases} D_1 & \text{for } D_1 \neq 0 \\ hex2dec(h_1h_2h_3h_4h_5h_6h_7h_8) \times 10^{-10} & \text{for } D_1 = 0 \end{cases} \tag{12}$$

where

$$D_1 = \left(\frac{1}{10^8} \sum_{m=1}^{M} \sum_{n=1}^{N} I(m,n) \right) \mod 1 \tag{13}$$

$$h_i = hash_i \oplus hash_{i+1} \oplus hash_{i+2} \oplus hash_{i+3} \tag{14}$$

where $hash_i$ is the ith hexadecimal number, when the MD5 hash with 128-bit hash value is expressed as a 32 digit hexadecimal number [28–30].

By updating the initial value of the Logistic map, a completely different PSCS sequence is generated in each SP process, improving diffusion and confusion properties of the encrypted images. The PSCS sequence is then divided into two subsequences, X_s with length of $(M \times N)$ for substitution, and X_p with length of $(M + N)$ for permutation.

The substitution process is an effective method to flat the image histogram because encrypted images with random-like histogram distribution are known to have excellent performance against statistics attacks [31]. It is defined in

$$E(m,n) = (\lfloor X_s(k) \times F \rfloor + I(m,n)) \mod F \tag{15}$$

where $\lfloor \cdot \rfloor$ is the floor function; m, n, k are integers, $1 \le m \le M$, and $1 \le n \le N$; $X_s(k)$ is the PSCS sequence for substitution where $k = (m-1)L + n$; $I(m,n)$ and $E(m,n)$ denote the input and output images of the substitution process; F is the maximum value of the input image $I(m,n)$, e.g. $F=256$ for grayscale image.

Correspondingly, in image decryption, the substitution process will use Eq. (16) to reconstruct image pixel values

$$I(m,n) = (\lfloor X_s(k) \times F \rfloor - E(m,n)) \mod F \tag{16}$$

The 2D-PSCS transform as an image permutation process has the excellent property of breaking the correlation of image pixels. X_p with length of $(M + N)$ is used to generate the row and column coefficient matrixes W_r and W_c, respectively, for the proposed 2D-PSCS transform in Eq. (7). Applying the proposed 2D-PSCS transform efficiently changes image pixel locations after the substitution process. It further improves the diffusion and confusion properties.

The users have the flexibility to choose the iterations Q of the SP process for image encryption in practical applications. Increasing iterations Q results in a higher level of security for encrypted images while requiring more computation cost. The users should balance the tradeoff between the security level and the encryption speed.

Image decryption is a simple inverse process of the proposed PSCS-IE algorithm as shown in Algorithm 1. It uses the decryption key K_d to generate the PSCS sequence, applies the inverse PSCS transform in Eq. (10) for permutation, and use Eq. (16) for substitution. After the same iterations of the inverse SP processes, the original image is reconstructed.

In short, the proposed PSCS-IE algorithm

(1) is a simple and robust encryption method with an enhanced level of security;
(2) offers the users the flexibility to select different iterations of the SP process to achieve their requirements of security and computation cost;
(3) has the excellent property in terms of confusion and diffusion;
(4) has a larger security key space.

5. Simulation results and comparisons

To show the performance of the proposed PSCS-IE algorithm, this section provides simulation results obtained from different images. Note that this paper sets the parameters and initial values in the encryption and decryption keys with length of 14 decimals for our simulations unless specified. However, the users have the flexibility to choose any other settings to meet their requirements in terms of security and computation.

The PSCS-IE algorithm can encrypt images with flexible iterations of the SP processes. Fig. 4(b)–(d) provides an observation of the SP encryption performance with different iterations. To withstand statistic attacks, one goal of the proposed PSCS-IE algorithm is to make the encrypted image unrecognizable and its histogram uniform-distributed, no matter how the histogram of the original image looks like. From results in Fig. 4(c), two iterations of SP processes are enough to achieve this goal. Thus, in all simulations and tests in the rest of this paper, we apply 2 iterations of the SP processes to encrypt images in the proposed PSCS-IE algorithm.

The proposed PSCS-IE algorithm has been applied to different types of images such as binary, grayscale and color images, as well as biometrics and medical images. Fig. 5 shows the encryption results of different images. The encrypted images are noise-like images without any leakage of the original information. This demonstrates that the proposed PSCS-IE algorithm can be used to fully protect various images for diverse applications. The reconstructed images are same as the original ones.

Fig. 6 compares the proposed PSCS-IE algorithm with the Chen's algorithm [20] and AES[1] [32]. The original binary image in Fig. 6(a) is a difficult case for image encryption because it contains large homogeneous regions.

[1] The Matlab code is located in http://buchholz.hs-bremen.de/aes/aes.htm.

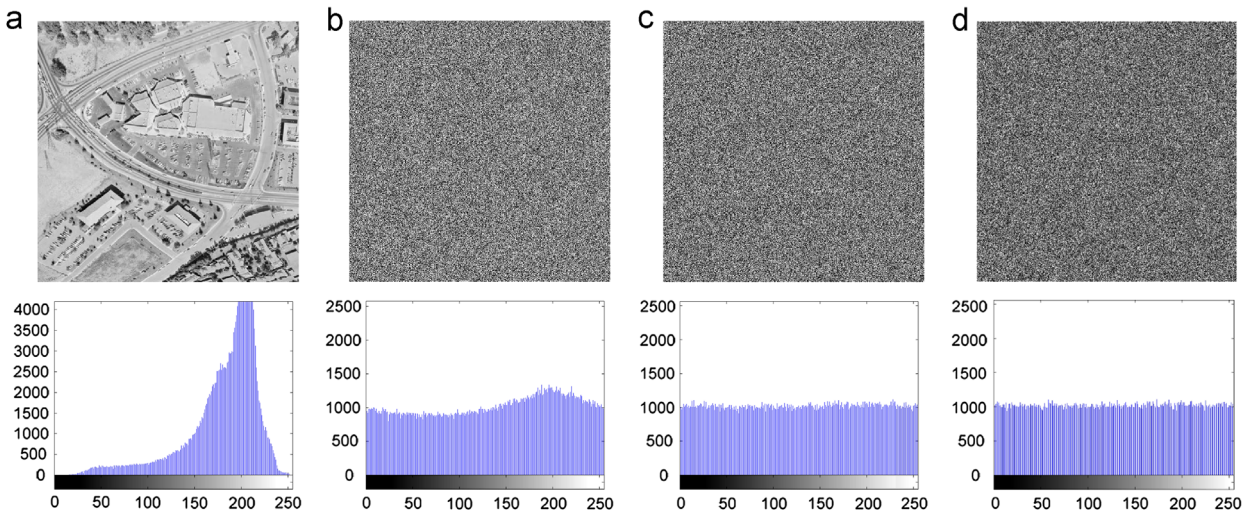

Fig. 4. Image encryption using the proposed PSCS-IE algorithm with different iterations of the SP processes: (a) the original image and its histogram; (b)–(d) are encrypted images and their histograms; (b) one iteration; (c) two iterations; (d) three iterations.

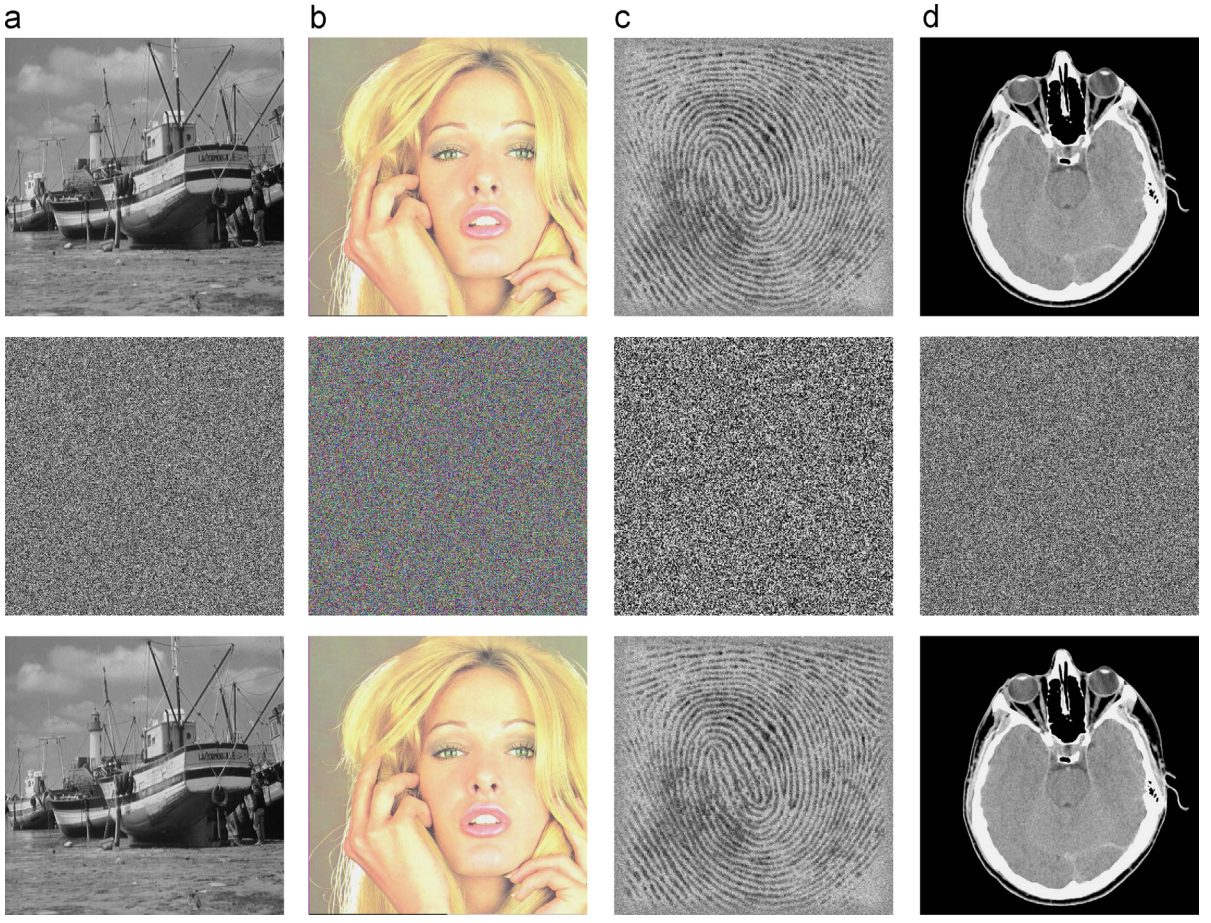

Fig. 5. Encrypting different images using the proposed PSCS-IE algorithm. The top, middle and bottom rows show the original, encrypted, reconstructed images, respectively. (a) grayscale image; (b) color image; (c) fingerprint (biometrics); (d) medical image.

The encrypted images by the Chen's algorithm and AES contain visual histogram patterns. The encrypted image in Fig. 6(d) by the AES has the information leakage. The proposed PSCS-IE algorithm successfully encrypts this difficult example as shown in Fig. 6(d). It outperforms other two algorithms.

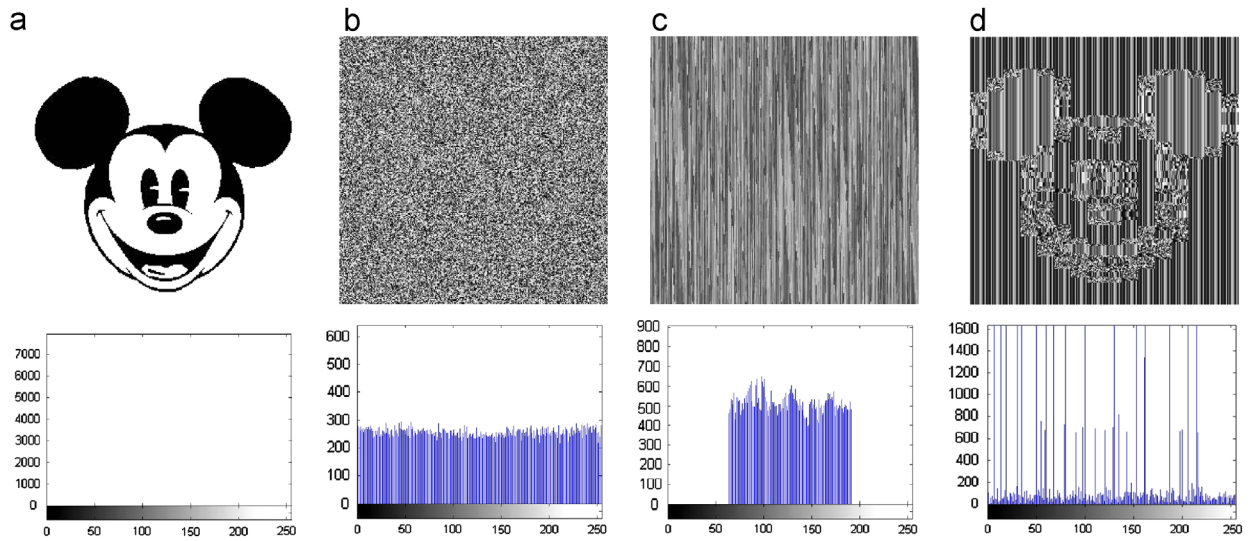

Fig. 6. Binary image encryption using different algorithms: (a) the binary original image and its histogram; (b)–(d) are encrypted images and their histograms; (b) the proposed PSCS-IE algorithm; (c) the Chen's algorithm [20]; (d) the AES [32].

6. Security analysis

Security is a vital issue and should be carefully taken into consideration both for the images to be protected and for the encryption algorithms. This section provides a detailed study of analyzing the security issues of the proposed PSCS-IE algorithm. The study includes the security key analysis, statistic analysis, and various attacks.

6.1. Security key analysis

Security keys are extremely important to an image encryption algorithm for ensuring the security of protected images in against the differential attacks and brute force attacks. Generally speaking, the security of an image encryption algorithm depends on its security key design [33]. An encryption algorithm should contain a sufficiently large key space and should be strongly sensitive to the change of security keys [34].

6.1.1. Security key space

As mentioned in Section 4, the security key of the proposed PSCS-IE algorithm is a combination of six subkeys: the iterations (Q) of the SP processes, the PSCS parameters (u, r, a), and initial values (C_0, X_0). To ensure that the PSCS has the best random and complex properties, all parameters are set within the range where each 1D chaotic map shows a strong chaotic behavior, i.e. $r\in[3.8,4]$ for the Logistic map, $a\in[0.9,1]$ for the Sine map, and $u\in[1.5,2]$ for the Tent map. The possible choices of these subkeys are ideally infinite, because they could be decimal numbers with an arbitrary length. As mentioned in the beginning of Section 5, this paper sets each subkey with length of 14 decimals. Consequently, the possible choices of r, a, u are 0.2×10^{14}, 0.1×10^{14} and 0.5×10^{14}, respectively. The initial values of the Logistic map and PSCS are within [0,1], and also set to be different values with length of 14 decimals. Therefore, the security key space of the proposed PSCS-IE algorithm is at least 10^{68} which is sufficiently large to withstand the brute force attacks [34].

6.1.2. Key sensitivity

In addition to a sufficiently large security key space, an encryption algorithm should also be strongly sensitive to its secure key changes. Here, we perform sensitivity tests in both the encryption and decryption processes as shown in Figs. 7–9.

Fig. 7 shows encryption results using the initial security key set and modified sets with a tiny change (10^{-14}) applied to each subkey, respectively. As can be seen in Fig. 7(b)–(g), all encrypted results look like noise images with uniform distributed histograms. Images in Fig. 7(h)–(l) are pixel-to-pixel differences between two encrypted images obtained by the initial and modified security key sets. These results demonstrate that slightly changing any subkey will lead to a completely different encryption result. The proposed PSCS-IE algorithm is strongly sensitive to the security key changes.

To test the key sensitivity in image decryption, a tiny change (10^{-14}) is also applied to each subkey. Fig. 8(c)–(h) shows the decryption results. As can be seen, the original image can be completely reconstructed in Fig. 8(c) only when the initial (correct) decryption key set is being utilized. However, any tiny change in subkey(s) will result in an unsuccessful decryption as shown in Fig. 8(d)–(h). These decryption results are unrecognized and their histograms keep uniform distributed. This ensures that the original image contents are fully protected.

Fig. 9 shows another key sensitivity test with each subkey set to 16 decimal length and a tiny change (10^{-16}) applied to several subkeys. Fig. 9(b)–(c) shows the noise-like encryption results using the initial key setting and the modified key setting with a slight change (10^{-16}) applied to parameter (u). The pixel-to-pixel differences between two encrypted images and their histograms are shown in Fig. 9(d). These demonstrate that a slight

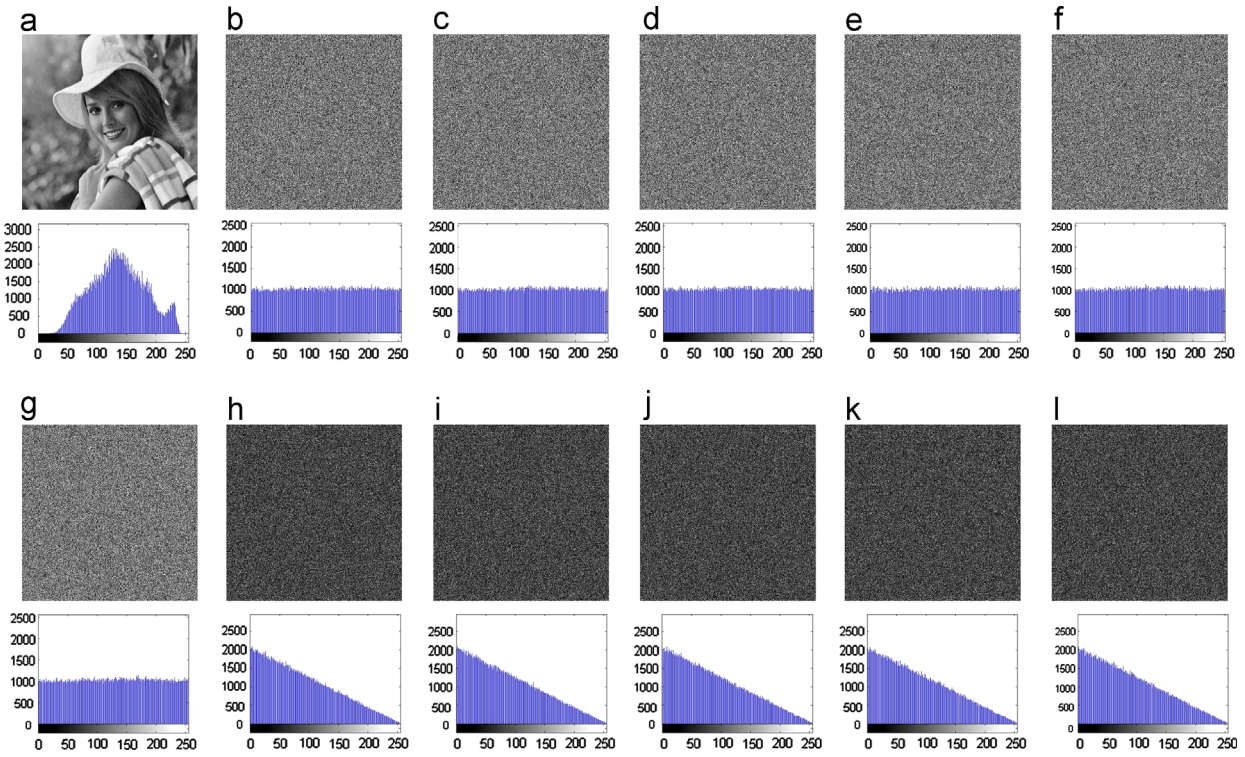

Fig. 7. Image encryption by slightly changing (10^{-14}) the PSCS's parameters and initial values: (a) the original image and its histogram; (b)–(f) are encrypted images and their histograms obtained by slightly changing (10^{-14}): (b) X_0; (c) C_0; (d) r; (e) u; (f) a; (g) the encrypted image and its histogram using the initial security key set; (h)–(l) are image differences and their histograms; (h) differences between (b) and (g); (i) differences between (c) and (g); (j) differences between (d) and (g); (k) differences between (e) and (g); (l) differences between (f) and (g).

change (10^{-16}) in subkey will lead to significant changes in the encrypted image. For image decryption, only using the correct decryption key setting is able to completely reconstruct the original image as shown in Fig. 9(e) which is reconstructed from the image in Fig. 9(b). Otherwise, even if, for example, there is a tiny change (10^{-16}) in X_0 in the decryption key, the reconstructed image is a noise-like image as shown in Fig. 9(f), which is completely different from the original one. These results imply that image reconstruction performs successfully only when the correct decryption key is being utilized.

6.2. Statistic analysis

Except for the histogram property of the encrypted image discussed in Section 5, this section analyzes security of the proposed PSCS-IE algorithm in terms of two statistic analysis methods including correlation analysis and Information Entropy.

6.2.1. Correlation analysis

A visually good-looking image generally contains pixels with high neighborhood correlations while a random-like image does not. An encryption algorithm intends to transform an original good-looking image into a random-like encrypted image with low correlation among neighborhood pixels.

2048 sample pixels are randomly selected from the original and the encrypted images in Fig. 4(a) and (c), respectively. Fig. 10 plots the distribution of these sample pixels and their neighborhood pixels at the horizontal, vertical, and diagonal directions. As can be seen on the top row, the pixels converge to the region in or close to the diagonal line $y=x$. This means that the neighborhood pixels in the original image in Fig. 4(a) are equal or close to each other, and thus have a high correlation. However, seen from the bottom row in Fig. 10, pixel values in the encrypted image in Fig. 4(c) are distributed to the entire data range of the image. This indicates that the neighborhood pixels in the encrypted image have little correlations showing good confusion property of the proposed PSCS-IE algorithm.

To quantitatively evaluate the correlations, we calculate correlation coefficients of neighborhood pixels at the horizontal, vertical and diagonal directions using [35]

$$corr(x,y)=\frac{E[(x-\mu_x)(y-\mu_y)]}{\sigma_x\sigma_y} \tag{17}$$

where μ_x and μ_y are the mean values of x and y, and σ_x and σ_y denote the standard deviations of x and y, respectively; The function $E[\cdot]$ is the expected value.

A correlation value close to 1 means a strong relationship between pixels and their neighbors, while the correlation value approaching to 0 indicates no or a weak relationship between pixels. Table 2 lists the correlation coefficients of neighborhood pixels in the encrypted images in Fig. 6(b)–(d) by the proposed PSCS-IE, Chen's and AES algorithms, respectively. As can be seen in Table 2, the PSCS-IE algorithm has the lowest correlation values at all directions and thus outperforms other two algorithms.

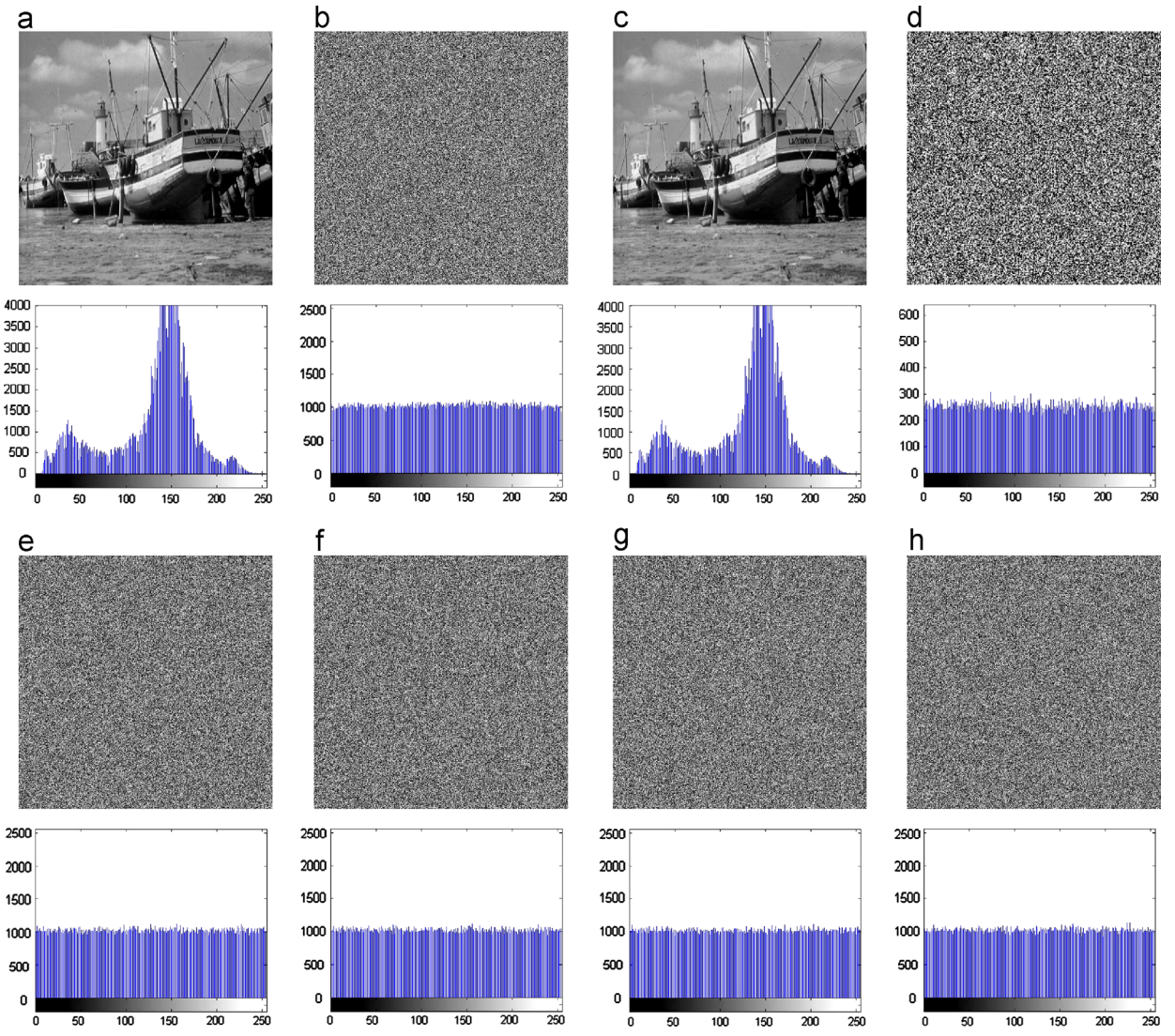

Fig. 8. Image decryption using different security keys: (a) the original image and its histogram; (b) the encrypted image and its histogram; (c) the reconstructed image using the correct key set and its histogram; (d)–(h) are the reconstructed images and their histograms using a tiny change (10^{-14}) in (d) r; (e) u; (f) a; (g) C_0; (d) X_0.

6.2.2. *Information Entropy*

In addition to evaluate signal randomness, Information Entropy can also be used to assess whether an encrypted image is a random-like image with pixel values randomly distributed. In this case, F and R in Eq. (5) represent the maximum and individual pixel values in an image, respectively. For a gray image, $F=256$ and each pixel can be represented by 8 binary bits. The maximum of Information Entropy $H(R)=8.0$ when $P(R=i)=\frac{1}{256}$, namely the image R is uniformly distributed.

28 test images are obtained from the USC-SIPI image database.[2] Table 3 lists the measure results of the Information Entropy of the images before and after being encrypted by the proposed PSCS-IE algorithm. As can be seen, Information Entropy scores of the PSCS-IE's encrypted images are 7.9988 in average. It is much close to the maximum value of Information Entropy. This means that the encrypted images are uniformly distributed. The proposed PSCS-IE algorithm shows excellent performance in image encryption.

6.3. *Differential attacks*

Differential attack is a form of cryptanalysis in which an attacker tries to recover the security key of the encryption algorithm by checking the non-random behaviors and properties caused by a set of predefined differences through each encryption step [34]. According to the

[2] The USC-SIPI image database is located in http://sipi.usc.edu/database.

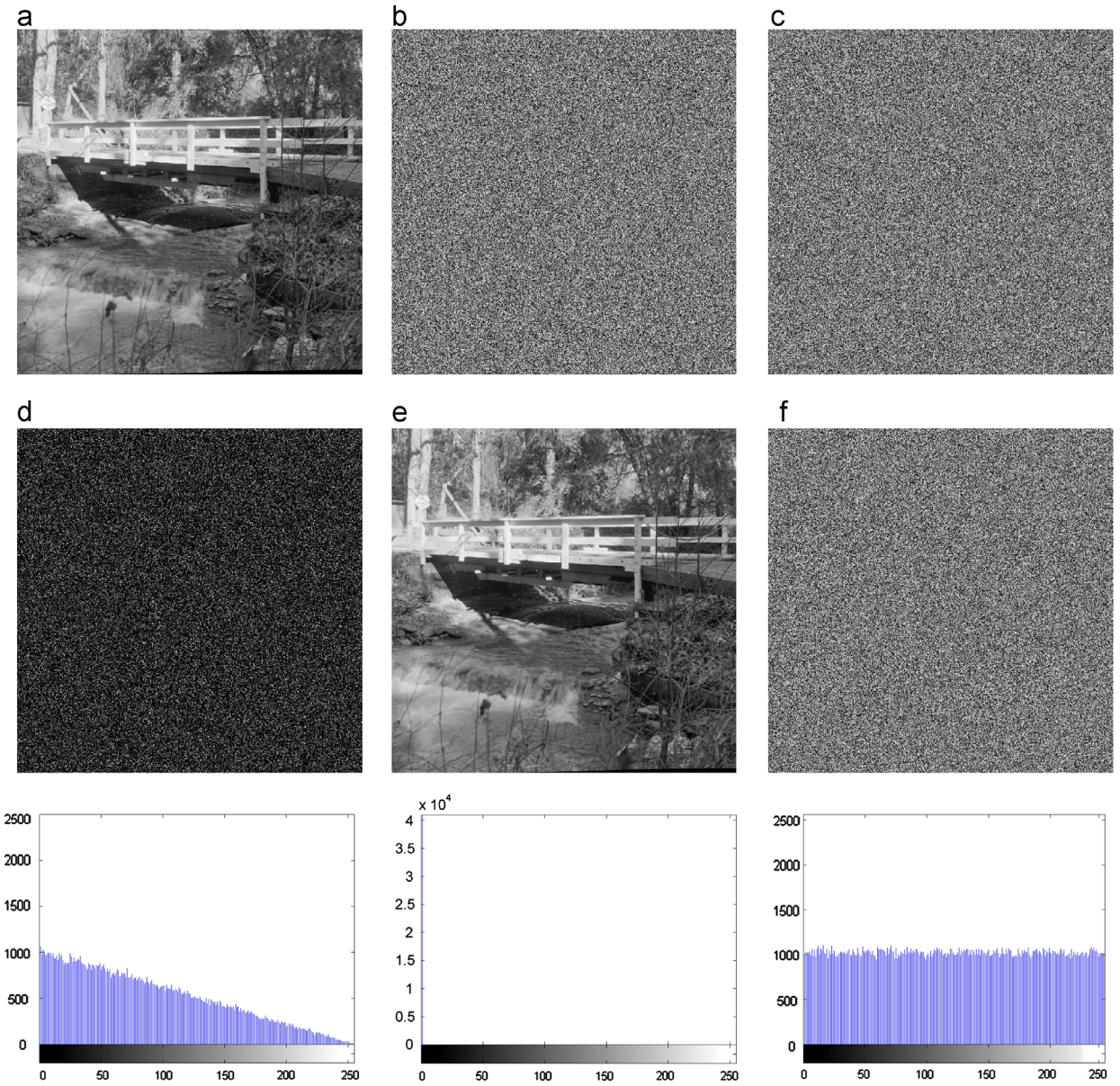

Fig. 9. Key sensitivity test using a longer subkey length (16) and a tiny change (10^{-16}) applied to subkeys: (a) the original image; (b) the encrypted image using initial security key setting with each subkey length of 16 decimals; (c) the encrypted image by slightly changing (10^{-16}) in parameter (u); (d) the differences between (b) and (c) and their histogram; (e) the reconstructed image of (b) using the correct key set and its histogram; (f) the reconstructed image of (b) using a slight change (10^{-16}) in X_0 and its histogram.

concept of differential attack, we use two images with only a pixel difference between each other, and check their corresponding encrypted images using the proposed PSCS-IE algorithm.

A modified image is generated by setting 0 to a pixel located in position (150, 100) in a Clock image with size of 256×256. The proposed PSCS-IE algorithm is used to encrypt both the original and modified images. We then obtain the pixel-to-pixel differences between two clock images and between two corresponding encrypted ones as shown in Fig. 11(c). As can be seen, a single pixel change in the original image results in significant changes spreading over the entire encrypted image. The attacker is unable to find any useful cue using the differential cryptanalysis. The proposed PSCS-IE is able to withstand differential attacks.

To quantitatively evaluate this differential effect to encrypted images, we use the Unified Average Changing Intensity (UACI) and Number of Pixel Change Rate (NPCR) as defined in Eqs. (18) and (19) where E_1 and E_2 are two encrypted images with size of $M \times N$ [20]:

$$UACI = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left[\frac{|E_1(m,n)-E_2(m,n)|}{255}\right] \times 100\% \qquad (18)$$

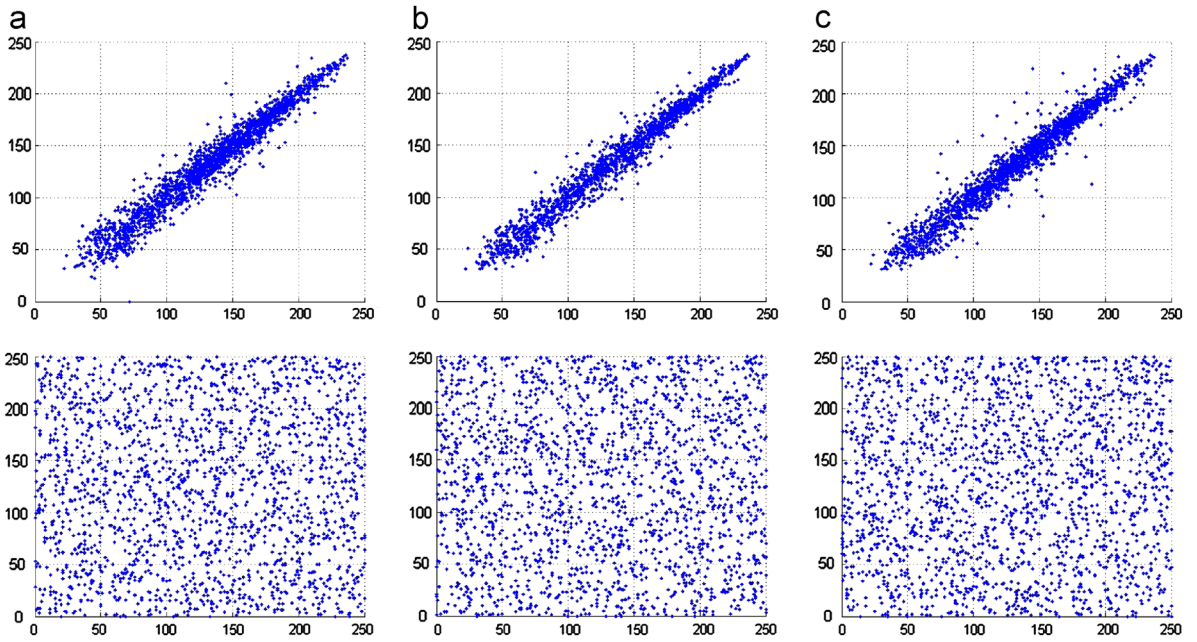

Fig. 10. Correlation of neighborhood pixels at different directions before and after encryption. The top row shows neighborhood pixel correlation in the original image in Fig. 4(a); the bottom row shows neighborhood pixel correlation in the encrypted image in Fig. 4(c). (a) Horizontal direction; (b) vertical direction; (c) diagonal direction.

Table 2
Correlation coefficients of neighborhood pixels at different directions.

Encryption algorithm	Vertical	Horizontal	Diagonal
Original image	0.9420	0.9455	0.9205
PSCS-IE	**−0.0054**	**0.0045**	**0.0031**
Chen's [20]	0.9728	0.0442	0.0469
AES [32]	0.8018	−0.0160	−0.0140

$$NPCR = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\mathcal{B}(m,n)}{MN} \times 100\% \tag{19}$$

where

$$\mathcal{B}(m,n) = \begin{cases} 1 & \text{for } E_1(m,n) \neq E_2(m,n) \\ 0 & \text{otherwise} \end{cases}$$

After applying the differential attack, the UACI calculates the average value of changed pixels between two encrypted images while the NPCR measures the percentage of the changed pixel numbers between two encrypted images. Table 3 shows the UACI and NPCR results of the proposed PSCS-IE algorithm using the same differential attack to the 28 test images used in Section 6.2.2. As can be seen, the average UACI and NPCR are 33.38% and 99.61%, respectively. They are close to their corresponding expected values of grayscale images with 33.464% for the UACI and 99.609% for the NPCR proved in [36]. This further proves that the PSCS-IE algorithm shows excellent performance in against differential attacks.

6.4. Strict avalanche criterion

Different from the NPCR and UACI that quantitatively evaluate the pixel-level changes, the strict avalanche

Table 3
Performance measures of the PSCS-IE algorithm to different images with respect to Information Entropy, differential analysis and strict avalanche criterion (SAC).

File name	Information entropy analysis		Differential analysis		SAC
	Original image H_P	Encrypted image H_S	NPCR (%)	UACI (%)	NBCR (%)
5.1.09	6.7093	7.9966	99.60	33.14	50.08
5.1.10	7.3118	7.9971	99.61	33.24	50.04
5.1.11	6.4523	7.9975	99.64	33.72	49.98
5.1.12	6.7057	7.9972	99.60	33.56	49.96
5.1.13	1.5483	7.9965	99.63	33.77	50.04
5.1.14	7.3424	7.9977	99.62	33.21	50.05
5.2.08	7.5237	7.9991	99.61	33.31	49.96
5.2.09	6.9940	7.9992	99.60	33.62	50.08
5.2.10	5.7056	7.9991	99.61	33.31	50.05
5.3.01	7.5237	7.9998	99.60	33.42	49.98
5.3.02	6.8303	7.9996	99.62	33.29	50.01
7.1.01	6.0274	7.9990	99.59	33.25	50.07
7.1.02	4.0045	7.9991	99.62	33.53	49.97
7.1.03	5.4957	7.9990	99.59	33.27	50.01
7.1.04	6.1074	7.9992	99.62	33.21	50.01
7.1.05	6.5632	7.9992	99.61	33.21	49.98
7.1.06	6.6953	7.9992	99.61	33.30	49.97
7.1.07	5.9916	7.9991	99.60	33.15	49.99
7.1.08	5.0534	7.9990	99.58	33.26	50.02
7.1.09	6.1898	7.9991	99.61	33.26	50.05
7.1.10	5.9088	7.9990	99.63	33.23	49.95
7.2.01	5.6415	7.9996	99.61	33.59	50.00
boat.512	7.1914	7.9992	99.61	33.42	49.97
elaine.512	7.5060	7.9992	99.60	33.37	49.96
gray21.512	4.3923	7.9993	99.61	33.37	50.03
numbers.512	7.7292	7.9994	99.60	33.36	50.01
ruler.512	0.5000	7.9987	99.61	33.77	49.95
testpat.1k	4.4077	7.9998	99.62	33.43	50.01
Mean value	5.9304	7.9988	99.61	33.38	50.01

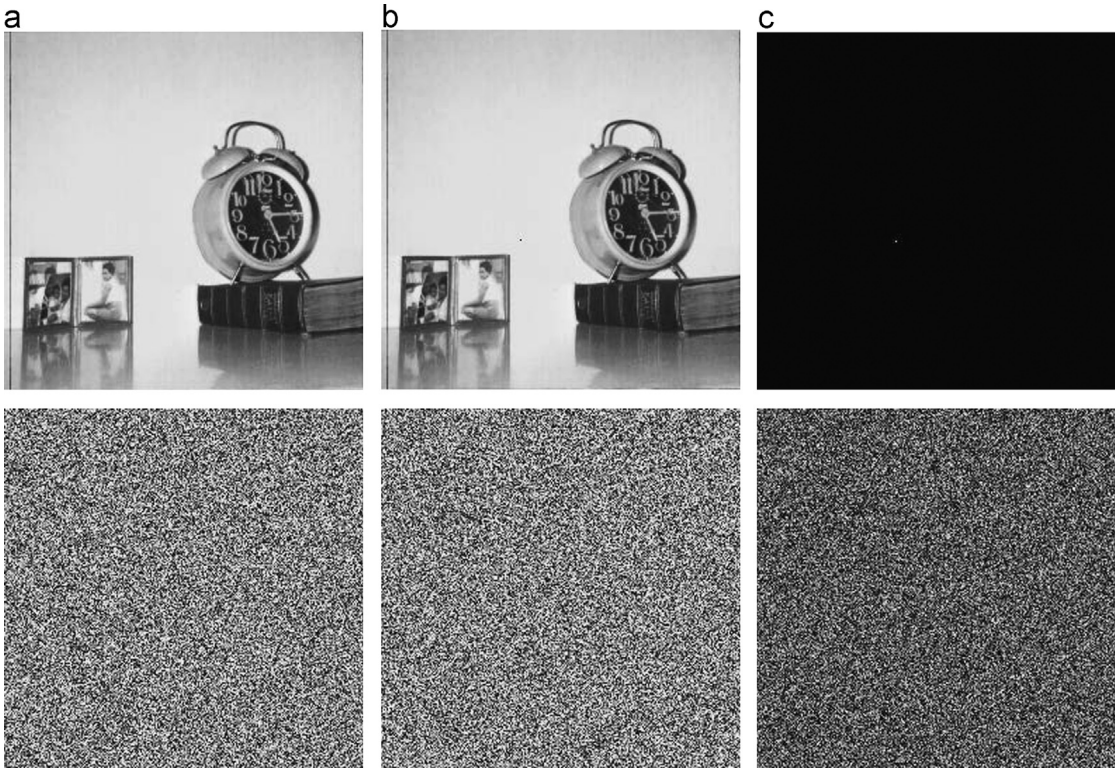

Fig. 11. One single pixel change in the original image leads to significant changes in the encrypted image. (a) the original and encrypted Clock images; (b) the Clock image with a single pixel change and its encrypted one; (c) image differences between the original and encrypted ones.

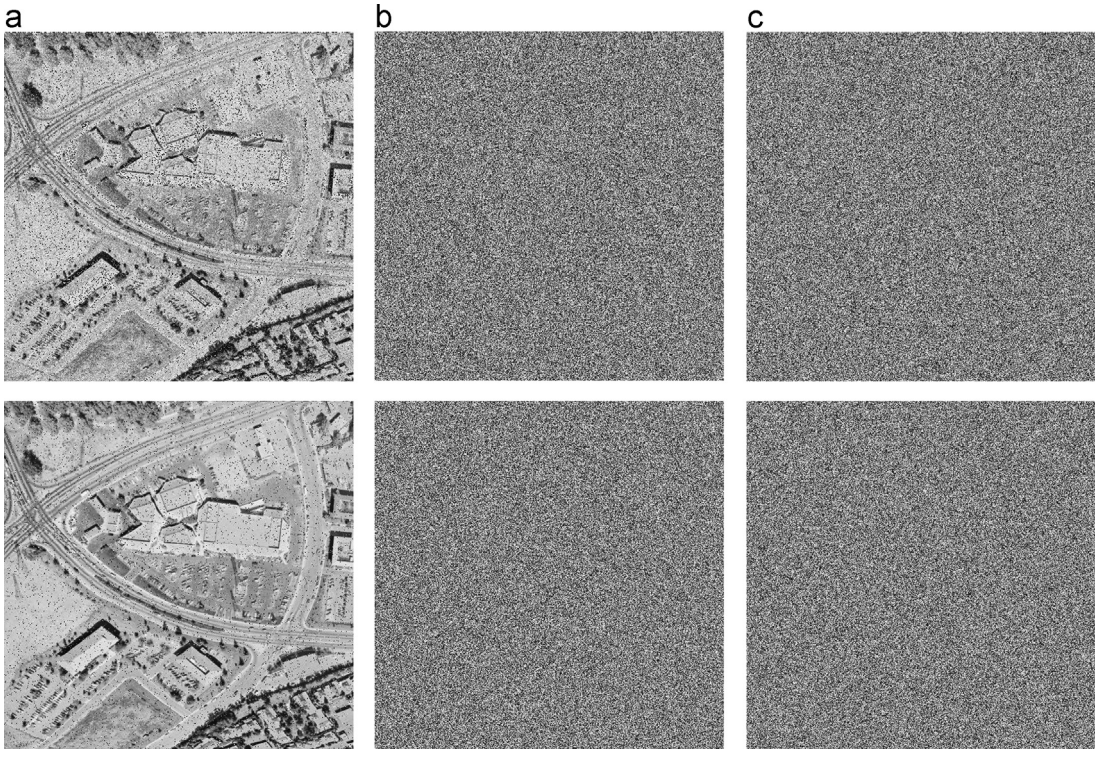

Fig. 12. Noise attacks to different algorithms in image reconstruction: the top row shows the images reconstructed from the encrypted image with 1% Gaussian noise; the bottom row shows the images reconstructed from the encrypted image with 5% Salt and Pepper Noise; the original image in Fig. 4(a) is encrypted by: (a) the proposed PSCS-IE algorithm; (b) the Liao's algorithm [4]; (c) the Wu's algorithm [40].

criterion (SAC) is proposed to observe the bit-level changes. The SAC states that a very small difference (i.e. one bit change) in the input will lead to an avalanche change in the output. According the SAC's definition in [37], we define the Number of Bit Change Rate (NBCR) in Eq. (20) to measure the SAC performance. The NBCR calculates the percentage of changed bit numbers between two bit streams. The ideal NBCR is 50% in average [38]:

$$NBCR = \frac{Hm[S_1, S_2]}{L_b} \times 100\% \tag{20}$$

where S_1 and S_2 are two bit streams with the bit length of L_b; The function $Hm[\cdot]$ is to calculate the Hamming distance of two bit streams.

Different from the differential attack, we get the modified image by changing the lowest bit of pixel value in [150, 100]. Then, after encryption,we convert two encrypted images into bit streams. Their NBCR is measured using Eq. (20). Table 3 lists the NBCR results. As can be seen, the average NBCR value of all images is 50.01% which is close to the ideal NBCR 50%. The proposed PSCS-IE algorithm meets the requirement of the strict avalanche criterion.

6.5. Noise attack

Almost all transmission channels are noise channels [39]. Data propagating over channels will be infected with different types of noise including the Gaussian noise and Salt and Pepper noise. An encryption algorithm should immune these noise infections (or attacks).

Fig. 12 shows the simulation results of noise analysis for different algorithms. In this experiment, the original image uses the image in Fig. 4(a). It is encrypted by the proposed PSCS-IE algorithm, Liao's algorithm [4] and Wu's algorithm [40], and then added with 1% Gaussian noise and 5% Salt and Pepper noise, respectively. These encrypted images with noise are reconstructed by the corresponding algorithms. As can be seen from the reconstructed results in Fig. 12, the Liao's and Wu's algorithms obtain noise-like images (Fig. 12(b) and (c)) and fail to recover the original information. However, the proposed PSCS-IE algorithm successfully reconstructs the original image with pleasant visual quality even containing noise. This demonstrates that the PSCS-IE algorithm outperforms two compared algorithms with respect to noise effects.

6.6. Data loss attack

The encrypted data may be partially modified or lost during transmission. An encryption algorithm should have the capability to immune the effect of the data loss. To perform data loss analysis, this section compares the proposed PSCS-IE algorithm with the AES, Liao's algorithm [4] and Wu's algorithm [40]. A “plane” image is encrypted by these four algorithms and immediately applied by a square pixel cutting with size of 40×40. These resulting images are then reconstructed by the corresponding algorithms as shown in Fig. 13. As can be seen from the reconstructed images, the data loss of the AES concentrates in a single large area. This may lead to the important information loss. The reconstructed images by the Liao's and Wu's algorithms are noise-like images, resulting in a complete loss of original information. By separating the data loss in a large area into small pieces and distributing them through the entire image, the PSCS-IE algorithm keeps the most important visual information in the

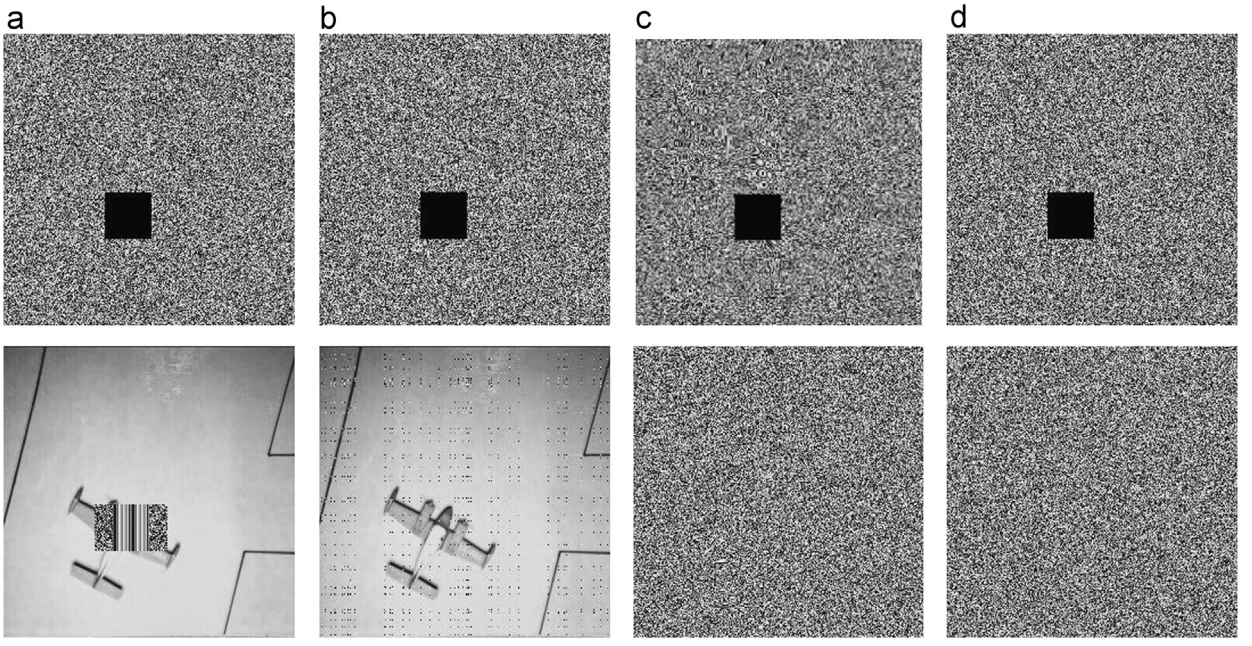

Fig. 13. Data loss attack to different encryption algorithms: the top row shows the encrypted images by different algorithms with a square data cutting applied; the bottom row shows the reconstructed images by corresponding algorithms. (a) AES; (b) the PSCS-IE; (c) the Liao's algorithm [4]; (d) the Wu's algorithm [40].

original image and thus outperforms the AES, Liao's and Wu's algorithms in the data loss attacks.

7. Conclusion

In this paper, we have introduced a new parametric switching chaotic system (PSCS). The PSCS embeds three well-known 1D chaotic sequences into one simple system. It has shown general properties, including the Sine and Tent maps as special cases, and complex chaotic behaviors due to the parameter-dependent outputs. The new PSCS is easy to be implemented in software and hardware. The 1D and 2D PSCS transforms have been proposed for efficiently scrambling data streams and images, respectively.

To investigate the PSCS's applications in image processing, we have introduced an image encryption algorithm using the proposed PSCS and its transforms. Simulation results have demonstrated that the proposed PSCS-IE algorithm shows excellent performance in encrypting different types of images. Security analysis has proved that the PSCS-IE algorithm is able to encrypt images with a high level of security and outperforms existing algorithms with respect to different tests and attacks.

Acknowledgement

This work was supported in part by the Macau Science and Technology Development Fund under Grant 017/2012/A1 and by the Research Committee at University of Macau under Grants SRG007-FST12-ZYC, MYRG113(Y1-L3)-FST12-ZYC and MRG001/ZYC/2013/FST.

References

[1] X. Wang, L. Teng, X. Qin, A novel colour image encryption algorithm based on chaos, Signal Processing 92 (2012) 1101–1108.
[2] S.M. Seyedzadeh, S. Mirzakuchaki, A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map, Signal Processing 92 (2012) 1202–1215.
[3] X. Tong, M. Cui, Image encryption scheme based on 3D baker with dynamical compound chaotic sequence cipher generator, Signal Processing 89 (2009) 480–491.
[4] X. Liao, S. Lai, Q. Zhou, A novel image encryption algorithm based on self-adaptive wave transmission, Signal Processing 90 (2010) 2714–2722.
[5] N. Zhou, X. Liu, Y. Zhang, Y. Yang, Image encryption scheme based on fractional Mellin transform and phase retrieval technique in fractional Fourier domain, Optics and Laser Technology 47 (2013) 341–346.
[6] N. Zhou, Y. Wang, L. Gong, Novel optical image encryption scheme based on fractional Mellin transform, Optics Communications 284 (13) (2011) 3234–3242.
[7] Y. Zhou, K. Panetta, S. Agaian, C.L.P. Chen, Image encryption using P-Fibonacci transform and decomposition, Optics Communications 285 (5) (2012) 594–608.
[8] T.H. Chen, K.H. Tsao, Y.S. Lee, Yet another multiple-image encryption by rotating random grids, Signal Processing 92 (9) (2012) 2229–2237.
[9] L. Li, A.A. Abd El-Latif, X.M. Niu, Elliptic curve ElGamal based homomorphic image encryption scheme for sharing secret images, Signal Processing 92 (4) (2012) 1069–1078.
[10] Y. Zhou, K. Panetta, S. Agaian, C.L.P. Chen, (n, k, p)-gray code for image systems, IEEE Transactions on Cybernetics 43 (2) (2013) 515–529.
[11] Z. Liu, H. Chen, T. Liu, P. Li, L. Xu, J. Dai, S. Liu, Image encryption by using gyrator transform and Arnold transform, Journal of Electronic Imaging 20 (1) (2011) 013020–013026.
[12] A. Akhshani, A. Akhavan, S.C. Lim, Z. Hassan, An image encryption scheme based on quantum Logistic map, Communications in Nonlinear Science and Numerical Simulation 17 (12) (2012) 4653–4661.
[13] Y. Wu, J.P. Noonan, S. Agaian, A wheel-switch chaotic system for image encryption, in: 2011 International Conference on System Science and Engineering (ICSSE), 2011, pp. 23–27.
[14] N.K. Pareek, V. Patidar, K.K. Sud, Image encryption using chaotic Logistic map, Image and Vision Computing 24 (9) (2006) 926–934.
[15] V. Patidar, N. Pareek, K. Sud, A new substitution-diffusion based image cipher using chaotic standard and Logistic maps, Communications in Nonlinear Science and Numerical Simulation 14 (7) (2009) 3056–3075.
[16] N. Zhou, Y. Wang, L. Gong, H. He, J. Wu, Novel single-channel color image encryption algorithm based on chaos and fractional Fourier transform, Optics Communications 284 (12) (2011) 2789–2796.
[17] N. Singh, A. Sinha, Optical image encryption using Hartley transform and Logistic map, Optics Communications 282 (6) (2009) 1104–1109.
[18] R. Ye, A novel chaos-based image encryption scheme with an efficient permutation-diffusion mechanism, Optics Communications 284 (22) (2011) 5290–5298.
[19] C. Fu, B. Lin, Y. Miao, X. Liu, J. Chen, A novel chaos-based bit-level permutation scheme for digital image encryption, Optics Communications 284 (23) (2011) 5415–5423.
[20] G. Chen, Y. Mao, C.K. Chui, A symmetric image encryption scheme based on 3D chaotic cat maps, Chaos, Solitons & Fractals 21 (3) (2004) 749–761.
[21] L. Bao, Y. Zhou, C.L.P. Chen, H. Liu, A new chaotic system for image encryption, in: 2012 International Conference on System Science and Engineering (ICSSE), 2012, pp. 69–73.
[22] A. Akhshani, S. Behnia, A. Akhavan, H.A. Hassan, Z. Hassan, A novel scheme for image encryption based on 2D piecewise chaotic maps, Optics Communications 283 (17) (2010) 3259–3266.
[23] L. Teng, X.Y. Wang, A bit-level image encryption algorithm based on spatiotemporal chaotic system and self-adaptive, Optics Communications 285 (20) (2012) 4048–4054.
[24] G.D. Ye, K.W. Wong, An efficient chaotic image encryption algorithm based on a generalized Arnold map, Nonlinear Dynamics 69 (4) (2012) 2079–2087.
[25] A. Lasota, M.C. Mackey, Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics, 2nd edition, Springer, New York, 1993.
[26] M.I. Sobhy, A.E.R. Shehata, Methods of attacking chaotic encryption and countermeasures, in: Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '01), 2001, pp. 1001–1004.
[27] C.E. Shannon, A mathematical theory of communication, Bell System Technical Journal 27 (1948) 379–423., pp. 623–656.
[28] H. Liu, X. Wang, K. Abdurahman, Image encryption using DNA complementary rule and chaotic maps, Applied Soft Computing 12 (5) (2012) 1457–1466.
[29] X. Wang, Q. Yu, A block encryption algorithm based on dynamic sequences of multiple chaotic systems, Communications in Nonlinear Science and Numerical Simulation 14 (2) (2009) 574–581.
[30] H. Liu, X. Wang, Color image encryption using spatial bit-level permutation and high-dimension chaotic system, Optics Communications 284 (16–17) (2011) 3895–3903.
[31] S. Li, X. Mou, Y. Cai, Improving security of a chaotic encryption approach, Physics Letters A 290 (3–4) (2001) 127–133.
[32] J.J. Buchholz, Matlab implementation of the Advanced Encryption Standard, 2001.
[33] A.J. Menezes, P.C. Van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, Inc., New York, 1997.
[34] G. Alvarez, S. Li, Some basic cryptographic requirements for chaos-based cryptosystems, International Journal of Bifurcation and Chaos 16 (8) (2006) 2129–2151.
[35] A.G. Bluman, Elementary Statistics: A Step by Step Approach, McGraw-Hill, Boston, 1997.
[36] C. Fu, J. Chen, H. Zou, W. Meng, Y. Zhan, Y. Yu, A chaos-based digital image encryption scheme with an improved diffusion strategy, Optics Express 20 (3) (2012) 2363–2378.
[37] R. Forre, The strict avalanche criterion: spectral properties of boolean functions and an extended definition, 1990.
[38] J.C.H. Castro, J.M. Sierra, A. Seznec, A. Izquierdo, A. Ribagorda, The strict avalanche criterion randomness test, Mathematics and Computers in Simulation 68 (1) (2005) 1–7.
[39] R.C. Gonzalez, R.E. Woods, Digital Image Processing, 3rd edition, Pearson Prentice Hall, 2007.
[40] Y. Wu, G. Yang, H. Jin, J.P. Noonan, Image encryption using the two-dimensional Logistic chaotic map, Journal of Electronic Imaging 21 (1) (2012)., pp. 013014–1.