Available online at www.sciencedirect.com
ScienceDirect
Physics Letters A 372 (2008) 394–400
PHYSICS LETTERS A
www.elsevier.com/locate/pla

A new image encryption algorithm based on hyper-chaos

Tiegang Gao [a,*], Zengqiang Chen [b]

[a] College of Software, Nankai University, Tianjin 300070, PR China
[b] Department of Automation, Nankai University, Tianjin 300070, PR China

Received 24 October 2006; received in revised form 17 July 2007; accepted 20 July 2007

Available online 26 July 2007

Communicated by A.P. Fordy

Abstract

This Letter presents a new image encryption scheme, which employs an image total shuffling matrix to shuffle the positions of image pixels and then uses a hyper-chaotic system to confuse the relationship between the plain-image and the cipher-image. The experimental results demonstrate that the suggested encryption algorithm of image has the advantages of large key space and high security, and moreover, the distribution of grey values of the encrypted y image has a random-like behavior.

Keywords: Hyper-chaos; Image encryption; Image total shuffling matrix; Key space

1. Introduction

With the rapid developments in digital image processing and network communication, electronic publishing and widespread dissemination of digital multimedia data over the Internet, protection of digital information against illegal copying and distribution has become extremely important. To meet this challenge, many new encryption schemes have been proposed [1–4]. Among them, chaos-based algorithms has suggested a new and efficient way to deal with the intractable problem of fast and highly secure image encryption, and it has been proved that in many aspects chaotic maps have analogous but different characteristics as compared with conventional encryption algorithms [5–10].

The chaos-based encryption was first proposed in 1989 [11], since then, many researchers have proposed and analyzed a lot of chaos-based encryption algorithms, these work all have been motivated by the chaotic properties such as the sensitive dependence on initial conditions and system parameters, pseudo-random property, non-periodicity and topological transitivity, etc. While classical encryption algorithms are sensitive to keys, so some elaborated constructions are need to achieve satisfying and safer chaos-based encryption.

It is well known that a good encryption algorithm should be sensitive to the cipher keys, and the key space should be large enough to make brute-force attacks infeasible [12]. Recently, in [13], a fast chaotic cryptographic scheme based on iterating a Logistic map was proposed, and no random numbers need to be generated and the look-up table used in the cryptographic process is updated dynamically. In [14], a two-dimensional chaotic cat map is generalized to 3D for designing a real-time secure symmetric encryption scheme, which employs the 3D cat map to confuse the relationship between the cipher-image and the plain-image. Also recently, authors in [15] thought that the algorithm for encoding binary images using one-dimensional chaotic map [16] is not secure enough, and there is the same problem with the algorithm proposed in [17], to overcome the drawbacks such as small key space and weak security of one-dimensional chaotic map, a nonlinear chaos algorithm is proposed in [18], which shows high-level security and acceptable efficiency.

Recently, because hyper-chaos has more than one positive Lyapunov exponent, and have more complex dynamical characteristics than chaos, so secure communication schemes based on hyper-chaotic systems have been investigated [19,20], but at present, there is little work about the study of encryption algo-

* Corresponding author.
E-mail address: gaotiegang@nankai.edu.cn (T. Gao).

0375-9601/$ – see front matter
doi:10.1016/j.physleta.2007.07.040

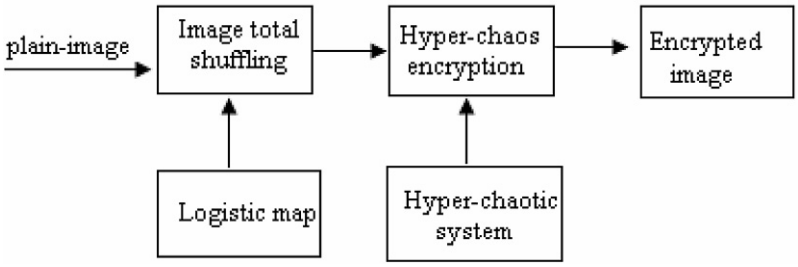


Fig. 1. Block diagram of the image encryption.

rithm based on hyper-chaos. In general, as the prediction time of a chaotic system is longer than that of a hyper-chaotic system [21], so it may be more valuable to study the application of hyper-chaos in encryption algorithms.

A new image encryption scheme is suggested in this Letter. Since digital images are usually represented as two-dimensional arrays, in order to disturb the high correlation among pixels, Arnold cat 2D map or chaotic 3D cap map is often used to shuffle the positions of the pixels in the image [3,14]. Different from the 2D or 3D chaotic map that is used to shuffle the pixel positions of the plain-image, the encryption proposed here consists of two processes, firstly, we shuffle the image based on total image shuffling matrix generated by using Logistic map, then encrypt the shuffled image by using hyper-chaos. The rest of this Letter is organized as follows. Section 2 presents the proposed algorithm. Section 3 describes some simulation outcomes, some security analysis are given in Section 4. Finally, Section 5 concludes the Letter.

2. The proposed encryption algorithm

The complete image encryption process consists of two parts, as shown in Fig. 1.

2.1. Image encryption based on total shuffling matrix

Image data has strong correlations among adjacent pixels. In order to disturb the high correlation among pixels, an image total shuffling matrix is used to shuffle the position of the plain-image. Without loss of generality, we assume that the dimension of the plain-image is $N \times M$, the position matrix of pixels is $P_{i,j}(I)$, $i = 0, 1, \ldots, M-1$; $j = 0, 1, \ldots, N-1$, where $P_{i,j}(I)$ stands for the grey value of the image. The procedure of shuffling image is described as follows

1. For Logistic map $x_{n+1} = 4x_n(1-x_n)$ and a given x_0, after doing some iterations, a new x_0 is derived, then let

$$l = \mathrm{mod}\big(x_0 \times 10^{14}, M\big). \tag{1}$$

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

2. Continue to do the iteration of Logistic map and do (1) until we get M different data which are all between 0 and $M-1$, these data can be reordered 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, thus a new image position matrix $P^h_{i,j}$ is generated based on row transformation.

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

For every row of the new matrix $P^h_{i,j}$, we will shuffle the column position of the image. The process is presented next.

1. Use the present x_0 to do the iteration of Logistic map and then let

$$l = \mathrm{mod}\big(x_0 \times 10^{14}, N\big). \tag{2}$$

It is easily seen, $l \in [0, N-1]$.

2. Continue to do the iteration of Logistic map and do (2) 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 $P_{i,j}$ 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 $P^{hl}_{i,j}$ is generated.

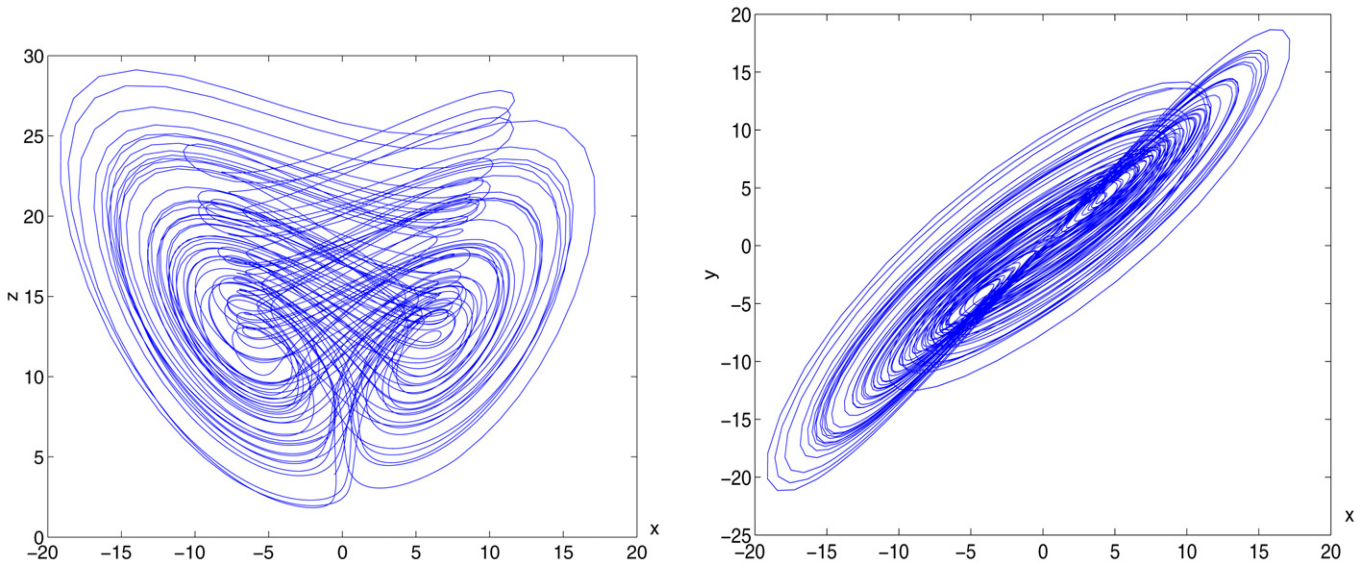
3. For the new $P^{hl}_{i,j}$, go to step 1, 2 to do column transformation for the second row, till the last row transformation is finished, thus a new image total shuffling matrix $P^{hl}_{i,j}$ is presented. For example, the original image and image shuffled can be shown in Fig. 3(a) and (c). If N and M are not very big, the algorithm have lower time complexity, which can be summarized in Table 1.

2.2. A hyper-chaotic system

In the proposed encryption scheme, a new hyper-chaotic system generated from Chen's chaotic system is used in key scheming, which is modeled by [22]

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1), \\ \dot{x}_2 = -x_1x_3 + dx_1 + cx_2 - x_4, \\ \dot{x}_3 = x_1x_2 - bx_3, \\ \dot{x}_4 = x_1 + k \end{cases} \tag{3}$$

where a, b, c, d and k are parameters, when $a = 36$, $b = 3$, $c = 28$, $d = -16$ and $-0.7 \leqslant k \leqslant 0.7$, the system is hyper-chaotic. The hyper-chaos attractors are shown in Fig. 2. with parameters $a = 36$, $b = 3$, $c = 28$, $d = -16$ and $k = 0.2$,



(a) x_1–x_3 plane (b) x_1–x_2 plane

Fig. 2. Hyper-chaos attractors of system (1) with $k = 0.2$.

its Lyapunov exponents are $\lambda_1 = 1.552$, $\lambda_2 = 0.023$, $\lambda_3 = 0$, $\lambda_4 = -12.573$. As the hyper-chaos has two positive Lyapunov exponents, so the prediction time of a hyper-chaotic system is shorter than that of a chaotic system [21], as a result, it is safer than chaos in security algorithm. For more detailed analysis of the complex dynamics of the system, please see relative reference [22].

2.3. Encryption algorithm design

After we get the shuffled image $P_{i,j}^{hl}$ based on total shuffling matrix using Logistic map, the above hyper-chaos is used to encrypt the shuffled image. The encryption scheme is based on the combination of state variables of the above hyper-chaotic system. Three of the four variables are combined differently, which may produce four different combinations, which is given in Table 2.

Then, the encryption process is given as follows

Step 1: Iterate the hyper-chaotic system for N_0 times by Runge–Kutta algorithm to avoid the harmful effect of transient procedure.

Step 2: The hyper-chaotic system is iterated, and as a result, four decimal fractions x_1, x_2, x_3, x_4 will be generated. These decimal values are preprocessed firstly as follows

$$x_i = \mathrm{mod}\big(\big(\mathrm{Abs}(x_i) - \mathrm{Floor}\big(\mathrm{abs}(x_i)\big)\big) \times 10^{14}, 256\big), \quad i = 1, 2, 3, 4, \tag{4}$$

where $\mathrm{Abs}(x)$ returns the absolute value of x. $\mathrm{Floor}(x)$ returns the value of x to the nearest integers less than or equal to x, $\mathrm{mod}(x, y)$ returns the remainder after division.

Step 3: Generate $\bar{x}_1$ by using the following formula:

$$\bar{x}_1 = \mathrm{mod}(x_1, 4). \tag{5}$$

As $\bar{x}_1 \in [0, 3]$, so from Table 2, we can select the corresponding group that are used to perform encryption operation if $\bar{x}_1$ equals to the serial number of sequence of the group. For example, if $\bar{x}_1 = 2$, then (x_1, x_3, x_4) is used to do encryption. The encryption operation is to do XOR between 3 bytes of image data of $P_{i,j}^{hl}$ and the 3 bytes of resulting group data, according to the following formula

Table 2
Different combinations of states of hyper-chaos

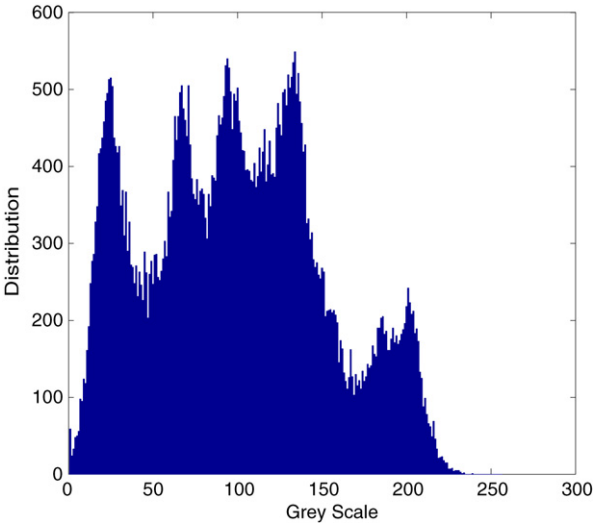
Serial number	Combination of states
0	(x_1, x_2, x_3)
1	(x_1, x_2, x_4)
2	(x_1, x_3, x_4)
3	(x_2, x_3, x_4)

$$\begin{cases} C_{3\times(i-1)+1} = P_{3\times(i-1)+1} \oplus B_{x_1}, \\ C_{3\times(i-1)+2} = P_{3\times(i-1)+2} \oplus B_{x_2}, \\ C_{3\times(i-1)+3} = P_{3\times(i-1)+3} \oplus B_{x_3}, \end{cases} \tag{6}$$

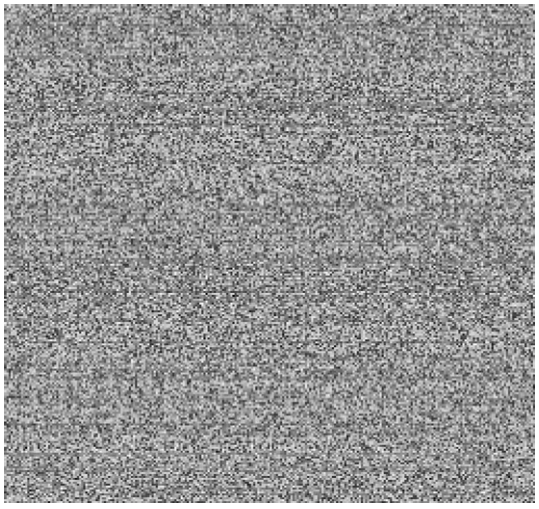
where $i = 1, 2, \ldots$ represents the ith iteration of the hyper-chaotic system. The symbol $\oplus$ represents the exclusive OR operation bit-by-bit. P_i, $i = 1, 2, \ldots, N \times M$, represents pixel values of the shuffled image $P_{i,j}^{hl}$, B_{x_1}, B_{x_2} and B_{x_3} represent state values of the corresponding group with respect to serial $\bar{x}_1$, i.e., they represent the chosen variables of Eq. (3) after being transformed by Eq. (4). The process does not end until the set $P_{i,j}^{hl} = \{P_1, P_2, \ldots, P_{N\times M}\}$ is all encrypted. Then the encrypted pixel set $C = \{C_1, C_2, \ldots, C_{N\times M}\}$ is written to the cipher-image.

The decryption algorithm is similar to the encryption algorithm. That is, for the encrypted image, firstly, decrypt the image using hyper-chaotic system with the same parameters and initial values as that used in encryption, and then anti-shuffle the resulting image, we will get the original image.

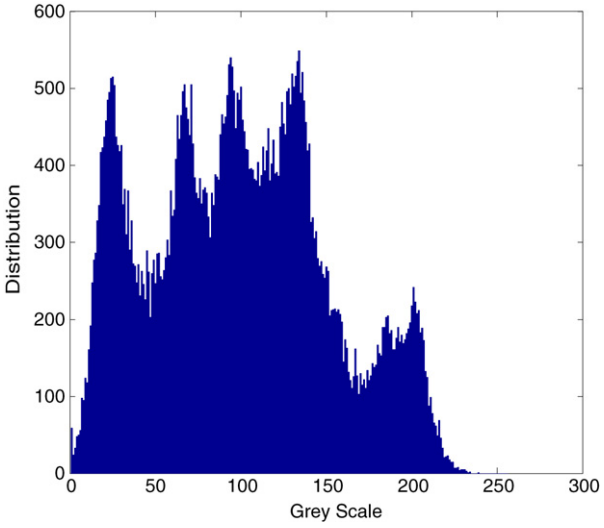
(a) Original image



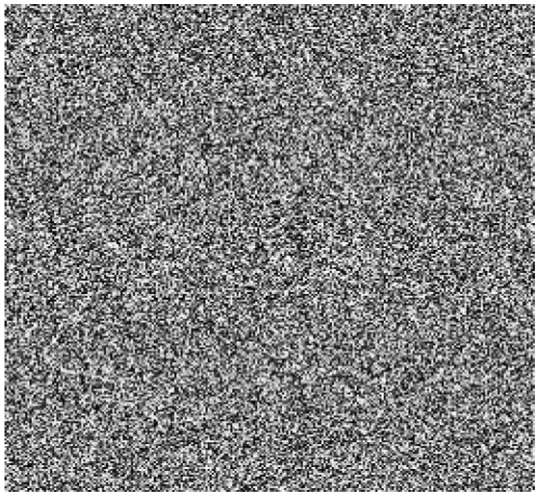
(b) Histogram of the original image



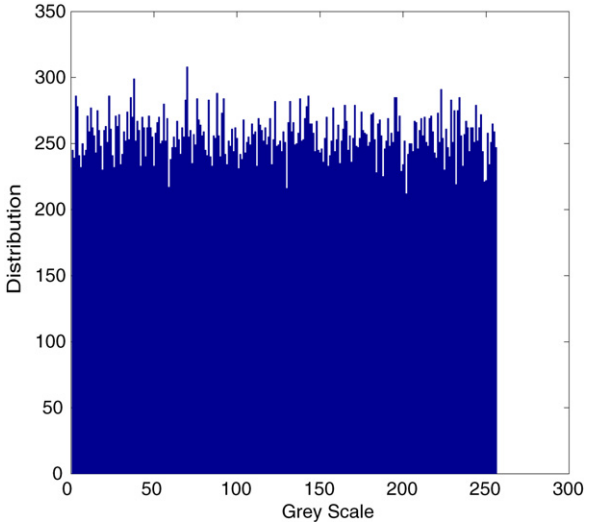
(c) Image encrypted by total shuffling matrix based on Logistic map



(d) Histogram of image (c)



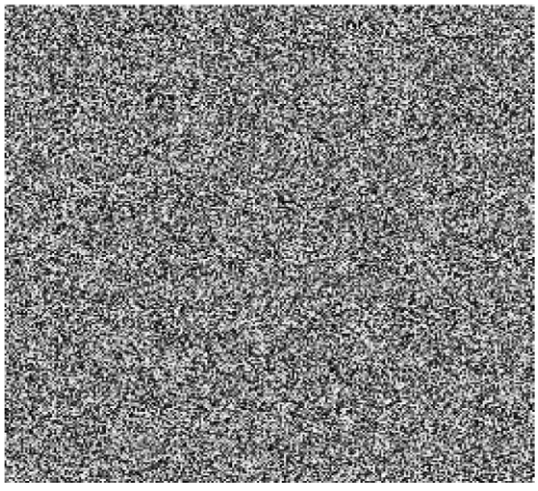
(e) Ciphered image



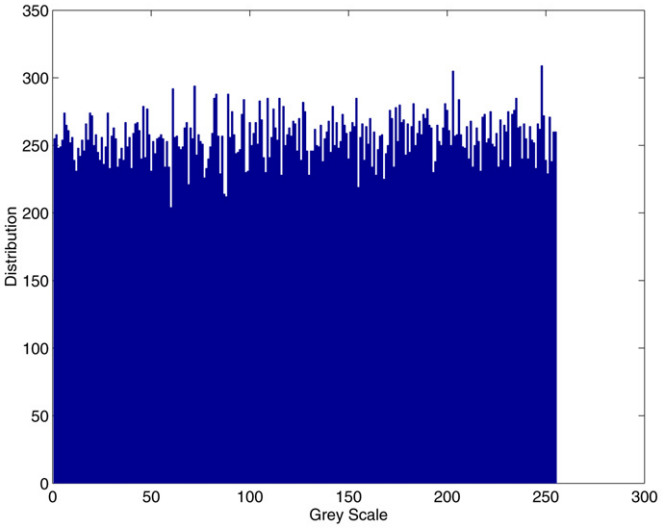
(f) Histogram of the ciphered image

Fig. 3. Image encryption and decryption experimental result.

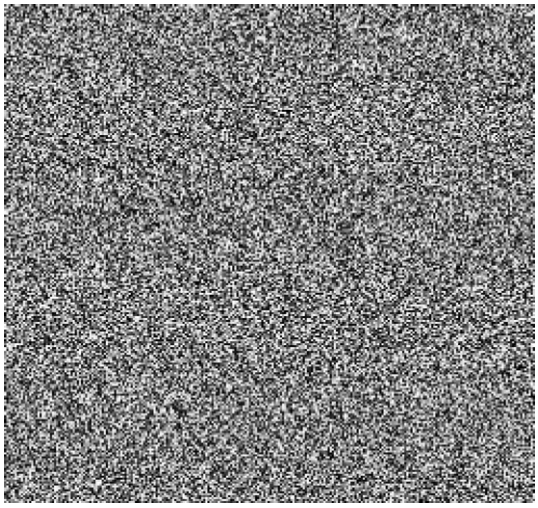
(a) Decrypted image with correct parameters



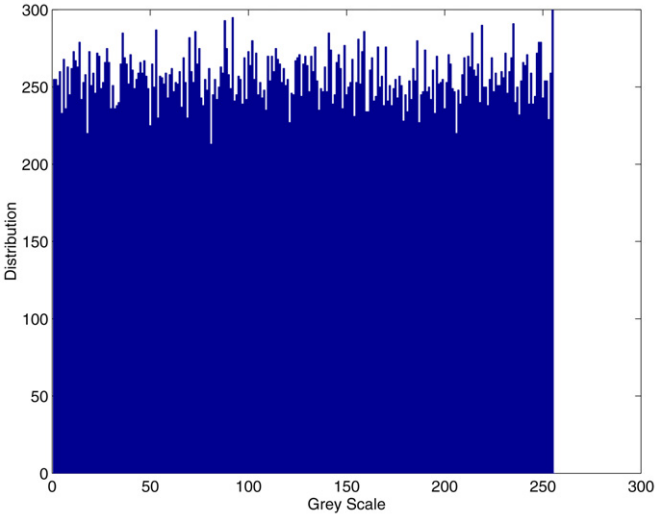
(b) Decrypted image with different initial value



(c) Histogram of decrypted image with different initial value



(d) Decrypted image with different initial iteration value



(e) Histogram of decrypted image with different initial iteration value

Fig. 4. Image encryption and decryption experimental result.

3. Experimental analysis

Experimental analysis of the proposed image encryption algorithm in this Letter has been done. The plain-image with the size 256×256 is shown in Fig. 3(a) and the histogram of the plain-image is shown in Fig. 3(b). Image we get through change of 256×256 magic square matrix is shown in Fig. 3(c) and the corresponding histogram in shown in Fig. 3(d). The encrypted image is shown in Fig. 3(e) and the histogram is shown in Fig. 3(f). 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 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 security analysis has been performed on the proposed image encryption scheme.

4.1. Key space analysis

In our algorithm, the initial values of Logistic map and hyper-chaotic system are used as secret keys, if the precision is 10^{-14}, the key space size is 10^{70}. Moreover, the initial iteration number N_0 and k can also be used as the secret keys. This is enough to resist all kinds of brute-force attacks.

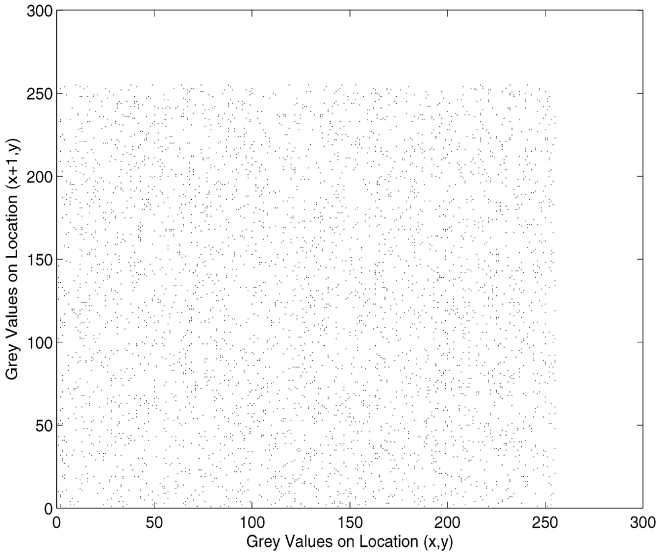
4.2. Key sensitivity test

Several key sensitivity tests are performed. Fig. 4(a)–(d) illustrate the sensitivity of our scheme to the secret key x_1, x_2, x_3, x_4, N_0. Fig. 4(a) is the decrypted image with the same parameters as that used in encryption algorithms, that is $x_1 = 0.3$, $x_2 = -0.4$, $x_3 = 1.2$, $x_4(0) = 1$ and $N_0 = 3000$. Fig. 4(b) is the decrypted image with all the parameters to be same as that used in encryption algorithm except $x_4(0) = 1.000000000001$. Fig. 4(d) is the decrypted image with a different initial iteration times $N_0 = 3001$. Figs. 4(c) and 4(e) are corresponding histograms of the decrypted image, respectively. So it can be concluded that hyper-chaos encryption algorithm is sensitive to the key, a small change of the key will generate a completely different decryption result and cannot get the correct plain-image.

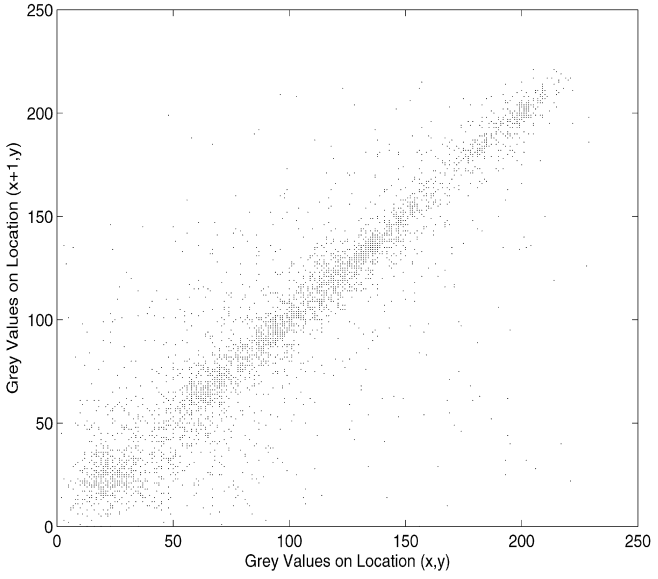
4.3. Analysis of correlation of two adjacent pixels

To test the correlation between two vertically adjacent pixels, two horizontally adjacent pixels, two diagonally adjacent pixels, respectively, in a ciphered image, some simulations are carried out. Firstly, randomly select 4096 pairs of two adjacent pixels from the image, then calculate the correlation coefficient of each pair by using the following formulas [14]:

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



(a) Correlations in the ciphered image



(b) Correlations in the original image

Fig. 5. Correlations of two horizontally adjacent pixels in the original image and in the ciphered image.

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

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

$$r_{xy} = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}}, \tag{7}$$

where x and y are grey values of two adjacent pixels in the image. Fig. 5. shows the correlation distribution of two horizontally adjacent pixels in the original image and that in the ciphered image. The correlation coefficients are 0.9241 and 0.0175, respectively. Other test results are shown in Table 2.

From Table 3, it can be seen that the proposed image encryption algorithm based on image total shuffling matrix have better

Table 3
Correlation coefficients of two adjacent pixels in two images

Model	Original image	Ciphered image
Horizontal	0.9241	−0.0142
Vertical	0.9524	−0.0074
Diagonal	0.9017	−0.0183

performance compared with algorithm proposed by Ref. [18], which used the same grey image as that in this Letter, but get the correlation coefficients of horizontal, vertical and diagonal are −0.01589, −0.06538 and −0.03231, respectively. In the meantime, as hyper-chaos has larger key spaces than chaos used in some literatures such as Refs. [2,3,20], and the image shuffling algorithm proposed here is more securer than that by using Arnold cat map transformation, which is periodic [3], so hyper-chaos may has some potential application in the application of image encryption algorithms.

5. Conclusions

In this Letter, a new image encryption algorithm based on hyper-chaos is proposed, which uses a new image total shuffling matrix to shuffle the pixel positions of the plain-image and then the states combination of hyper-chaos is used to change the grey values of the shuffled-image. Some security analysis are given to demonstrate that the key space of the new algorithm is large enough to make brute-force attacks infeasible, digital simulations have been carried out with detailed numerical analysis, demonstrating the high security of the new image encryption scheme, which may has some potential application in image encryption and information transmission based on Internet.

Acknowledgements

The author would like to thank the reviewers for their constructive suggestions. This work was supported by CNSF Grant #60574036, the Key Program of Natural Science Fund of Tianjin (Grant #07JCZDJC06600), China and the Program for New Century Excellent Talents of China (NCET) to Z.Q. Chen.

References

[1] C.C. Chang, M.S. Hwang, T.S. Chen, J. Syst. Software 58 (2001) 83.
[2] J. Fridrich, Int. J. Bifur. Chaos 8 (1998) 1259.
[3] Z.H. Guan, F.J. Huang, W.J. Guan, Phys. Lett. A 346 (2005) 153.
[4] J. Scharinger, J. Electron Imaging 7 (1998) 318.
[5] L. Kocarev, IEEE Circ. Syst. Mag. 1 (2001) 6.
[6] L. Kocarev, G. Jakimovski, IEEE Trans. Circ. Syst. 1 (48) (2001) 163.
[7] Y.B. Mao, G. Chen, S.G. Lian, Int. J. Bifur. Chaos 14 (2004) 3613.
[8] M.S. Baptista, Phys. Lett. A. 240 (1998) 50.
[9] L.H. Zhang, X.F. Liao, X.B. Wang, Chaos Solitons Fractals 24 (2005) 759.
[10] S.L. Bu, B.H. Wang, Chaos Solitons Fractals 19 (2004) 919.
[11] R. Matthews, Cryptologia 8 (1989) 29.
[12] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, second ed., Wiley, New York, 1995.
[13] K.W. Wong, Phys. Lett. A 298 (2002) 238.
[14] G. Chen, Y.B. Mao, C.K. Chui, Choas Solitons Fractals 21 (2004) 749.
[15] S. Li, X. Zheng, Proc. IEEE Int. Conf. Circuits Syst. 2 (2002) 708.
[16] J.C. Yen, J.I. Go, Proc. IEEE Int. Conf. Circuits Syst. 4 (2000) 49.
[17] F. Beldhouche, U. Qidwai, IEEE Ann. Tech. Conf. (2003) 39.
[18] H.J. Gao, Y.S. Zhang, S.Y. Liang, D.Q. Li, Chaos Solitons Fractals 29 (2006) 393.
[19] V.S. Udaltsov, J.P. Goedgebuer, L. Larger, et al., Opt. Spectrosc. 95 (2003) 114.
[20] J.P. Goedgebuer, L. Larger, H. Port, Phys. Rev. Lett. 80 (1998) 2249.
[21] S. Yanchuk, T. Kapitaniak, Phys. Rev. E 64 (2001) 056235.
[22] T.G. Gao, Z.Q. Chen, Z.Y. Yuan, G. Chen, Int. J. Mod. Phys. C 17 (2006) 471.