A self-cited pixel summation based image encryption algorithm*

Guo-Dong Ye(叶国栋)[1,2,†], Xiao-Ling Huang(黄小玲)[1], Leo Yu Zhang(张愉)[3], and Zheng-Xia Wang(王政霞)[4]

[1] *Faculty of Mathematics and Computer Science, Guangdong Ocean University, Zhanjiang* 524088, *China*
[2] *College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou* 310027, *China*
[3] *Department of Electronic Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong, China*
[4] *College of Information Science and Engineering, Chongqing Jiaotong University, Chongqing* 400074, *China*

(Received 8 August 2016; revised manuscript received 11 October 2016; published online 29 November 2016)

In this paper, a novel image encryption algorithm is presented based on self-cited pixel summation. With the classical mechanism of permutation plus diffusion, a pixel summation of the plain image is employed to make a gravity influence on the pixel positions in the permutation stage. Then, for each pixel in every step of the diffusion stage, the pixel summation calculated from the permuted image is updated. The values from a chaotic sequence generated by an intertwining logistic map are selected by this summation. Consequently, the keystreams generated in both stages are dependent on both the plain image and the permuted image. Because of the sensitivity of the chaotic map to its initial conditions and the plain-image-dependent keystreams, any tiny change in the secret key or the plain image would lead to a significantly different cipher image. As a result, the proposed encryption algorithm is immune to the known plaintext attack (KPA) and the chosen plaintext attack (CPA). Moreover, experimental simulations and security analyses show that the proposed permutation-diffusion encryption scheme can achieve a satisfactory level of security.

Keywords: pixel summation, image encryption algorithm, intertwining logistic map, gravity, security

PACS: 05.45.Gg, 05.45.Vx, 05.45.–a **DOI:** 10.1088/1674-1056/26/1/010501

1. Introduction

Image is one kind of multimedia that has become increasingly important for communication in our daily life. It has many applications in different fields, for example, medical science, remote sensing, military, and traffic management. With the rapid development of Internet and wireless communication networks, where they are openly accessible, the protection of private images against intruders deserves much more attention than before. Traditional encryption algorithms such as AES (Advanced Encryption Standard), DES (Data Encryption Standard), and IDEA (International Data Encryption Algorithm) have been found to be unsuitable for image encryption[1–4] due to some intrinsic properties of images such as bulky data format, strong correlation between adjacent pixels, and high redundancy. Currently, a lot of image encryption algorithms[5–11] have been proposed to meet the challenge of security and efficiency requirements. Among these, many excellent features have been proved by chaotic maps,[12,13] for example ergodicity, unpredictability, nonlinearity, pseudo-randomness, and sensitivity to both system parameters and initial conditions, which provide better performance and act close to confusion and diffusion in cryptography.[14,15]

Recently, many new chaotic image encryption algorithms have been studied.[14,16,17] The architecture of permutation plus diffusion is the preferred model for designing chaos-based image encryption algorithms. On the one hand, permutation is used to exchange the pixel positions by a one-to-one mapping. As a result, the high correlation existing between adjacent pixels of the plain image can be greatly reduced. On the other hand, to alter the distribution of the gray values in the cipher image, diffusion operation is taken further to distribute all pixels. Sequentially, any tiny change (even just one bit) in a pixel will spread out to the other pixels and lead to a completely different output. A new hyperchaotic 2D map[18] was built from serpentine curve equations, which was employed in image encryption using permutation and diffusion. Then, in order to save time in the encryption process and achieve real-time communication, a new image encryption algorithm with combined permutation and diffusion was proposed in Ref. [19], in which the operation is accelerated by reducing the scanning effort for pixels. Fouda *et al.*[20] presented an encryption scheme with a single-round iteration, where the time consumption can be reduced due to the avoidance of real number arithmetic operations.

Unfortunately, some image encryption algorithms based on chaos were found to be insecure. For example, a chaotic encryption algorithm was proposed for color image in Ref. [21], where, with a permutation for row and column, together with a diffusion function for R, G, and B components respectively, a cipher image is obtained from a single-round iter-

*Project supported by the National Natural Science Foundation of China (Grant Nos. 61602124, 61273021, 11526057, and 11301091), the Natural Science Foundation of Guangdong Province, China (Grant Nos. 2016A030310333, 2015A030313614, and 2015A030313620), the Science & Technology Planning Projects of Zhanjiang City, China (Grant Nos. 2015B01098 and 2015B01051), the Project Foundation of Chongqing Municipal Education Committee of China (Grant No. KJ1500501), the Program for Scientific Research Start-up Funds of Guangdong Ocean University of China, and the Special Funding Program for Excellent Young Scholars of Guangdong Ocean University of China.

†Corresponding author. E-mail: guodongye@hotmail.com; guodongye@gmail.com

© 2017 Chinese Physical Society and IOP Publishing Ltd

ation. However, two problems were analyzed by Arroyo *et al.*,[22] i.e., non-exhaustive definition of the key space and low sensitivity to the change of the plain image. Actually, the cryptosystem[21] can be interpreted as a one-round SPN (substitution permutation network).[22] Moreover, using CPA (chosen-plaintext attack), it is proved that any one-round SPN can be successfully cryptanalyzed.[22–24] Then, DNA (Deoxyribonucleic acid) sequence operation and a hyper-chaotic system based image fusion encryption algorithm was presented in Ref. [25]. The whole encryption scheme is divided into two processes: shuffling the image pixels using the chaotic Chen system and changing the gray values using DNA operations. However, for a cipher image of size $p \times q$, Xie *et al.*[26] re-evaluated the security of the algorithm proposed in Ref. [25] and reported that less than $[\log_2(4pq)]/2+1$ chosen plain images are sufficient to successfully launch a CPA. Also, there are some other image encryption algorithms that were broken[27–30] using cryptanalysis techniques similar to that used in Refs. [22] and [24].

Based on cryptanalysis, the common security defects of the existing image encryption schemes are summarized as follows: (i) small key space, (ii) keystream generated independent of the plain image, (iii) a single-round iteration of the permutation plus diffusion structure. To solve these problems, an image encryption algorithm using self-cited summation is proposed in this paper. First of all, a three-dimensional intertwining logistic map is employed to enlarge the key space. Then, in both permutation and diffusion stages, the keystreams generated are dependent on the pixel summation of the plain image and the permuted image, respectively. Third, three rounds of iteration are adopted in the proposed algorithm. Then, detailed analysis will be performed to verify the security of the new design. The rest of this paper is organized as follows. Section 2 describes the mathematical model of the proposed algorithm based on pixel summation. Numerical experiments are given in Section 3 to demonstrate the performances. Security analyses are presented in Section 4 to show the efficiency of the proposed design. Finally, Section 5 concludes the paper.

2. The self-cited pixel summation based algorithm

2.1. Pre-knowledge

2.1.1. Intertwining logistic map

To generate chaotic sequences for permutation and diffusion operations, a nonlinear chaotic map or system is usually employed in a chaos-based image encryption algorithm. Because of the simplicity and low computation complexity, the logistic map[16,28] has been widely used in recent years. It is defined by

$$x_{i+1} = ux_i(1-x_i), \tag{1}$$

where the iterated values $x_i \in (0,1)$ and control parameter $u \in (0,4]$. The orbit of the map will be chaotic if $u \in (3.564,4]$ (see Fig. 1 for the setting of initial condition and control parameter). However, on the one hand, there is only one initial condition and one control parameter in the one-dimensional logistic map (1). Hence, it cannot well resist the brute-force attack under the current computation environment. One the other hand, the logistic map has some weaknesses,[31,32] for example blank windows, weak keys, and uneven distribution of sequences. Therefore, it is not secure enough to only use the logistic map in any image encryption algorithm.

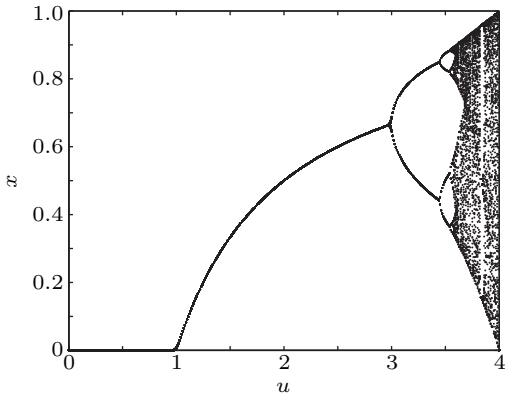


Fig. 1. Bifurcation diagram of the logistic map.

In order to solve the problems of the logistic map (1), a new three-dimensional chaotic intertwining logistic map[31,32] is proposed as follows:

$$\begin{cases} x_{i+1} = [\mu \times \lambda_1 \times y_i \times (1-x_i) + z_i] \bmod 1, \\ y_{i+1} = [\mu \times \lambda_2 \times y_i + z_i/(1+x_{i+1}^2)] \bmod 1, \\ z_{i+1} = [\mu \times (x_{i+1} + y_{i+1} + \lambda_3) \times \sin(z_i)] \bmod 1, \end{cases} \tag{2}$$

where the control parameters are set to be $\mu \in (0, 3.999]$, $|\lambda_1| > 33.5$, $|\lambda_2| > 37.5$, $|\lambda_3| > 35.7$, and mod is the modulus operator. Compared with the logistic map, an intertwining logistic map has four parameters and three initial conditions with a uniform distribution of chaotic sequences but no security issues as presented in the logistic map. More about the properties of the intertwining logistic map can be found in Refs. [31] and [32]. Figure 2 shows the chaotic behavior of the intertwining logistic map with initial conditions $x_0 = 0.0512$, $y_0 = 0.1616$, $z_0 = 0.7277$, and control parameters $\mu = 3.8986$, $\lambda_1 = 34.5061$, $\lambda_2 = 38.003$, $\lambda_3 = 39.3236$. It is observed that the orbits are different if any tiny change, for example 10^{-14}, occurred in any initial conditions.

2.1.2. Gravity model

In physics, there is gravity between any two bodies in nature. The value of the gravity F can be calculated by

$$F = G\frac{m_1 m_2}{r^2}, \tag{3}$$

where m_1 and m_2 are the mass of two different objects, G is the gravitational constant ($\approx 6.67 \times 10^{-11}$ Nm^2/kg^2), and r represents the distance of these two bodies. From Eq. (3), one can easily see that $F_1 > F_2$ in Fig. 3 with three identical bodies.

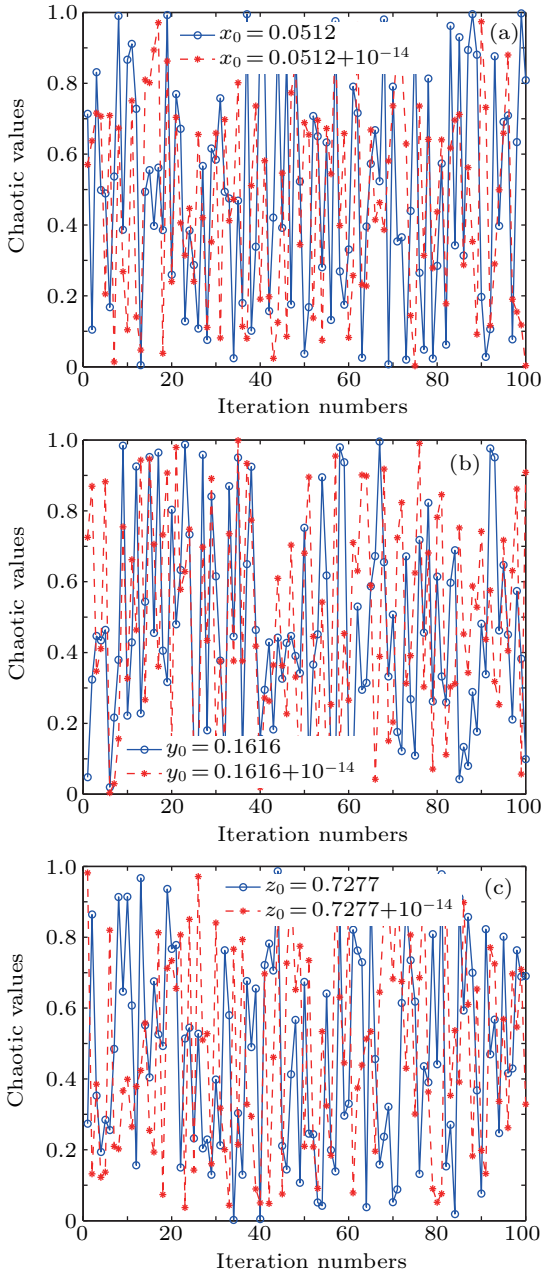


Fig. 2. (color online) Chaotic behavior of the intertwining logistic map in x coordinate (a), y coordinate (b), and z coordinate (c).

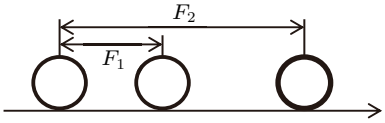


Fig. 3. Gravity between three bodies with the same mass.

2.2. Self-cited pixel summation based permutation operation

Suppose that the pre-encrypted gray image is A of size $p \times q$. A color image can be divided into three channels R (red), G (green), and B (blue), each of which can be treated as a matrix corresponding to a gray-scale image. Because there exist some strong correlations between adjacent pixels in natural images, we try to decorrelate it to the ideal value (i.e., zero) in the new designed image encryption algorithm. Here, permutation of pixel positions is taken to implement the confusion effect. A good re-location effect should make the permuted image look like random noise.

To frustrate the common CPA and KPA (known plain attack), the keystream generated in the encryption process should be dependent on the plain image. It is noted that the permutation function is just to re-arrange the pixel positions but does not change the pixel values. That is, the pixel summation of the permuted image is equal to that of the plain image. Combining it with the gravity, we firstly compute the summation sa of the plain image and take it as the image quality (denoted as a body S) to influence itself in the permutation stage. For example, figure 4 gives an explanation of the Lena image in size of 256×256. As to all pixels in the plain image, their new positions in the permuted image will be changed by the body S with pixel summation, $sa = \Sigma A = 6434179$, according to the gravity therein.

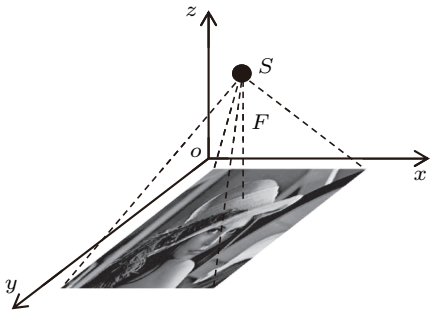


Fig. 4. Gravity for the Lena image.

Since there is a distance between any two bodies to produce the gravity force, we assume that the plain image is put in the three-dimensional Cartesion coordinates, and the top–left corner of the image is located at coordinate (1,1,0). To simplify the ergodic permutation, each row (or each column) is treated as a unit in this new method. Pixel summation is treated as a body and is supposed to be put into the position of (1,1, (sa mod 256) + 100) in the space Cartesian coordinate system. So the distance d between the body S and the first pixel $A(1,1)$ is equal to $d = (sa \bmod 256) + 100$. Let m_1 in Eq. (3) be the pixel summation of the plain image, while m_2 be the number of row (column) in a square. Here, to avoid the black image attack, we add one into m_1, i.e., $m_1 = sa + 1$. The distance is designed as

$$r = \begin{cases} \sqrt{(i-1)^2 + (j_0-1)^2 + d^2}, & i = 1,2,\ldots,p, \\ \sqrt{(i_0-1)^2 + (j-1)^2 + d^2}, & j = 1,2,\ldots,q, \end{cases} \tag{4}$$

where i and j are the numbers of row and column respectively, and i_0 and j_0 are two constants to distinguish the row and column. So, considering the circular permutation in the row and

column, the body S will have a force on the i-th row by strength Fr_i in Eq. (5). Similarly, the case of the j-th column can be easily calculated as follows:

$$Fr_i = G\frac{(sa+1)\times i^2}{(i-1)^2+(j_0-1)^2+d^2}, \quad i=1,2,\ldots,p, \quad (5)$$

$$Fc_j = G\frac{(sa+1)\times j^2}{(i_0-1)^2+(j-1)^2+d^2}, \quad j=1,2,\ldots,q. \quad (6)$$

With initial conditions x_0, y_0, and z_0 iterated with the intertwining logistic map, we get a chaotic sequence $\{x_1,y_1,z_1,x_2,y_2,\ldots\}$ after some rounds of iteration. To increase the randomness, some former iterated values are thrown away. Then, two sets $H=\{h_1,h_2,h_3,\ldots,h_p\}$ and $L=\{l_1,l_2,l_3,\ldots,l_q\}$ are obtained for row and column permutation respectively. However, before being applied to the first stage of permutation, a pre-process should be done as below for all elements and then the sets H and L are updated

$$\begin{cases} h_i = \lceil (h_i + Fr_i)\times 10^{14}\rceil \bmod q, & i=1,2,\ldots,p, \\ l_j = \lceil (l_j + Fc_j)\times 10^{14}\rceil \bmod p, & j=1,2,\ldots,q, \end{cases} \quad (7)$$

where $\lceil h\rceil$ returns the smallest integer greater than or equal to h. Using the above updated H and L, circular shifting operation can be applied to each row and column in the plain image. That is, the i-th row is moved to the right by h_i units and circled on the left ($i=1,2,\ldots,p$), while the j-th column is moved to the bottom by l_j units and circled on the top ($j=1,2,\ldots,q$). As a result, a permuted image B is formed when all the permutation operations for rows and column are finished.

2.3. Self-cited pixel summation based diffusion operation

In order to resist statistical attack, the distribution of gray values in the cipher image should be different from that of the plain image. Usually, diffusion operation is taken after the permutation stage to spread the influence of any tiny change in the plain image to produce a significantly different cipher image. As we know, if the cipher image shows a fairly uniform distribution, then it can increase the difficulty for differential attack. Of course, the keystream generated dependence on the permuted image is very important for resisting KPA and CPA.

Firstly, all the pixels in B are arranged into a vector $\boldsymbol{b}=\{b_1,b_2,\ldots,b_{p\times q}\}$ in the order from top to bottom and then from left to right. Again, the pixel summation sb of the permuted image (i.e., $sb=\Sigma b$) in the last permutation stage is used as an image quality for the permuted image in diffusion. Here, sb is employed to select chaotic values produced from the intertwining logistic map with another key $\bar{x}_0$, $\bar{y}_0$, and $\bar{z}_0$. Let Len be the length of the permuted image B, i.e., $\text{Len}=p\times q$.

Assume that a chaotic sequence $\{\bar{x}_1,\bar{y}_1,\bar{z}_1,\bar{x}_2,\bar{y}_2,\ldots\}$ is produced using the new secret keys iterated by the intertwining logistic map. Then, we select Len values from it and form a set $\boldsymbol{K}=\{k_1,k_2,\ldots,k_{\text{Len}}\}$. However, before being used, a pre-process is taken on K as characterized below, which makes all the elements in K fall into 0–255,

$$k_i = \lceil k_i\times 10^{14}\rceil \bmod 256, \quad i=1,2,\ldots,\text{Len}. \quad (8)$$

For each pixel in vector $\boldsymbol{b}$, the diffusion process is implemented as below, for the i-th pixel in the form of pixel-by-pixel ($i=1,2,\ldots,\text{Len}$)

$$\begin{cases} sb = sb - b_i, \\ t = sb \bmod v + 1, \quad \text{here } 0<v\le \text{Len}, \\ c_i = b_i \dot{+} c_{i-1} \dot{+} k_i \dot{+} [(sb+k_t) \bmod 256], \end{cases}, \quad (9)$$

where sb is self-updated for each pixel as shown by the first formula, c_{i-1} and c_i represent the former and current pixels in the cipher image respectively, c_0 is a constant, and $\dot{+}$ denotes the modular operation under 256. The value v employed here is to let the keystream used in the diffusion process be related to the permuted image. To save time, v should not be too big. Here, $v=10$ is set in our method. From Eq. (9), one can see that the pixel summation of the permuted image has an influence on the produced chaotic sequences. Therefore, various keystreams for different permuted images in diffusion stages can efficiently resist the CPA and KPA. After re-shaping again the vector $\{c_1,c_2,\ldots,c_{\text{Len}}\}$ into a matrix C, the cipher image can be obtained.

Note that the pixel summation of the plain image is used twice in the encryption processes, as can be seen from Eqs. (7) and (9). So, if any one pixel in the plain image has changed slightly, the corresponding cipher image will deviate dramatically. That is, the algorithm is very sensitive to the plain image. Moreover, due to the properties of the chaotic map, the algorithm also has high sensitivity to the initial conditions and control parameters. As a result, the proposed image encryption algorithm should be secure when facing the black image attack, CPA, and KPA, as further analyzed later.

2.4. Encryption and decryption diagrams

The architecture of the proposed image encryption algorithm is based on a permutation-diffusion mechanism. The encryption process is outlined in the following steps with encryption diagram and decryption diagram shown in Figs. 5 and 6, respectively. To enhance security, we suggest iterating the following steps for 3 rounds.

Step 1 Read the plain image with size $p\times q$, and denote it by matrix $\boldsymbol{A}$.

Step 2 Compute the pixel summation sa and the distance d.

Step 3 Calculate the strengths Fr and Fc by Eqs. (5) and (6).

Step 4 Iterate the intertwining logistic map with a group of keys x_0, y_0, z_0, and get random sets H and L.

Step 5 Pre-process H and L using Eq. (7).

Step 6 Do the circular shifting operation for $\boldsymbol{A}$ and get the permuted image $\boldsymbol{B}$.

Step 7 Arrange $\boldsymbol{B}$ into a vector $\boldsymbol{b}$, and compute the length Len.

Step 8 Iterate the intertwining logistic map with another group of keys $\bar{x}_0$, $\bar{y}_0$, $\bar{z}_0$, and get K by Eq. (8).

Step 9 Compute the pixel summation of vector $\boldsymbol{b}$, and then do the diffusion operation by Eq. (9).

Step 10 Reshape the vector c into a matrix C of size $p \times q$, and obtain the cipher image.

As we can observe from the above steps, the computation complexity of each step is equal to or smaller than $O(pq)$, so the total computation complexity of the encryption should be $3O(pq)$ for three rounds, which is linear to the size of the image to be encrypted. Since the algorithm is symmetric, so the complexity for decryption is the same as that of encryption.

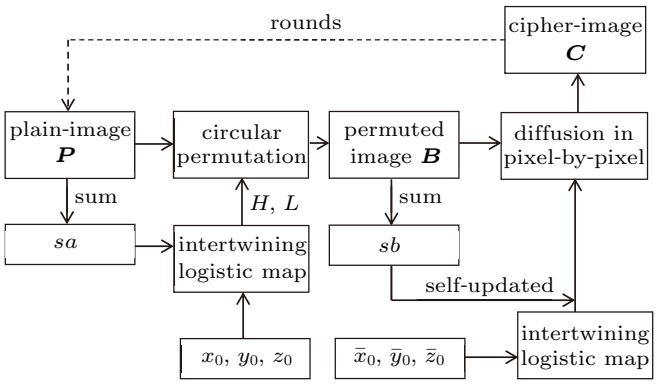


Fig. 5. Diagram of the encryption process.

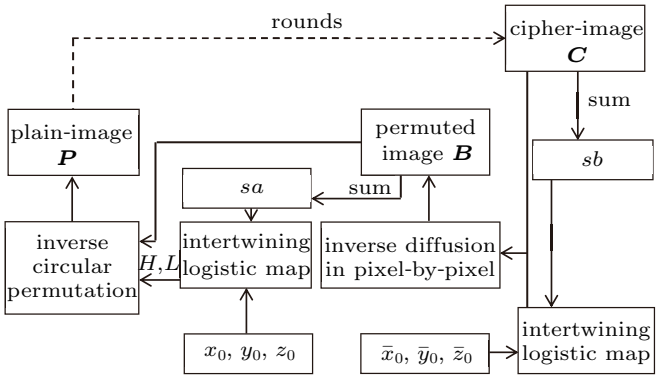
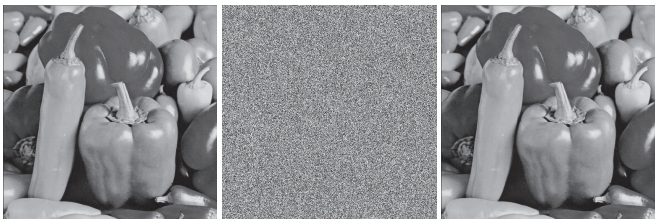


Fig. 6. Diagram of the decryption process.

3. Numerical simulations

This section presents some experimental results of the proposed encryption algorithm. All the simulations are implemented by software Matlab R2011b on a platform of Windows 7 with Intel(R) Core(TM) i3-2350, 2.30 GHz CPU. Peppers of size 512×512, as depicted by Fig. 7(a) are used for the test. Figure 7(b) shows the cipher image by applying our method with three rounds. Here, the initial conditions are randomly chosen as $x_0 = 0.3435$, $y_0 = 0.0818$, $z_0 = 0.7007$, $\bar{x}_0 = 0.1424$, $\bar{y}_0 = 0.6636$, $\bar{z}_0 = 0.0115$, while the control parameters are set to be $\mu = 3.9013$, $\lambda_1 = 35.2636$, $\lambda_2 = 38.2231$, $\lambda_3 = 37.109$ for the intertwining logistic map. With all correct keys and parameters for decryption, the original plain image can be recovered as displayed in Fig. 7(c).



(a) (b) (c)

Fig. 7. Peppers: (a) plain image, (b) cipher image, (c) correct decrypted image.

4. Security analyses

4.1. Key space analysis

To resist the brute-force attack, the basic requirement for the key space is $2^{100} \approx 10^{30}$.[11] In our method, the keys are composed of six initial conditions: x_0, y_0, z_0, $\bar{x}_0$, $\bar{y}_0$, and $\bar{z}_0$. Hence, the key space can reach as large as 10^{84} considering all combinations if the computer precision is set to 10^{-14}. Therefore, the brute-force attack is infeasible for the proposed algorithm. Table 1 shows a comparison of key space contained in some recent references.

Table 1. Comparison of key spaces.

References	[13]	[25]	[33]	Ours
Key space	10^{77}	10^{70}	10^{38}	10^{84}

4.2. Key and plain-image sensitivity analyses

A good image encryption algorithm should keep high sensitivity to every secret key to against CPA and linear cryptanalysis.[20] Here, the Peppers image is selected again to do the test. Figure 8(a) shows that a wrong decrypted image can only be gained when a small change with 10^{-14} is added into the key x_0. Similarly, any tiny change in other keys such as y_0, z_0, $\bar{x}_0$, $\bar{y}_0$, and $\bar{z}_0$, will also produce a completely wrong decryption as shown in Figs. 8(b), 8(c), 8(d), 8(e), and 8(f), respectively. Hence, high sensitivity to keys is achieved in the proposed algorithm. Moreover, as to the control parameters, one can see in Figs. 8(g), 8(h), and 8(i) that our method is also very sensitive to them.

By the requirement of the avalanche effect for encryption schemes, any change in a pixel of the plain image, even just one-bit, should produce a totally different cipher image. To

show the efficiency of our new algorithm, figure 9 displays the results for the Lena image, where figure 9(b) is the cipher image from the plain image in Fig. 9(a), while figure 9(c) gives another cipher image also from Fig. 9(a) but just with one-bit change. Figure 9(d) displays the difference between them. Clearly, the proposed algorithm also has a high plain-image sensitivity.

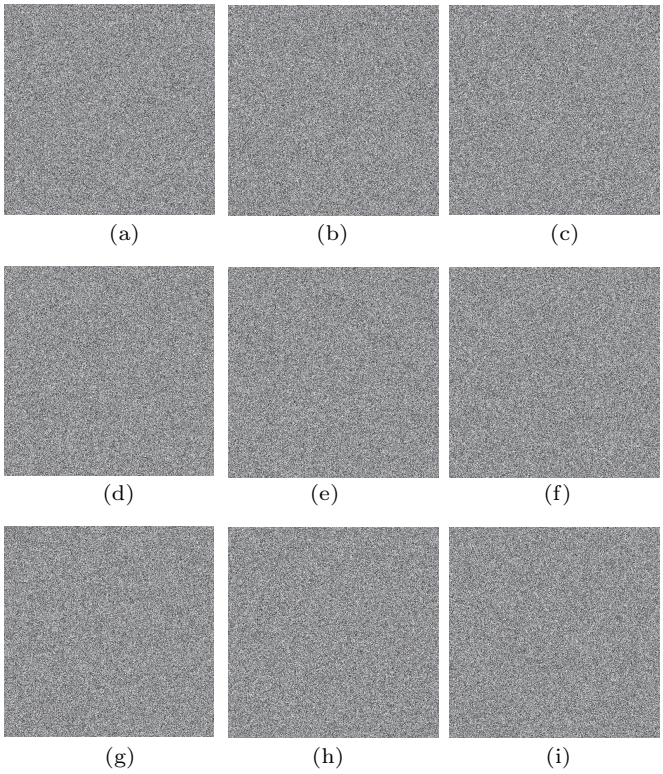


Fig. 8. Wrong decrypted image for Peppers with (a) x_0+10^{-14}, (b) y_0+10^{-14}, (c) z_0+10^{-14}, (d) $\bar{x}_0+10^{-14}$, (e) $\bar{y}_0+10^{-14}$, (f) $\bar{z}_0+10^{-14}$, (g) $\mu+10^{-14}$, (h) λ_1+10^{-14}, (i) λ_2+10^{-14}.

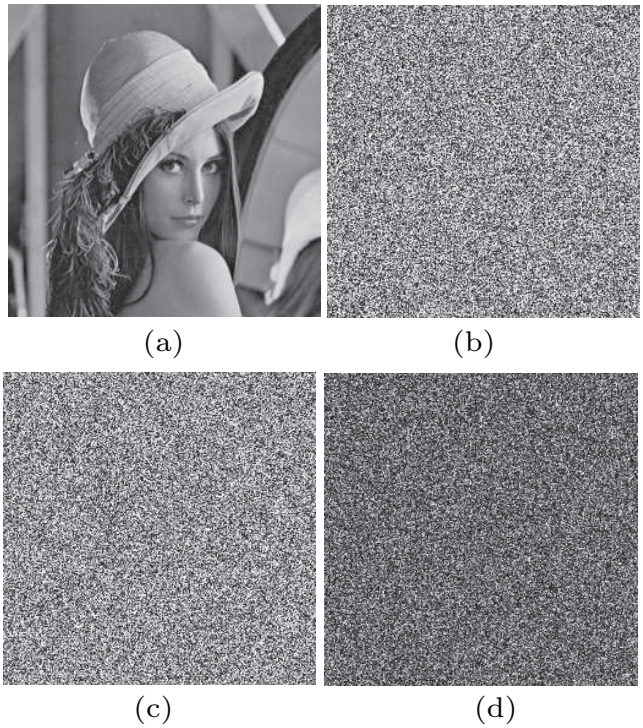


Fig. 9. Plain-image sensitivity test for Lena: (a) plain image, (b) first cipher image, (c) second cipher image, (d) difference between (b) and (c).

Furthermore, NPCR (number of pixels change rate) and UACI (unified averaged changing intensity) are commonly taken to measure the influence of any tiny change in one pixel of the plain image on the cipher image, which are calculated by

$$\mathrm{NPCR} = \frac{\sum_{ij} D(i,j)}{p \times q} \times 100\%, \tag{10}$$

$$\mathrm{UACI} = \frac{1}{p \times q}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\%, \tag{11}$$

where $p \times q$ is the image size, and

$$D(i,j) = \begin{cases} 0, & C_1(i,j) = C_2(i,j), \\ 1, & \text{otherwises}. \end{cases}$$

As to different images, the results of NPCR and UACI are listed in Table 2. Theoretically, the value of NPCR is 99.6094%, while the value of UACI is 33.4635%.[34] Hence, all the values in Table 2 are close to the ideal cases.

Table 2. Results of UACI and NPCR.

Directions	Lena	Boat	Barb	Peppers
NPCR	99.6597	99.6105	99.6273	99.6044
UACI	33.5066	33.5005	33.5160	33.5025

4.3. Histogram analysis

The image histogram shows the distribution of every pixel from 0 to 255 for a gray image. An efficient image encryption algorithm should make the histogram of the cipher image be fairly uniform and be significantly different from its corresponding plain image. To test the robustness against statistical attack, figure 10 displays the results for the plain images Baboon and Barb. It is obvious that the statistical attack is difficult for the proposed method. Moreover, the black image of size 512×512 shown in Fig. 11 is also tested by our method, which highlights the efficiency and the security of the proposed algorithm.

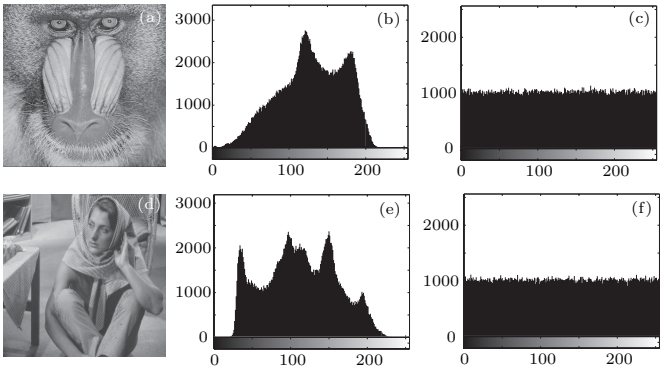


Fig. 10. Histogram test for Baboon: (a) plain image, (b) histogram of plain image, (c) histogram of cipher image; Barb: (d) plain image, (e) histogram of plain image, (f) histogram of cipher image.

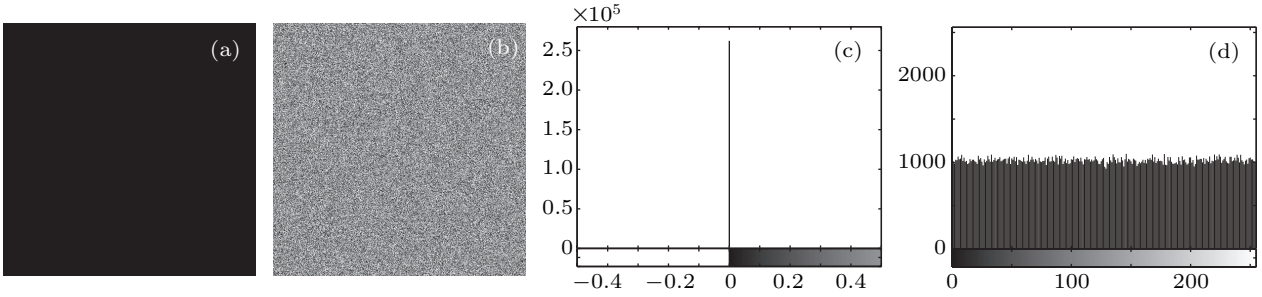


Fig. 11. Histogram test for Black image: (a) plain image, (b) cipher image, (c) histogram of plain image in panel (a), (d) histogram of cipher image in panel (b).

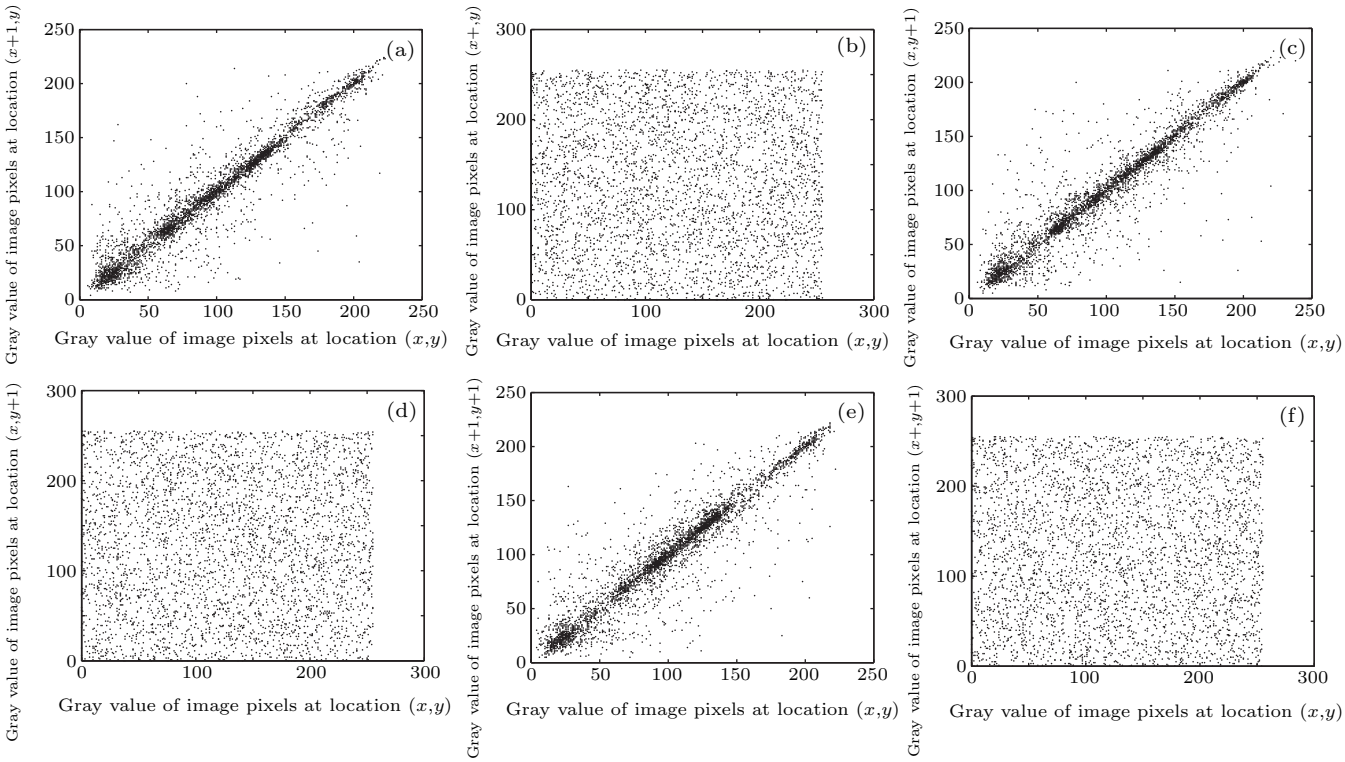


Fig. 12. Correlation test for Lena image. Horizontally: (a) plain image, (b) cipher image; vertically: (c) plain image, (d) cipher image; diagonally: (e) plain image, (f) cipher image.

4.4. Correlation analysis

For any natural image, high correlations can be found between pixels and their adjacent pixels. Usually, the following equation is calculated to get the correlation coefficients for both the plain image and its corresponding cipher image:

$$r_{uv} = \frac{M\Sigma_{i=1}^{M}(u_i v_i) - \Sigma_{i=1}^{M} u_i \Sigma_{i=1}^{M} v_i}{\sqrt{(M\Sigma_{i=1}^{M} u_i^2 - (\Sigma_{i=1}^{M} u_i)^2)(M\Sigma_{i=1}^{M} v_i^2 - (\Sigma_{i=1}^{M} v_i)^2)}}, \quad (12)$$

where u_i and v_i represent the gray values of two adjacent pixels. Hence, the correlation coefficients are close to zeros in the cipher image.

Table 3. Correlation coefficients for Lean image.

Directions	Vertically	Horizontally	Diagonally
Plain-image	0.9771	0.9631	0.9490
Cipher-image	0.0925	0.0430	0.0533

Here, Lena (see Fig. 9(a)) is chosen randomly to do the test with the results given in Table 3 and displayed in Fig. 12 horizontally, vertically, and diagonally, respectively.

4.5. KPA and CPA analyses

KPA and CPA are usually taken by cryptanalysts for security evaluation.[22,26–30] In the proposed algorithm, pixel summation is considered in both permutation and diffusion stages. On the one hand, chaotic sequences H and L in Step 5 are generated to be dependent on the plain image for circular permutation. On the other hand, the keystream k_t in Eq. (9) is computed by pixel summation sb of the permuted image, and so there is a plain image dependency in the diffusion operation. Therefore, our method has a good feedback mechanism to the plain image. Although some special images may be chosen, no useful information would be supplied to the attacker. As a

result, the cryptanalysis methods such as Refs. [22], [26]–[30] will not work properly to attack our new algorithm.

4.6. Comparisons

In this section, to show the better performance of the proposed algorithm, some comparisons are given in Tables 4–6. Table 4 is a comparison with Ref. [15] by UACI and NPCR tests using 256×256 Lena image, while Tables 5 and 6 show the comparison of speed analysis (unit: second) with Ref. [20] using different rounds of encryption. Validated by these comparison results, it is concluded that the proposed image encryption algorithm can promise a better performance over the others.

Table 4. Comparisons of UACI and NPCR.

Pixel positions	(9,163)	(250,218)	(66,79)	(161,100)
NPCR in [15]	99.9176	99.6368	99.7604	99.7162
UACI in [15]	46.0224	34.6713	29.7939	37.5332
NPCR in ours	99.6124	99.5972	99.5926	99.6277
UACI in ours	33.4021	33.4126	33.4805	33.5988

Table 5. Comparisons of speed for image size 512×512.

Methods	One round	Two rounds	Three rounds	Four rounds
Ref. [20]	0.1888 s	0.3791 s	0.5335 s	0.7410 s
Ours	0.1139 s	0.2278 s	0.3572 s	0.4025 s

Table 6. Comparisons of speed for image size 1024×1024.

Methods	One round	Two rounds	Three rounds	Four rounds
Ref. [20]	0.8065 s	1.6645 s	2.4945 s	3.3712 s
Ours	0.3884 s	0.7613 s	1.1138 s	1.4914 s

5. Conclusions

A permutation-diffusion structure and self-cited pixel summation based image encryption algorithm have been presented in this paper. Image pixels are combined to produce the keystreams used in both permutation and diffusion operations. As a result, any tiny change in the secret keys or the plain image will lead to a significant alternation in the cipher image. Security analyses including key space, key sensitivity, plain-image sensitivity, histogram, correlation, KPA, and CPA demonstrate the high security and efficiency of the proposed encryption algorithm. Moreover, our method can be applied to any size of image in the gray level and can also be easily extended to color images with three components R, G, and B. Therefore, the new image encryption algorithm is suitable for real-time communications over public channels.

Acknowledgements

The authors would like to thank the two anonymous referees for valuable suggestions which are very useful in improving the quality of this manuscript. Also, thanks to professor Guanrong Chen at City University of Hong Kong for correcting the expressions throughout the paper.

References

[1] Zhang X Y, Zhang G J, Li X, Ren Y Z and Wu J H 2016 *Chin. Phys. B* **25** 054201
[2] Norouzi B, Mirzakuchaki S, Seyedzadeh S M and Mosavi M R 2014 *Multimed. Tools Appl.* **71** 1469
[3] Wu Y, Zhou Y C, Noonan J P and Agaian S 2014 *Inform. Sci.* **264** 317
[4] Zhou G M, Zhang D X, Liu Y J, Yuan Y and Liu Q 2015 *Neurocomputing* **169** 150
[5] Xiao D, Cai H K and Zheng H Y 2015 *Chin. Phys. B* **24** 060505
[6] Pareek N K, Patidar V and Sud K K 2006 *Image Vision Comput.* **24** 926
[7] Seyedzadeh S M, Norouzi B and Mirzakuchaki S 2014 *J. Sys. Software* **97** 128
[8] Zhou N R, Hua T X, Gong L H, Pei D J and Liao Q H 2015 *Quantum Inf. Proces.* **14** 1193
[9] Liao X F, Lai S Y and Zhou Q 2010 *Signal Proces.* **90** 2714
[10] Zhou N R, Zhang A D, Zheng F and Gong L H 2014 *Opt. Laser Technol.* **62** 152
[11] Alvarez G and Li S J 2006 *Int. J. Bifur. Chaos* **16** 2129
[12] Wong K W, Kwok B S H and Law W S 2008 *Phys. Lett. A* **372** 2645
[13] Hua Z Y, Zhou Y C, Pun C M and Chen C L P 2015 *Inform. Sci.* **297** 80
[14] Fridrich J 1998 *Int. J. Bifur. Chaos* **8** 1259
[15] Ye R S 2014 *Fund. Inform.* **133** 87
[16] Anees A, Siddiqui A M and Ahmed F 2014 *Commun. Nonlinear Sci. Numer. Simul.* **19** 3106
[17] Ye G D 2011 *Imaging Sci. J.* **59** 183
[18] Boriga R, Dăscălescu A C and Priescu I 2014 *Signal Proces. Image Commun.* **29** 887
[19] Wang Y, Wong K W, Liao X F and Chen G R 2011 *Appl. Soft Comput.* **11** 514
[20] Fouda J S A E, Effa J Y, Sabat S L and Ali M 2014 *Commun. Nonlinear Sci. Numer. Simul.* **9** 578
[21] Wang X Y, Teng L and Qin X 2012 *Signal Proces.* **92** 1101
[22] Arroyo D, Diaz J and Rodriguez F B 2013 *Signal Proces.* **93** 1358
[23] Li C Q, Zhang L Y, Ou R, Wong K W and Shu S 2012 *Nonlinear Dyn.* **70** 2383
[24] Zhang L Y, Liu Y S, Wong K W, Pareschi F, Zhang Y S, Rovatti R and Setti G 2015 arXiv:1512.09263
[25] Zhang Q, Guo L and Wei X P 2013 *Optik* **124** 3596
[26] Xie T, Liu Y S and Tang J 2014 *Optik* **125** 7166
[27] Zhang Y S, Xiao D, Wen W Y and Li M 2014 *Nonlinear Dyn.* **76** 1645
[28] Li C Q, Li S J, Asim M, Nunez J, Alvarez G and Chen G R 2009 *Image Vision Comput.* **27** 1371
[29] Rhouma R, Solak E and Belghith S 2010 *Commun. Nonlinear Sci. Numer. Simul.* **15** 1887
[30] Zhang Y S, Xiao D, Wen E Y and Li M 2014 *Multimed. Tools Appl.* **73** 1885
[31] Sam I S, Devaraj P and Bhuvaneswaran R S 2012 *Nonlinear Dyn.* **69** 1995
[32] Wang X Y and Luan D P 2013 *Commun. Nonlinear Sci. Numer. Simul.* **18** 3075
[33] Murillo-Escobar M A, Cruz-Hernández C, Abundiz-Pérez F, López-Gutiérrez R M and Acosta Del Campo O R 2015 *Signal Proces.* **109** 119
[34] Dăscălescu A C and Boriga R E 2013 *Nonlinear Dyn.* **74** 307