Nonlinear Dyn
DOI 10.1007/s11071-011-0006-6

ORIGINAL PAPER

A new image encryption method: parallel sub-image encryption with hyper chaos

Omid Mirzaei · Mahdi Yaghoobi · Hassan Irani

Received: 10 November 2010 / Accepted: 28 February 2011

Abstract A new image encryption scheme, based on a total shuffling and parallel encryption algorithm is proposed in this paper. Two chaotic systems have been used in the encryption algorithm to confuse the relationship between the plain-image and the cipher-image. To make the encryption procedure more confusing and complex, the plain-image is first divided into 4 sub-images and then the position of each sub-image is changed pseudo-randomly according to a logistic map. Next, a total shuffling matrix is used to shuffle the position of pixels in the whole image and then sub-images are encrypted simultaneously in a parallel manner. The experimental results on USC data base demonstrate that the proposed encryption algorithm has a low time complexity and has the advantages of large key space and high security. Moreover, the robustness of this locally encryption method is much more in contrast with other encryption schemes and the distribution of gray values has a random-like behavior in the encrypted image.

O. Mirzaei (✉) · M. Yaghoobi · H. Irani
Engineering Department, Artificial Intelligence Group, Islamic Azad University, Mashhad Branch, Iran
e-mail: omid.mirzaei@gmail.com

M. Yaghoobi
e-mail: yaghobi@mshdiau.ac.ir

H. Irani
e-mail: yoonesirani@gmail.com

Published online: 17 March 2011

Keywords Parallel image encryption · Hyper chaos · Image shuffling · Image division

1 Introduction

Rapid developments in digital image processing and widespread dissemination of digital multimedia data over the internet have made us to protect this vital information against illegal copying and distribution. To reach this goal, many new encryption schemes have been proposed [1–5]. Comparing with conventional encryption algorithms, chaos-based ones have suggested more secure and fast encryption methods [6–10].

The first chaos-based encryption algorithm was proposed in 1989 [11]. Since then, many researchers have investigated and analyzed many chaos-based encryption algorithms, these works all have been motivated by the chaotic properties such as the sensitive dependence on initial conditions and systems parameters, pseudo-random property, nonperiodicity, and topological transitivity.

It is crucial for a good encryption algorithm to be sensitive to the cipher keys, and the key space should be large enough to make brute-force attacks infeasible [12]. Recently, a fast chaotic cryptographic scheme based on iterating a logistic map was suggested [13], in which no random numbers need to be generated and the look-up table used in the cryptographic process is updated dynamically. After that, a two-dimensional

chaotic cat map was generalized to 3D for designing a real-time secure symmetric encryption scheme to confuse the relationship between the cipher-image and the plain one [14]. Also recently, the authors in [15] thought that the algorithm for encoding binary images using a one-dimensional chaotic map [16] is not secure enough to overcome the drawbacks such as small key space and weak security of a one-dimensional chaotic map, a nonlinear algorithm is proposed in [17], which shows high-level security and acceptable efficiency.

A new multilevel image encryption scheme is suggested in this paper, different from others being proposed so far. The plain-image is first divided into 4 sub-images, and then these blocks are disordered to make a disordered image. Then a total shuffling matrix is used to shuffle the position of the pixels in the whole image and the states combination of two chaotic systems are used to change the gray values of the plain-image. After this, the whole image is encrypted simultaneously in a parallel manner.

The rest of this paper is organized as follows. Section 2 explains about the division of plain-image into sub-images and discusses about the effects of sub-images number and the change in their position in complexity of encryption procedure. It also introduces Lorenz and Chen's chaotic systems and presents image total shuffling algorithm and new image encryption scheme. Section 3 describes some simulation outcomes. All security analysis are given is Sect. 4, and finally, Sect. 5 concludes the paper and suggests some future improvements.

2 Division, shuffling and the proposed encryption algorithm for a plain image

2.1 Division of plain-image into sub-images

Division is the first step in the proposed encryption algorithm. Firstly, the whole image is divided into 4 equal blocks and then a pseudo-random array containing 4 pseudo-random numbers is used to disorder the initial arrangement of image blocks. These 4 values are obtained from the below logistic map:

$$x_{n+1} = 4x_n(1 - x_n) \tag{1}$$

For a given x_0, after doing some iterations a new x_0 is derived, then let:

$$\text{rand} = \left[\text{mod}\left(x_0 \times 10^{13}, 4\right)\right] + 1 \tag{2}$$

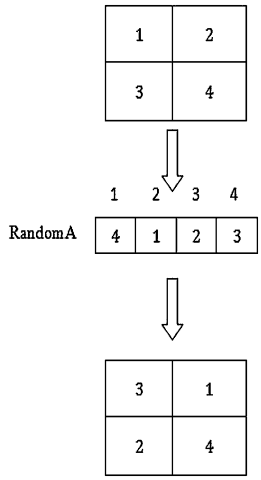

Fig. 1 Division and disordering of original image blocks

Continue to do the iteration of the logistic map and do (2) until we get 4 different data which are all between 1 and 4.

This kind of procedure will make the encryption operation more confusing and complex as it adds one extra step to the encryption process, and furthermore the length of the key will become longer. This process can be more understood from Fig. 1.

As you can see in Fig. 1, the initial image blocks become totally disordered using the pseudo-random array "RandomA." The disordered image is obtained as follows.

First, the block number "1" is changed with block number "4." Next, the block number "2" is changed with block number "1." Then the block number "3" is changed with block number "2" and finally the block number "4" is changed with block number "3."

Doing this procedure, we apply the first complexity to our encryption scheme that makes it more robust against widespread attacks.

2.2 Generation of image total shuffling matrix

An image total shuffling matrix is used in order to shuffle the position of the pixels in the plain-image and to disturb the high correlation among adjacent pixels and weaken this strong correlation among them. Without loss of generality, we assume that the dimension of the plain-image is $M \times N$ and the position matrix of pixels is $P_{i,j}, i = 0, 1, \ldots, M-1; j = 0, 1, \ldots, N-1$. The generation procedure of shuffling matrix is described as follows:

(1) For the logistic map $x_{n+1} = 4x_n(1 - x_n)$ and a given x_0, do some iterations, a new x_0 is derived,

then let:

$$l = \mathrm{mod}\left(x_0 \times 10^{13}, M\right) \tag{3}$$

Obviously $l \in [0, M-1]$

(2) Continue to do the iteration of the logistic map and do (3) until we get M different data which are all between 0 and $M-1$; these data can be recorded in the form of $\{h_i, i = 1, 2, \ldots, M\}$, where $h_i \neq h_j$ if $i \neq j$. Then rearrange the row of matrix $P_{i,j}$ according to $\{h_i, i = 1, 2, \ldots, M\}$, that is, move the h_1 row to the first row, h_2 row to the second row, therefore, a new position matrix $P^h_{i,j}$ is generated based on the transformation.

Now, we will produce column shuffling matrix by shuffling the new matrix $P^h_{i,j}$ column by column. The process is presented next:

(3) Use the present x_0 to do the iteration of Logistic map and then let:

$$l = \mathrm{mod}\left(x_0 \times 10^{13}, N\right) \tag{4}$$

It can be easily seen that $l \in [0, N-1]$

(4) Continue to do the iteration of the logistic map and do (4) until we get N different data which are all between 0 and $N-1$; these data can be expressed $\{l_i, i = 1, 2, \ldots, N\}$, where $l_i \neq l_j$ if $i \neq j$. Then rearrange the data of every column for the first row of matrix $P^h_{i,j}$ according to $\{l_i\}$, that is, move the l_1 column to the first column, l_2 column to the second column, thus a new column transformation of the first row of matrix $P^h_{i,j}$ is generated.

From the second row until the last row of matrix $P^h_{i,j}$, do the same column transformation in the same way as the second step, thus a new image total shuffling matrix $P^{hl}_{i,j}$ is given, and if N and M are not very big, the algorithm has lower time complexity, which can be summarized in Table 1.

Table 1 Time complexity of image total shuffling algorithm

Size of the image	The average number of iteration needed to accomplish a row transformation
32×32	80
64×64	300
128×128	520
256×256	1600

2.3 Lorenz and Chen's chaotic systems

In the proposed encryption scheme, the Lorenz chaotic system is one that is employed in key scheming, which is modeled by Beldhouche and Qidwai [19].

$$\begin{cases} \dot{x}_1 = p(x_2 - x_1) \\ \dot{x}_2 = -x_1 x_3 + r x_1 - x_2 \\ \dot{x}_3 = x_1 x_2 - t x_3 \end{cases} \tag{5}$$

where p, r, and t are parameters, when $p = 10$, $r = 28$, $t = 8/3$, the system is chaotic.

Another chaotic system in our scheme is Chen's chaotic system, which is described as follows [19]:

$$\begin{cases} \dot{x}_4 = a(x_5 - x_4) \\ \dot{x}_5 = (c - a)x_4 - x_4 x_6 + c x_5 \\ \dot{x}_6 = x_4 x_5 - b x_6 \end{cases} \tag{6}$$

where a, b, and c are parameters, when $a = 35$, $b = 3$, $c = 28$, the system is chaotic. Simulation shows the system orbit is extremely sensitive to the parameter c.

Although the equation of Chen's system are very similar to that of Lorenz system, the topologically they are not equivalent [18]. The chaotic behaviors of the two systems are shown in Fig. 2.

2.4 New image encryption algorithm

The state variable combination of two chaotic systems has been used in this image encryption scheme. Three of the variables are combined differently, which may produce 20 different combination tables, which is given in Table 2. Then the encryption process is given as follows:

Step 1: Assume the dimension of the original gray-scale image is $M \times N$, this image is considered as a matrix named B.

Step 2: First, the plain-image is divided into 4 sub-images and then the position of each sub-image is changed according to the array "RandomA" which is initialized pseudo-randomly.

Step 3: Next, the original image is shuffled according to the total shuffling matrix.

Step 4: Now, repeat the operations of steps 5, 6, and 7 until the whole encrypted image is produced from the parallel encryption of sub-images.

Step 5: Iterate the Lorenz and Chen's chaotic systems for N_0 and M_0 times to avoid the harmful effect of transitional procedure, respectively ($N_0 \neq M_0$).

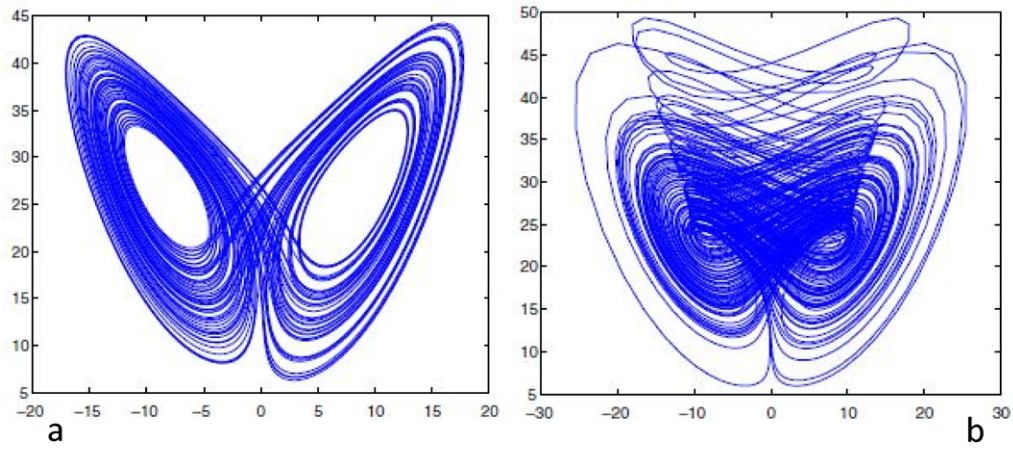

Fig. 2 (**a**) Chaos attractors of Lorenz system, (**b**) chaos attractors of Chen's system

Table 2 Different combinations of states between Lorenz and Chen's chaotic systems

Serial number	Combination of states	Serial number	Combination of states
0	(x_1, x_2, x_3, x_4)	1	(x_1, x_2, x_3, x_5)
2	(x_1, x_2, x_3, x_6)	3	(x_1, x_2, x_4, x_5)
4	(x_1, x_2, x_4, x_6)	5	(x_1, x_2, x_5, x_6)
6	(x_1, x_3, x_4, x_5)	7	(x_2, x_3, x_4, x_6)
8	(x_1, x_3, x_5, x_6)	9	(x_1, x_4, x_5, x_6)
10	(x_2, x_3, x_4, x_5)	11	(x_2, x_3, x_4, x_6)
12	(x_2, x_3, x_5, x_6)	13	(x_2, x_4, x_5, x_6)
14	(x_3, x_4, x_5, x_6)		

Step 6: The Lorenz and Chen's systems are iterated simultaneously, and as a result, six decimal fractions $x_1, x_2, x_3, x_4, x_5, x_6$ will be generated. These decimal values are determined firstly as follows:

$$x_i = \mathrm{mod}\big(\big(\mathrm{abs}(x_i)\big) - \mathrm{floor}\big(\mathrm{abs}(x_i)\big) \times 10^{15}, 256\big) \quad i = 1, 2, \ldots, 6 \tag{7}$$

Where $\mathrm{abs}(x)$ returns the absolute value of x and $\mathrm{floor}(x)$ returns the value of x to the nearest integers less than or equal to x, to make the states of the two chaotic systems correlative, let

$$x_4 = x_4 \oplus x_1, \quad x_5 = x_5 \oplus x_2, \quad x_6 = x_6 \oplus x_3 \tag{8}$$

Step 7: Generates $\bar{x}_1$ by using the following formula:

$$\bar{x}_1 = \mathrm{mod}(x_1, 15) \tag{9}$$

As $\bar{x}_1 \in [0, 14]$. So from Table 2, we select the corresponding group that is used to perform encryption operation if $\bar{x}_1$ equals to the serial number of sequence of the group. The encryption operation is to do XOR between 4 bytes of original image data and the 4 bytes of resulting group data, according to the following formula:

$$\begin{aligned}
&C_{i,j}^{4xk+1} = \big(B_{i,j}^{4xk+1} \oplus B_{x_1}\big) \oplus C_{i,j}^{4xk} \\
&\quad i = 1, 2, \ldots, \frac{M}{2}; j = 1, 2, \ldots, \frac{N}{2} \\
&C_{i,j}^{4xk+2} = \big(B_{i,j}^{4xk+2} \oplus B_{x_2}\big) \oplus C_{i,j}^{4xk+2} \\
&\quad i = 1, 2, \ldots, \frac{M}{2}; j = \frac{N}{2} + 1, \ldots, N \\
&C_{i,j}^{4xk+3} = \big(B_{i,j}^{4xk+3} \oplus B_{x_1}\big) \oplus C_{i,j}^{4xk+3} \\
&\quad i = \frac{M}{2} + 1, \ldots, M; j = \frac{N}{2} + 1, 2, \ldots, \frac{N}{2}
\end{aligned} \tag{10}$$

$$C_{i,j}^{4xk+4} = \left(B_{i,j}^{4xk+4} \oplus B_{x_1}\right) \oplus C_{i,j}^{4xk+4}$$

$$i = \frac{M}{2} + 1, \ldots, M; j = \frac{N}{2} + 1, 2, \ldots, \frac{N}{2}$$

where $k = 0, 1, \ldots$ represents the $(k-1)$th iteration of the two chaotic systems. The symbol $\oplus$ represents the exclusive OR operation bit-by-bit. B_{x_i}, $i = 1, 2, 3, 4$ represents state values of the corresponding group with respect to serial $\bar{x}_1$. The initial $C_{i,j}^0$ is set to be 128, the process does not end until the whole original image is all encrypted. Note that each of the four above equations is used for encrypting one of the plain sub-images.

A general view of the encryption procedure can be obtained from Fig. 3.

3 Experimental analysis

Hackers can easily penetrate to the encrypted images by analyzing them from different statistical aspects.

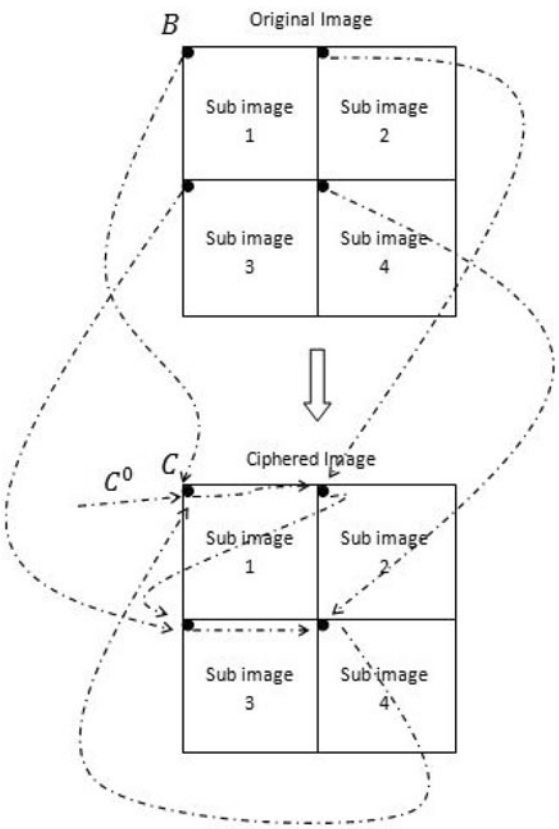

Fig. 3 General view of the proposed parallel encryption algorithm

Their goal is to find the most similar features between the plain-image and the cipher one in order to reach the original image quickly. Hence, a good encryption algorithm should produce cipher-images such that they have salient differences with their corresponding plain-images from statistical points of view.

Experimental analysis of the proposed image encryption algorithm in this paper has been done. The plain image with the size 512×512 is shown in Fig. 4a and the histogram of the plain image is shown in Fig. 4b. The image we get through change of the 512×512 image total shuffling matrix is shown in Fig. 4c and the corresponding histogram is shown in Fig. 4d. The encrypted image is shown in Fig. 4e and the histogram is shown in Fig. 4f. From the figure, one can see that the histogram of the ciphered image is fairly uniform and is significantly different from that of the original image.

4 Security analysis

A good encryption algorithm should resist all kinds of known attacks, it should be sensitive to the secret keys, and the key space should be large enough to make brute-force attacks infeasible. Some statistical analysis has been gathered in this section using plain-image and cipher-image histograms. Then a method is introduced for estimating the similarity between the pixels of the plain image and the encrypted one. Furthermore, a number is considered as the correlation coefficient for each image to show the amount of correlation among different pixels of image.

4.1 Key space analysis

In our algorithm, the initial value of x_0 that is used for image division is set to 0.556 which needs a 32-bits space and also the initial values of the Lorenz and Chen's systems, $p = 10$, $r = 28$, $t = 8/3 \cong 2.66$, $a = 35$, $b = 3$, and $c = 28$ are used as secret keys. So, we need 96 bits to store these initial values. Another 104 bits is needed for storing: $x_1(0) = 0.3$, $x_2(0) = -0.4$, $x_3(0) = 1.2$, $x_4(0) = 10.2$, $x_5(0) = -3.5$, $x_6(0) = 4.4$. Moreover, the initial iteration numbers N_0 and M_0 are also used as the secret keys. With $N_0 = 3000$ and $M_0 = 2000$, we need 64 extra bits to store them.

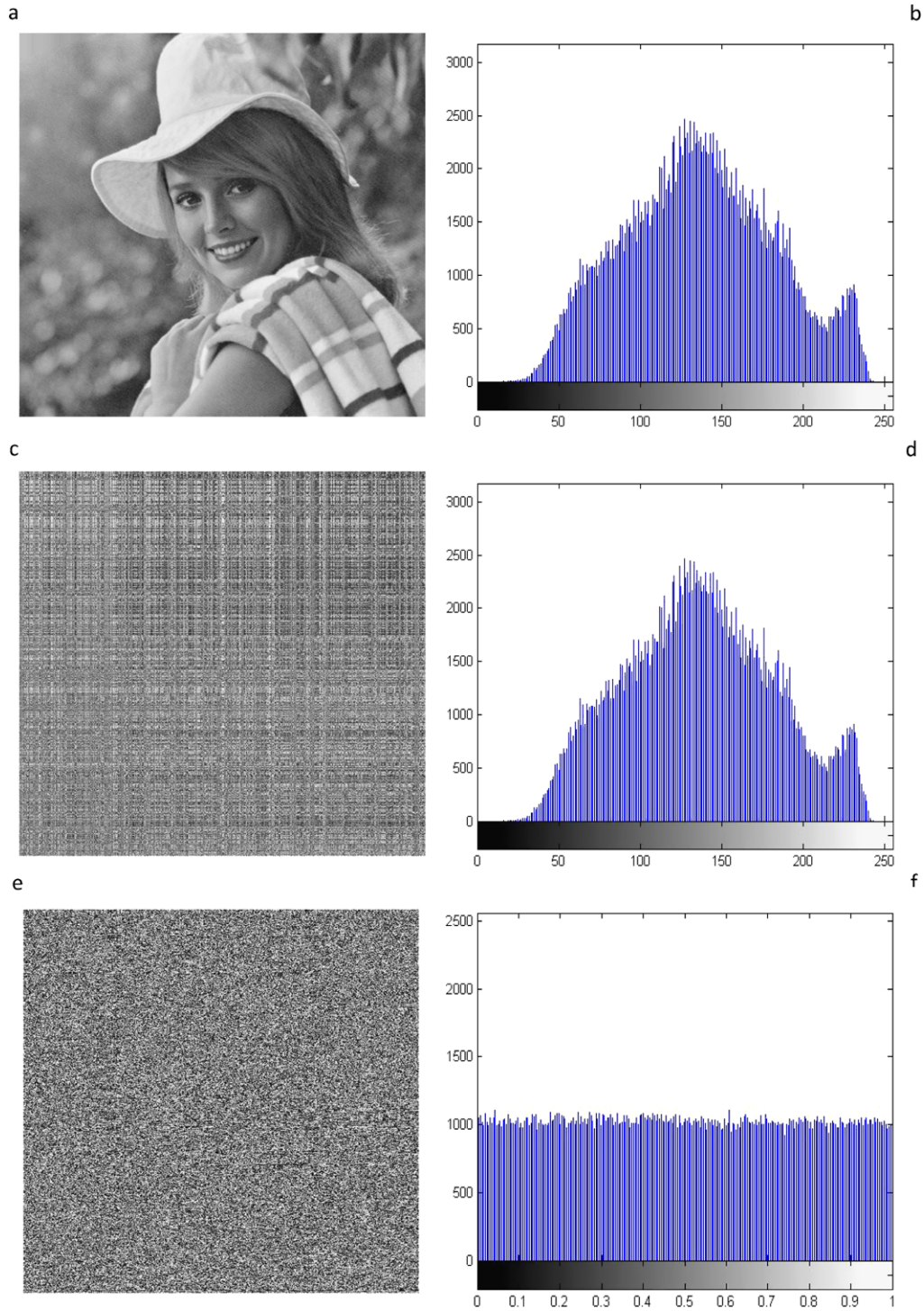

Fig. 4 (**a**) Original image, (**b**) histogram of the original image, (**c**) shuffled image, (**d**) histogram of the shuffled image, (**e**) ciphered image, (**f**) histogram of the ciphered image

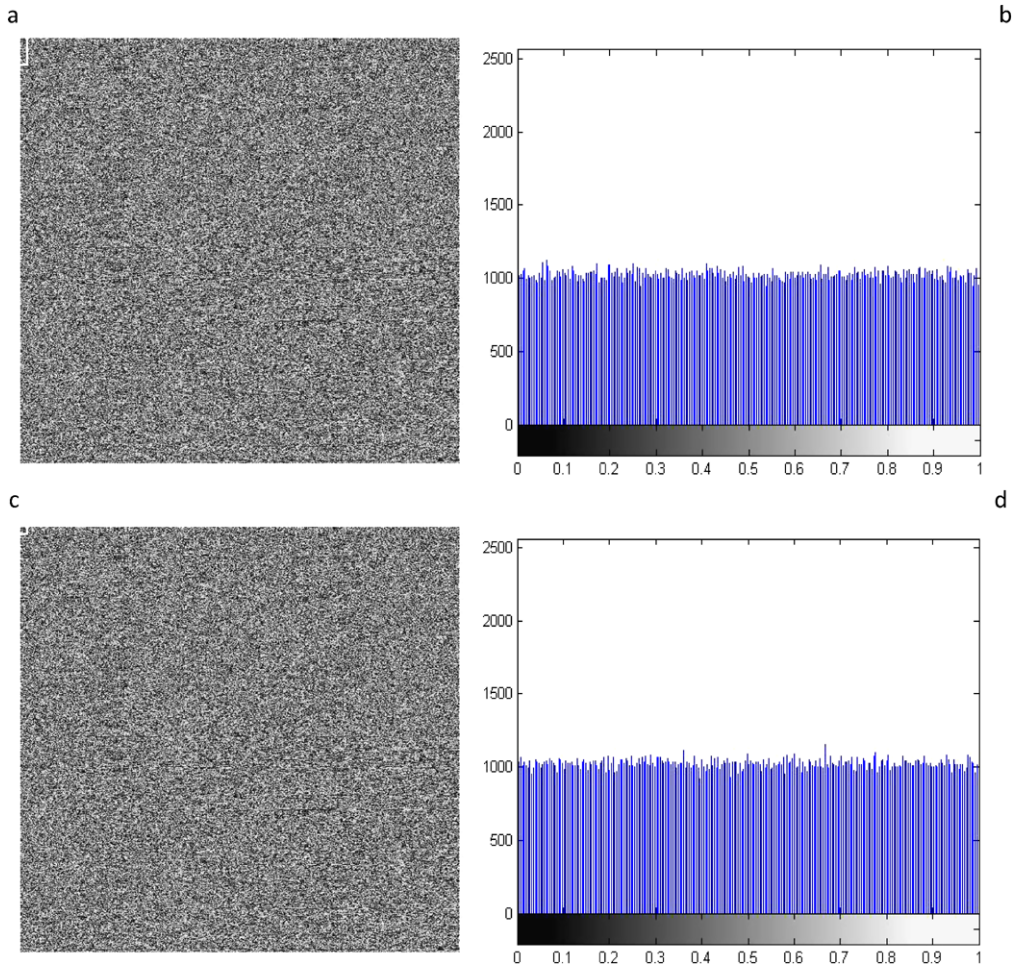

Fig. 5 (**a**) Encrypted image with actual parameters, (**b**) histogram of encrypted image with actual encryption parameters, (**c**) encrypted image with different initial value, (**d**) histogram of encrypted image with different initial value

Therefore, the total number of bits needed for storing all of the encryption parameters is 296, and thus the cryptosystem has 2^{296} different combinations and this large key space is enough to resist all kinds of brute-force attacks.

4.2 Key sensitivity test

Several key sensitivity tests are performed in this paper. Figure 5a–d illustrates the sensitivity of our scheme to the secret key x_1, x_2, x_3, x_4, x_5, x_6, N_0, and M_0. Figure 5a is the encrypted image with the parameters to be $x_1(0) = 0.3$, $x_2(0) = -0.4$, $x_3(0) = 1.2$, $x_4(0) = 10.2$, $x_5(0) = -3.5$, $x_6(0) = 4.4$, $N_0(3000)$, and $M_0 = 2000$ and Fig. 5c is the same encryption result with all the parameters equal to actual ones except $x_1(0) = 0.3000001$. Figures 5b and 5d are the corresponding histograms of encrypted images. So, it can be concluded that the new chaotic encryption algorithm is sensitive to the key such that a small change of the key will generate a completely different decryption result and cannot get the correct plain image.

4.3 Similarity of the adjacent pixels

In the case of similarity of pixels, it should be mentioned that the closer they are to the main diagonal of the image matrix the more similar they are. In cryptography, we are seeking for methods that reduce this similarity to its minimum value in the ciphered image. Then there would be a very little

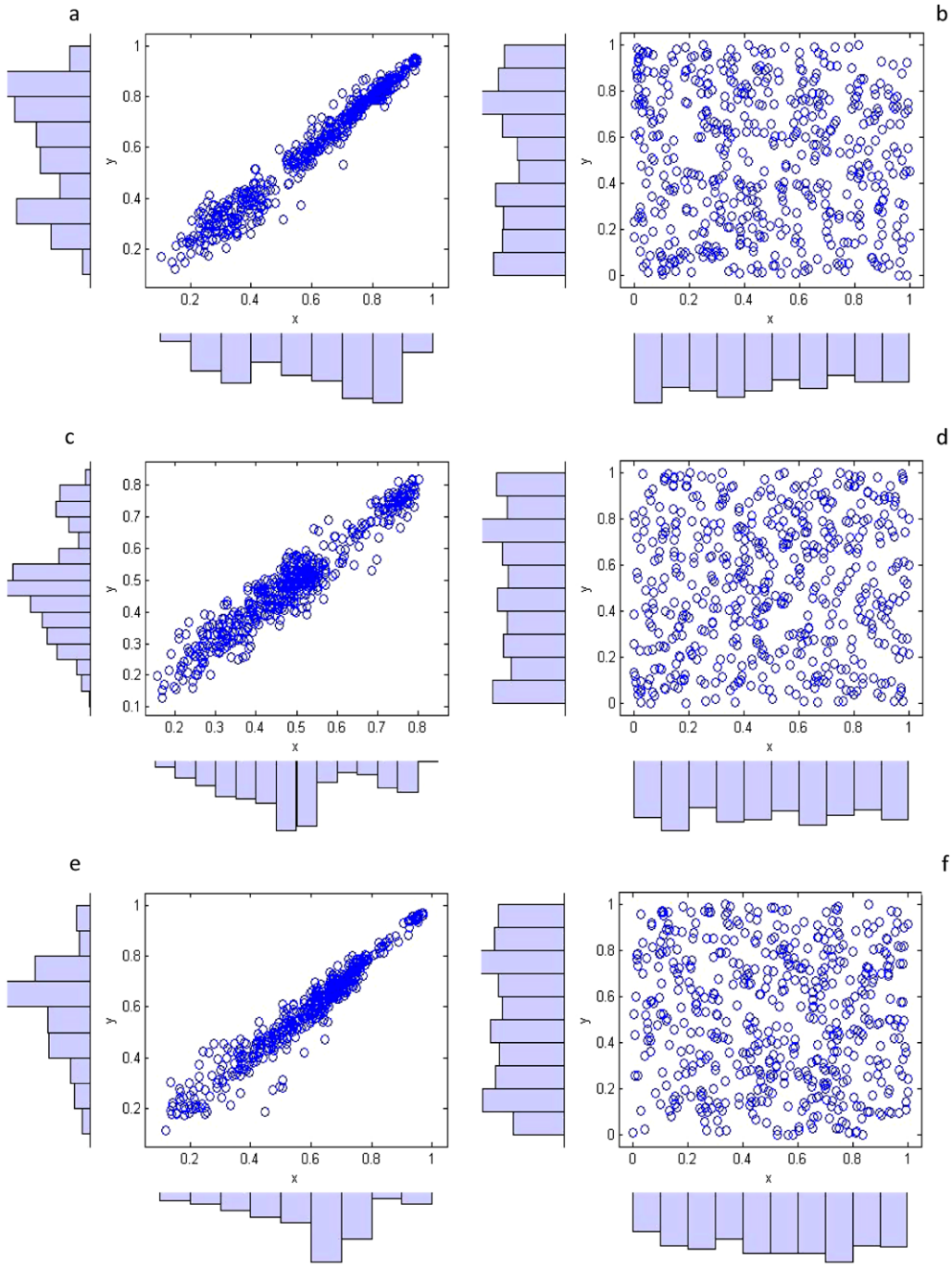

Fig. 6 (**a**) Similarity of two horizontal adjacent pixels in plain image. (**b**) Similarity of two horizontal adjacent pixels in ciphered image. (**c**) Similarity of two vertical adjacent pixels in plain image. (**d**) Similarity of two vertical adjacent pixels in ciphered image. (**e**) Similarity of two diagonal adjacent pixels in plain image. (**f**) Similarity of two diagonal adjacent pixels in ciphered image

chance for others to reach the original image by comparing these similarities between pixels. The similarity test for horizontal, vertical, and diagonal adjacent pixels has been performed for the proposed encryption algorithm and the results are gathered in Fig. 6.

4.4 Analysis of correlation of two adjacent pixels

To test the correlation between two vertically adjacent pixels, two horizontally adjacent pixels, and two diagonally adjacent ones, respectively, in a ciphered image, some simulations are carried out. Firstly, randomly select 2,500 pairs of two adjacent pixels from the image, then calculate the correlation coefficient of each pair by using the following formulas:

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{11}$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}\big(x_i - E(x_i)\big)^2 \tag{12}$$

$$\mathrm{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N}\big(x_i - E(x_i)\big)\big(y_i - E(y_i)\big) \tag{13}$$

$$r_{xy} = \frac{\mathrm{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{14}$$

where x and y are gray values of two adjacent pixels in the image. Table 3 shows the simulation results for the "elaine.512" image chosen from the USC data base.

The closer this value is to zero the less correlation exists between two adjacent pixels. The results show that the correlation coefficient is very close to zero in ciphered image, and thus the proposed encryption algorithm is less predictable and more secure.

4.5 Analysis of information entropy

Entropy is one of the most outstanding features that make the images to have a random-like behavior. This parameter was first introduced by Claude E. Shannon in 1949 and can be obtained using the following formula:

$$H(s) = \sum_{i=0}^{2N-1} P(s_i)\log\left(\frac{1}{p(s_i)}\right) \tag{15}$$

where N is the number of gray scale levels in an image (ex: $N = 256$ for 8 bit image pixels) and $P(s_i)$ is the occurrence probability of gray scale "I" in the image.

The entropy value will be 8 for images that are produced totally randomly. The closer the entropy of an encryption algorithm is to 8 the less predictable, and thus more secure is that scheme. The entropy value for the proposed encryption algorithm has been measured for a sample image and the result is shown in Table 4.

Table 3 Correlation coefficients of two adjacent pixels in two images

Model	Original image	Ciphered image
Horizontal	0.9784	−0.0893
Vertical	0.9758	0.0034
Diagonal	0.9750	0.0010

Table 4 The entropy of original gray scale image and its corresponding encrypted one

Entropy of plain-image	Entropy of cipher-image
7.5060	7.9993

Table 5 The comparison of encryption speed between our proposed method and the cryptosystem in [16]

Algorithm	Encryption time (seconds)
The proposed algorithm in [16]	63.450736
Our proposed algorithm	55.855276

4.6 Encryption speed

We have used Matlab 7.8 to run encryption programs in a computer with a Pentium 4 CPU 2.66 GHz, 4 GB Memory, and 300 GB hard-disk capacity, and the operating system is Microsoft Windows Vista Business. Table 5 shows the comparison of experiment results between the proposed parallel encryption method and the cryptosystem in [16], which our encryption method is based on. Compared to this encryption system, we can see that the operation speed of our method is about 8 seconds faster for the "elaine.512" image.

5 Conclusions and future works

In this paper, we introduced a new parallel algorithm for image encryption. First of all, the plain image is divided to 4 equal blocks and then the position of each block is shuffled. Then a total shuffling algorithm is applied to the whole image. After this, we use different values for encrypting each pixel in each of the 4 blocks of the whole image. These values are: the corresponding pixel value of the original image, the value which is obtained from the other block and the values extracted from Table 2.

One of the most outstanding benefits of the proposed encryption algorithm is that we increased the diffusivity by correlating the encryption procedure of each pixel in each block to its corresponding pixel (the same row and column) in the other block. Moreover, the experimental results show that the algorithm possesses high security and a large key space. The proposed encryption algorithm is also fast; there are only some XOR operations and table lookup operations for each pixel.

The future extensions of this work can be as follows:

(1) This encryption algorithm can be implemented by dividing the plain image into more adaptive sub-images.
(2) We can use other chaotic systems and concepts (e.g., cycling chaos) except Lorenz and Chen for obtaining different results.

References

1. Beldhouche, F., Qidwai, U.: Binary image encoding using ID chaotic map. In: Proceedings of the IEEE Annual Technical Conference, pp. 39–43 (2003)
2. Bu, S.L., Wang, B.H.: Improving the security of chaotic encryption by using a simple modulating method. Chaos Solitons Fractals **19**, 919–924 (2004)
3. Chang, C.C., Hwang, M.S., Chen, T.S.: A new encryption algorithm for image cryptosystems. J. Syst. Softw. **58**, 83–91 (2001)
4. Chee, C.Y., Xu, D., Steven, R., Bishop, B.: A zero-crossing approach to uncover the mask by chaotic encryption with periodic modulation. Chaos Solitons Fractals **21**, 1129–1134 (2004)
5. Chen, G., Mao, Y.B., Chui, C.K.: A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Solitons Fractals **21**, 749–761 (2004)
6. Chien, T.-I., Liao, T.-L.: Design of secure digital communication systems using chaotic modulation, cryptography and chaotic synchronization. Chaos Solitons Fractals **24**, 241–255 (2005)
7. Kocarev, L.: Chaos-based cryptography: a brief overview. IEEE Circuits Syst. Mag. **1**(3), 6–21 (2001)
8. Kocarev, L., Jakimovski, G.: Chaos and cryptography: from chaotic maps to encryption algorithms. IEEE Trans. Circuits Syst. **48**(2), 163–169 (2001)
9. Fridrich, J.: Symmetric ciphers based on two-dimensional chaotic maps. Int. J. Bifurc. Chaos Appl. Sci. Eng. **8**(6), 1259–1284 (1998)
10. Gao, H.J., Zhang, Y.S., Liang, S.Y., Li, D.Q.: A new chaotic algorithm for image encryption. Chaos Solitons Fractals **29**, 393–399 (2006)
11. Li, S., Zheng, X.: Cryptanalysis of a chaotic image encryption method. In: Proceedings of the IEEE International Conference on Circuits and Systems, vol. 2, pp. 708–711 (2002)
12. Liu, H., Wang, X.: Color image encryption based on one-time keys and robust chaotic maps. Comput. Math. Appl. **59**(10), 3320–3327 (2010)
13. Lü, J.H., Chen, G.R.: A new chaotic attractor coined. Int. J. Bifurc. Chaos Appl. Sci. Eng. **12**(3), 659–661 (2002)
14. Mao, Y.B., Chen, G., Lian, S.G.: A novel fast image encryption scheme based on the 3D chaotic baker map. Int. J. Bifurc. Chaos Appl. Sci. Eng. **14**, 3613–3624 (2004)
15. Matthews, R.: One the derivation of a chaotic encryption algorithm. Cryptologia **8**(1), 29–42 (1989)
16. Zhou, Q., Wong, K.-w., Liao, X., Xiang, T., Hu, Y.: Parallel image encryption algorithm based on discretized chaotic map. Chaos Solitons Fractals **00**, 1081–1092 (2007)
17. Schneier, B.: Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd edn. Wiley, New York (1995)
18. Gao, T., Chen, Z.: Image encryption based on a new total shuffling algorithm. Chaos Solitons Fractals **00**, 213–220 (2006)
19. Wang, X.-Y., Feng, C., Tian, W.: A new compound mode of confusion and diffusion for block encryption of image based on chaos. Commun. Nonlinear Sci. Numer. Simul. **15**(9), 2479–2485 (2009)