Nonlinear Dyn (2013) 71:259–267
DOI 10.1007/s11071-012-0658-x

An image encryption scheme based on time-delay and hyperchaotic system

Guodong Ye · Kwok-Wo Wong

Received: 11 July 2012 / Accepted: 24 October 2012 / Published online: 8 November 2012

Abstract In this paper, a novel image encryption scheme based on time-delay and hyperchaotic system is suggested. The time-delay phenomenon is commonly observed in daily life and is incorporated in the generation of pseudo-random chaotic sequences. To further increase the degree of randomness, the output of the hyperchaotic system is processed before appending to the generated sequence. A novel permutation function for shuffling the position index, together with the double diffusion operations in both forward and reverse directions, is employed to enhance the encryption performance. Experimental results and security analyses show that the proposed scheme has a large key space and can resist known-plaintext and chosen-plaintext attacks. Moreover, the encryption scheme can be easily modified to adopt other hyperchaotic systems under the same structure.

Keywords Time-delay · Hyperchaotic system · Image encryption · Position index permutation

G. Ye (✉)
College of Science, Guangdong Ocean University, Zhanjiang 524088, Guangdong, China
e-mail: guodongye@gmail.com

G. Ye · K.-W. Wong
Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Avenue, Kowloon Tong, Hong Kong

K.-W. Wong
e-mail: itkwwong@cityu.edu.hk

1 Introduction

With a fast development of information technology and internet infrastructure, it is very convenient to transmit and share all kinds of digital contents nowadays. This saves us much time and cost. However, the security issue needs much attention to prevent the unauthorized access of personal information. Many encryption algorithms have been proposed to protect the privacy of digital images transmitted over a public network. Among them, the approach of using chaotic systems for image encryption [1–4] has attracted much research interest in recent years.

Making use of the favorable characteristics such as high sensitivity to initial condition and parameters, ergodicity and pseudo-randomness, chaotic systems are employed for data encryption. In 1989, Matthews [5] suggested a chaos-based encryption algorithm. Traditional ciphers such as DES, AES and IDEA are block ciphers which may not have high performance in dealing with the large amount of data in an image. Therefore, researchers tried to design image cryptosystems based on chaos [6–10]. Experimental results show that this kind of image encryption scheme can effectively shuffle and diffuse the two-dimensional image pixels.

An effective encryption algorithm must possess a sufficiently large key space to prevent the brute-force attack. Low-dimensional systems fail to meet this requirement due to the small number of control parameters. On the contrary, hyperchaotic systems are usually high-dimensional systems which exhibit richer chaotic

properties. In [11], a total shuffling function was employed to permute the image pixels while a hyperchaotic system was adopted to carry out the diffusion function. Zhu [12] proposed to generate a chaotic key stream by modifying the hyperchaotic sequences. Three phase functions possessing the necessary properties of a secure image encryption algorithm including the confusion and diffusion properties were designed by Kanso et al. [13] using 3D chaotic maps. Besides, there are many other image encryption algorithms [14–22] found in the literature such as the S-box-based methods [20–22]. However, most of them did not consider a natural phenomenon, i.e., time delay, which often appears in our daily life. If it is incorporated in image cryptosystems, the encryption process will become more practical and natural.

In this paper, we design a new chaos-based image cryptosystem making use of the time-delay concept. It possesses the classical architecture of the substitution-diffusion type [16]. We adopt a new position index permutation method, after preprocessed the pseudo-random sequence generated by a hyperchaotic system. This approach results in high randomness, uncorrelated adjacent pixels and low complexity [12, 19]. The bidirectional diffusion [8] is employed so that a single bit change in the plain-image could cause a large difference in the whole cipher-image. Image encryption can be applied in many areas such as military, meteorology and medical science. Some are in the form of patents [23, 24].

The rest of this paper is organized as follows. The proposed image encryption scheme is described in Sect. 2, with the details of time delay, position index permutation and diffusion. Partial algorithms are given and analyzed in this section. In Sect. 3, experimental results are presented to show the effectiveness of our scheme. The related security analyses including size of the key space, statistical analysis, and sensitivity analysis are discussed. Finally, a conclusion is drawn in Sect. 4.

2 The proposed image encryption scheme

The hyperchaotic system studied in [11] is adopted in our design. It is governed by the following set of equations with four control parameters a, b, c, d and four initial values x_0, y_0, z_0 and w_0:

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = -xz + dx + cy - w \\ \dot{z} = xy - bz \\ \dot{w} = x + k \end{cases} \tag{1}$$

As stated in [11], the system is hyperchaotic (see Fig. 1) when the parameters are chosen as $a = 36$, $b = 3$, $c = 28, d = -16$ and $k \in [-0.7, 0.7]$. The system equations are solved using the fourth-order Runge–Kutta algorithm with step size $h = 0.001$.

The following preprocessing [12] is performed on the iteration values x_j, y_j, z_j and w_j to make them more random:

$$s_j = s_j \times 10^5 - \text{round}\big(s_j \times 10^5\big), \quad j = 1, 2, 3, \ldots \tag{2}$$

where the function round(s) rounds s to the nearest integer.

2.1 Time-delay function

The time-delay phenomenon can be observed in most natural processes. Here, we employ the classical logistic function (3) as the random number generator in our cryptosystem:

$$\widetilde{x}_{j+1} = \mu \widetilde{x}_j (1 - \widetilde{x}_j), \quad j = 1, 2, 3, \ldots \tag{3}$$

The system is chaotic if $\mu \in (0.3569946, 4]$. We introduce time delay into the chaotic sequence x obtained from (1) by setting

$$\overline{x}_k = x_{k+\tau_k^{(1)}}, \quad k = 1, 2, 3, \ldots \tag{4}$$

Here, $\tau_k^{(1)} = 0, 1, 2, \ldots$ denotes the kth time-delay value which is given by (3). Then we can update the x values using the time-delay concept and obtain $\overline{x}$. Similar operations are performed for y, z, w with time delay $\tau_k^{(2)}$, $\tau_k^{(3)}$, $\tau_k^{(4)}$, to form the new sequences $\overline{y}$, $\overline{z}$, $\overline{w}$. As the time delay will cause extra time cost, we should limit it to an appropriate range. In this paper, we suppose that the time delay falls within t ($t \leq 10$ can meet the requirement) interval steps as given by (5):

$$\tau_i = \text{floor}\big(\text{mod}\big(\widetilde{x}_i \times 10^3, t\big)\big), \quad i = 1, 2, \ldots \tag{5}$$

Here, the function floor(γ) rounds γ to the nearest integer less than or equal to γ while mod(γ, t) returns the remainder after dividing γ by t.

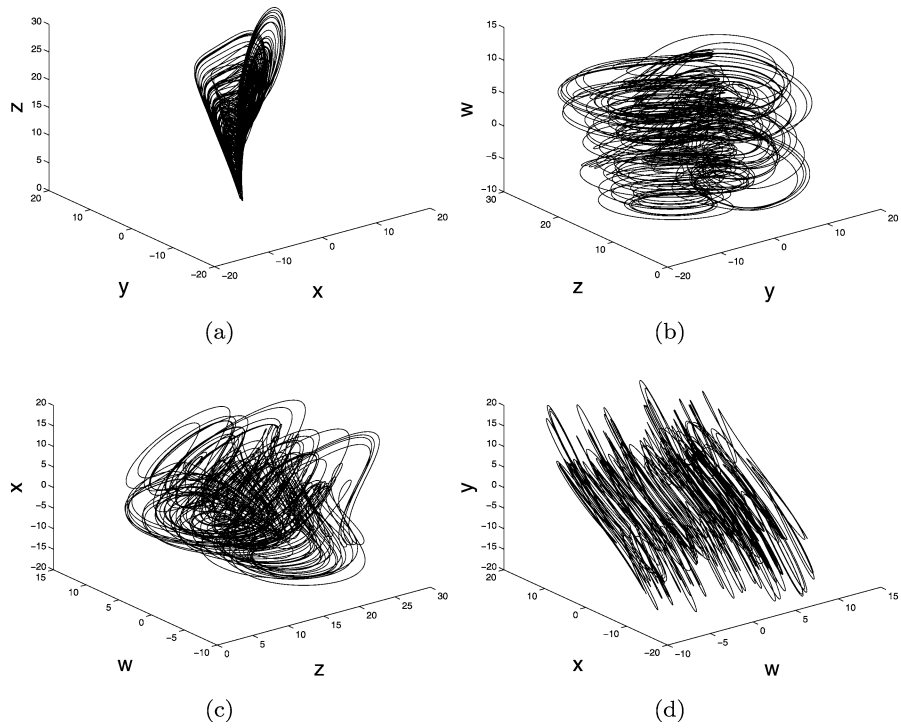

Fig. 1 Hyperchaotic phenomenon on planes (**a**) x–y–z, (**b**) y–z–w, (**c**) z–w–x, (**d**) w–x–y

An ideal pseudo-random sequence must satisfy the essential properties such as uniform distribution, delta function of autocorrelation, and zero cross-correlation. To compare with the method in [12], three autocorrelation plots are shown in Fig. 2. Figure 2(c) shows a better autocorrelation performance than Figs. 2(a) and (b) when time delay is incorporated. Moreover, the cross-correlation plots depicted in Fig. 3 are better than those given by Fig. 2 in reference [12].

In our substitution-diffusion type image cryptosystem, the sequences $\overline{x}$ and $\overline{y}$ generated by system (1) are used for permutation while the sequences $\overline{z}$ and $\overline{w}$ are employed for diffusion.

2.2 Permutation of position index

A permutation of plain-image pixels is usually performed to reduce the high correlation among neighboring pixels. A new method for permuting the position index is designed. Without loss of generality, we assume that the original plain-image A has a size of $m \times n$.

We sort the chaotic sequences $\overline{x}_{1,m\times n}$, $\overline{y}_{1,m\times n}$ and $A_{m\times n}$ using Program Fragment 1 and obtain the position index sequences ind 1 and ind 2.

Program Fragment 1
$[u, \mathrm{ind}\,1] = \mathrm{sort}(\overline{x})$
$[u, \mathrm{ind}\,2] = \mathrm{sort}(\overline{y})$

Along the row direction, we rearrange and permute the 2-D plain-image matrix A into a 1-D sequence $y_{1,m\times n}$ using (6)

$$y(i) = A(f1, f2), \quad i = 1, 2, \ldots, m \times n \tag{6}$$

where $f1 = [\mathrm{ind}\,1/n]$, $f2 = \lceil \mathrm{ind}\,1, n \rceil$, $[u]$ rounds u to the nearest integer towards infinity, and $\lceil u, v \rceil$ returns the remainder of the division. In particular, when $\lceil u, v \rceil$ is equal to 0, it should be replaced by the integer v, as stated in Program Fragment 2.

For the permutation in columns, we can use the same method by replacing ind 1 with ind 2. Then Program Fragment 3 is obtained by a straightforward modification on Program Fragment 2. After the row and column permutations have been performed, a new sequence is obtained, which is then rearranged to a 2-D matrix, i.e., the permuted image.

Program Fragment 2 (for row index permutation)
for $i = 1 : m \times n$

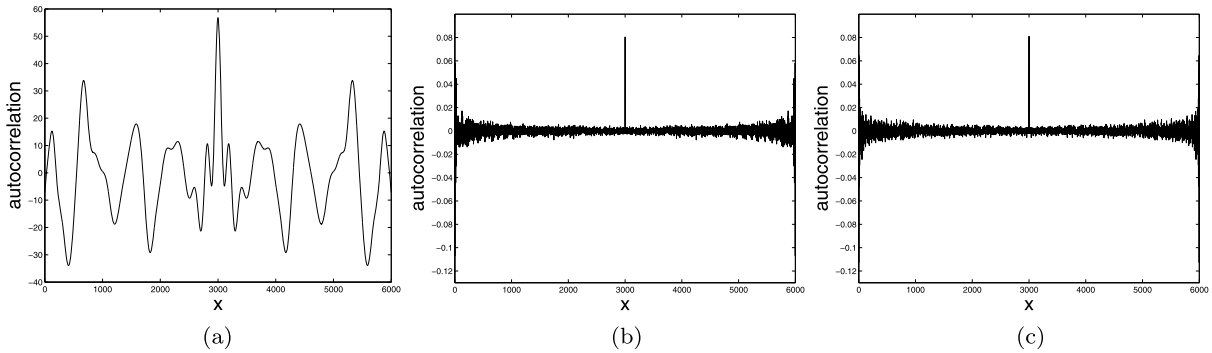

Fig. 2 Autocorrelation comparison (**a**) before preprocessing, (**b**) after preprocessing, (**c**) after preprocessing with delay

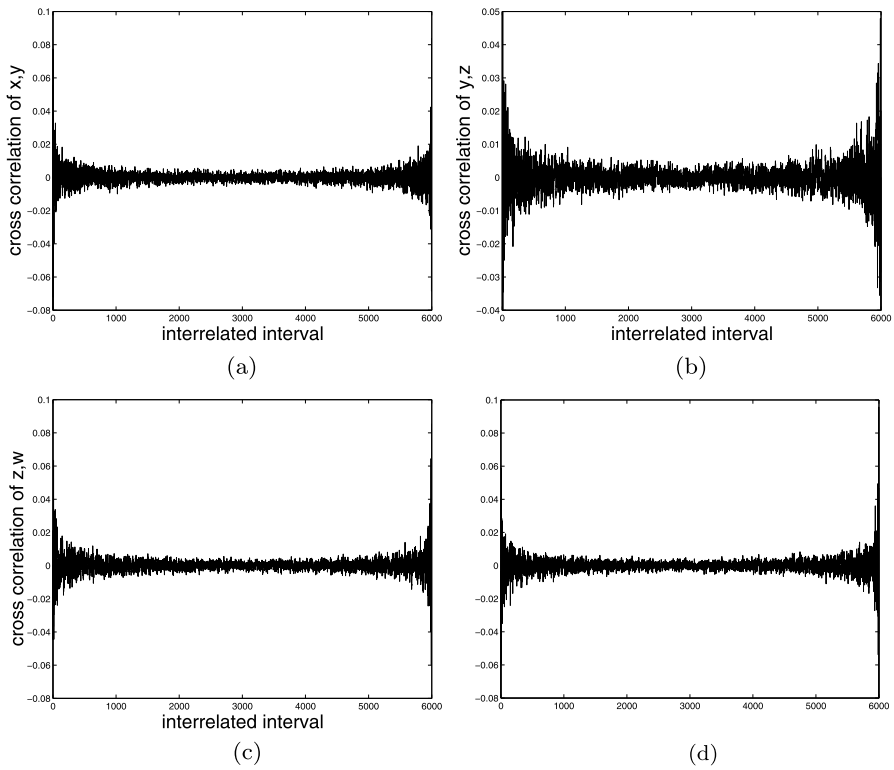

Fig. 3 Cross-correlation plots (**a**) x–y, (**b**) y–z, (**c**) z–w, (**d**) w–x

$f1 = \text{ceil}(\text{ind}1(i)/n);$
$f2 = \text{mod}(\text{ind}1(i), n);$
if $f2 == 0$
$f2 = n;$
end
$y(i) = A(f1, f2);$
end

Program Fragment 3 (for column index permutation)
for $j = 1 : m \times n$
$f1 = \text{ceil}(\text{ind}2(j)/m);$
$f2 = \text{mod}(\text{ind}2(j), m);$
if $f2 == 0$
$f2 = m;$
end
$y(j) = A(f1, f2);$
end

From the original Lena image shown in Fig. 4(a), we obtain the permuted image depicted in Fig. 4(b). A comparison with the Arnold cat map and the tradi-

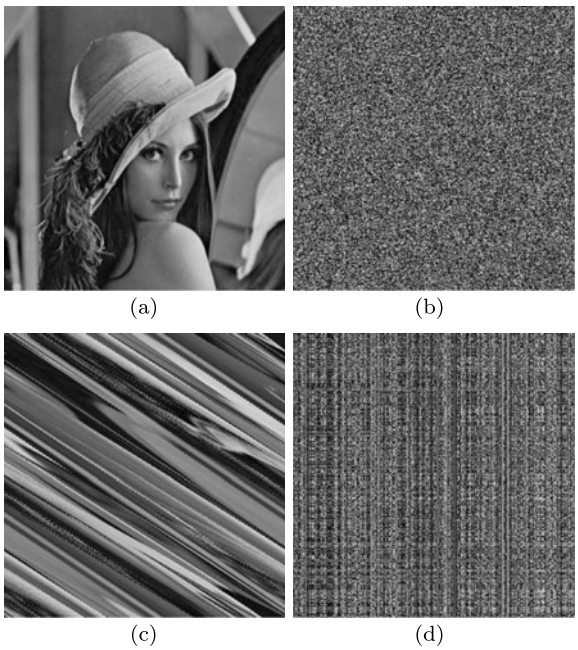

Fig. 4 (**a**) The original Lena image; Permuted image using: (**b**) our method, (**c**) Arnold cat map, (**d**) classical row and column permutation

tional method of row and column permutation shows that our method leads to a stronger permutation effect. However, it is well-known that permutation-only operations cannot resist the chosen-plaintext and known-plaintext attacks. Hence, it is necessary to have the diffusion operation as described in the next subsection.

2.3 Diffusion

Different from most of the existing cryptosystems, the double-direction diffusion [8] is applied on the permuted image. Firstly, forward diffusion from the first pixel to the last pixel is carried out by (8). However, $\overline{z}_i$ and $\overline{w}_i$ should be within $[0, 255]$ before they can be used in (7).

$$\begin{cases} \overline{z} = \mathrm{mod}(\mathrm{floor}(\overline{z} \times 10^{14}), 256) \\ \overline{w} = \mathrm{mod}(\mathrm{floor}(\overline{w} \times 10^{14}), 256) \end{cases} \tag{7}$$

$$c_i = c_{i-1} + p_i + \alpha_1 \overline{z}_{i-1} + \beta_1 \overline{z}_i, \quad i = 1, 2, 3, \ldots, m \times n \tag{8}$$

where c_0 is a selected constant, c_i and c_{i-1} denote the current and the former encrypted pixels, respectively. In order to make the cryptosystem depend on the plain-image, we let c_0 be the average of the first and the last pixels of the permuted image. p_i is the permuted pixel, $\overline{z}_i$ denotes the ith element of the chaotic sequence, with $\overline{z}_0$ equal to the first pixel of the permuted image. α_1, β_1 are two additional control parameters and $+$ refers to the modular addition function.

For a good encryption scheme, a tiny change in the plain-image should influence every pixel in the cipher-image. The diffusion governed by (8) does not stop at the end, but to continue in the opposite direction, as given by

$$e_i = e_{i+1} + c_i + \alpha_2 \overline{w}_{i-1} + \beta_2 \overline{w}_i, \quad i = m \times n - 1, m \times n - 2, \ldots, 2, 1. \tag{9}$$

Here, e_i, c_i are the ith current and former stage of the encrypted pixel, $\overline{w}_i$ denotes the ith element of the chaotic sequence. α_2 and β_2 are two additional control parameters. To keep the natural order, we need to rearrange $\overline{w}_i$ into its inverted sequence before performing the reverse diffusion (9).

When the above-mentioned bidirectional diffusion is finished, we obtain e. The final encrypted image is formed when e is rearranged into a matrix from top to bottom and from left to right.

The decryption procedures are similar to the encryption ones, but in a reversed order. The reverse forms of (9) and (8) are given by (10) and (11), respectively. The inversions of Program Fragments 2 and 3 are obtained by exchanging A and y. With the correct key, it is able to recover the plain-image from the cipher-image.

$$c_i = e_i - e_{i+1} - \alpha_2 \overline{w}_{i-1} - \beta_2 \overline{w}_i, \quad i = m \times n - 1, m \times n - 2, \ldots, 2, 1. \tag{10}$$

$$p_i = c_i - c_{i-1} - \alpha_1 \overline{z}_{i-1} - \beta_1 \overline{z}_i, \quad i = 1, 2, 3, \ldots, m \times n. \tag{11}$$

3 Experimental results and security analyses

All the operations are implemented using Matlab 7.0 on a personal computer equipped with an Intel(R) Core(TM) i3-2350M, 2.30 GHz CPU, running Windows 7. A 500×512 8-bit Barb image shown in Fig. 5(a) is chosen as the test image. Figure 5(b) depicts the corresponding cipher-image after the first round of the proposed encryption method. The initial conditions are $a = 36, b = 3, c = 28, d = -16, k =$

0.42, $x_0 = 1.36$, $y_0 = -3.87$, $z_0 = 10.4$, $w_0 = -8.5$ in system (1). Moreover, $\mu = 4$, $\widetilde{x}_0^{(1)} = 0.543$, $\widetilde{x}_0^{(2)} = 0.232$, $\widetilde{x}_0^{(3)} = 0.104$, $\widetilde{x}_0^{(4)} = 0.705$ in system (3) generate four different time-delay sequences.

3.1 Key space analysis

To make the brute-force attack infeasible, the key space of an image encryption algorithm, including all the initial conditions and control parameters, should be sufficiently large. The key space of our algorithm is constructed by x_0, y_0, z_0, w_0 in system (1), $\widetilde{x}_0^{(1)}$, $\widetilde{x}_0^{(2)}$, $\widetilde{x}_0^{(3)}$, $\widetilde{x}_0^{(4)}$ in system (3). The control parameters such as $\mu, \alpha_1, \alpha_2, \beta_1$ and β_2 are not counted. Thus, the size of key space can reach 10^{112} if the computation precision is 10^{-14}. It is large enough to resist brute-force attack.

3.2 Sensitivity analysis

A good encryption scheme should be sensitive to every secret key. From the perspective of cryptography, a totally different cipher-image should be obtained even for a tiny change in the key. Furthermore, the cryptosystem should also be sensitive to the plain-image. A bit difference in any pixel should lead to a completely different encryption result. Figures 5(c) and (d) show the decrypted images using a wrong key with only a 10^{-14} difference in x_0 and $\widetilde{x}_0^{(1)}$, respectively. Figure 5(e) presents a completely different encrypted image if 10^{-14} is added to y_0 while Fig. 5(f) depicts the corresponding decrypted image using the original y_0.

3.3 Statistical analysis

In this subsection, we perform statistical analysis using another image Cameraman of size 256×256, as shown in Fig. 6(a).

3.3.1 Histogram

The statistical properties of an image can be characterized by the histogram showing the distribution of the pixel values [19]. Figure 6(b) is the histogram of the plain-image shown in Fig. 6(a). Figure 6(c) shows the histogram of the encrypted image. It is different from Fig. 6(d) which corresponds to a small change in the key y_0. Figure 6(e) depicts the histogram of the decrypted image using a wrong key. From these histograms, we can conclude that the proposed chaotic encryption algorithm can effectively flatten the histogram [9].

Table 1 Correlation coefficients

Direction	Plain-image	Cipher-image
Diagonal	0.9704	−0.0986
Horizontal	0.9759	−0.0630
Vertical	0.9820	0.0509

3.3.2 Correlation

The correlation of two adjacent pixels in a natural image is high. Therefore, it should be reduced by a permutation of the image pixels. If the encryption algorithm is effective, the encrypted image should have a low correlation among neighboring pixels. To evaluate the degree of correlation, 2500 pairs of adjacent pixels in horizontal, vertical and diagonal directions are randomly selected. The correlation coefficients are computed using (12) and the corresponding results are listed in Table 1. Figures 7(a) and (b) show the correlation distribution of two diagonal adjacent pixels in the plain-image and the cipher-image, respectively.

$$r_{xy} = \frac{\operatorname{cov}(x, y)}{\sqrt{D(x)D(y)}} \tag{12}$$

where $\operatorname{cov}(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$, $D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$, $E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i$. where x_i and y_i represent the gray values of two adjacent pixels.

3.4 Differential analysis

The strength of an image encryption algorithm against differential attack is usually measured by two indicators. The number of pixels change rate (NPCR) denotes the number of different pixels in the cipher-image when a pixel of the plain-image is altered. The unified average changing intensity (UACI) refers to the average intensity of the differences between the plain-image and the cipher-image These two performance indicators are calculated by (13) and (14), respectively. After one encryption round of the proposed algorithm is applied to the Lena image, we have NPCR = 99.619 % and UACI = 33.482 % which are

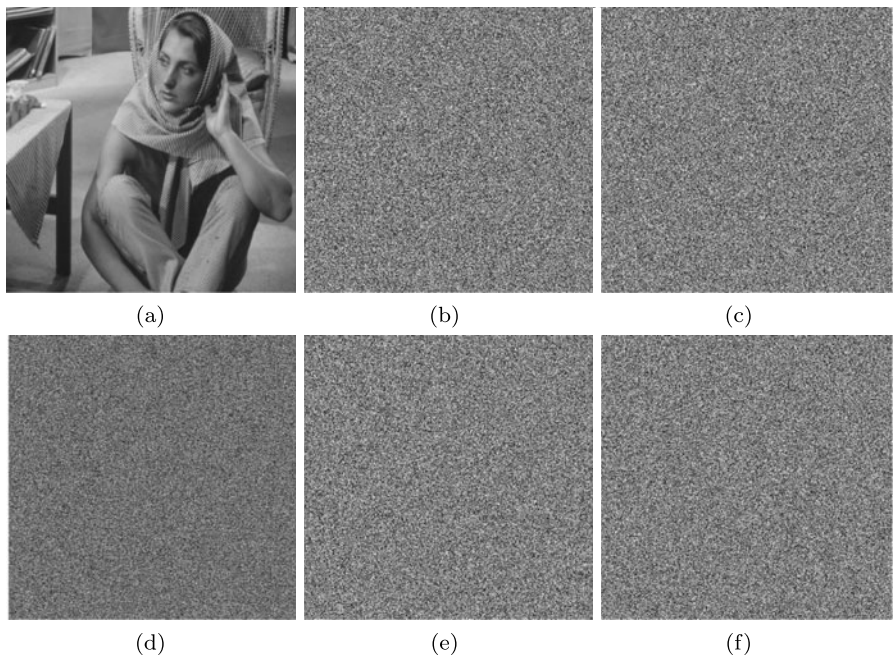

Fig. 5 Encryption test: (**a**) Barb image of size 500×512, (**b**) encrypted image, (**c**) decrypted image with $x_0 = 1.36 + 10^{-14}$, (**d**) decrypted image with $\widetilde{x}_0^{(1)} = 0.543 + 10^{-14}$, (**e**) encrypted image with $y_0 = -3.87 + 10^{-14}$, (**f**) decrypted image from (**e**) with $y_0 = -3.87$

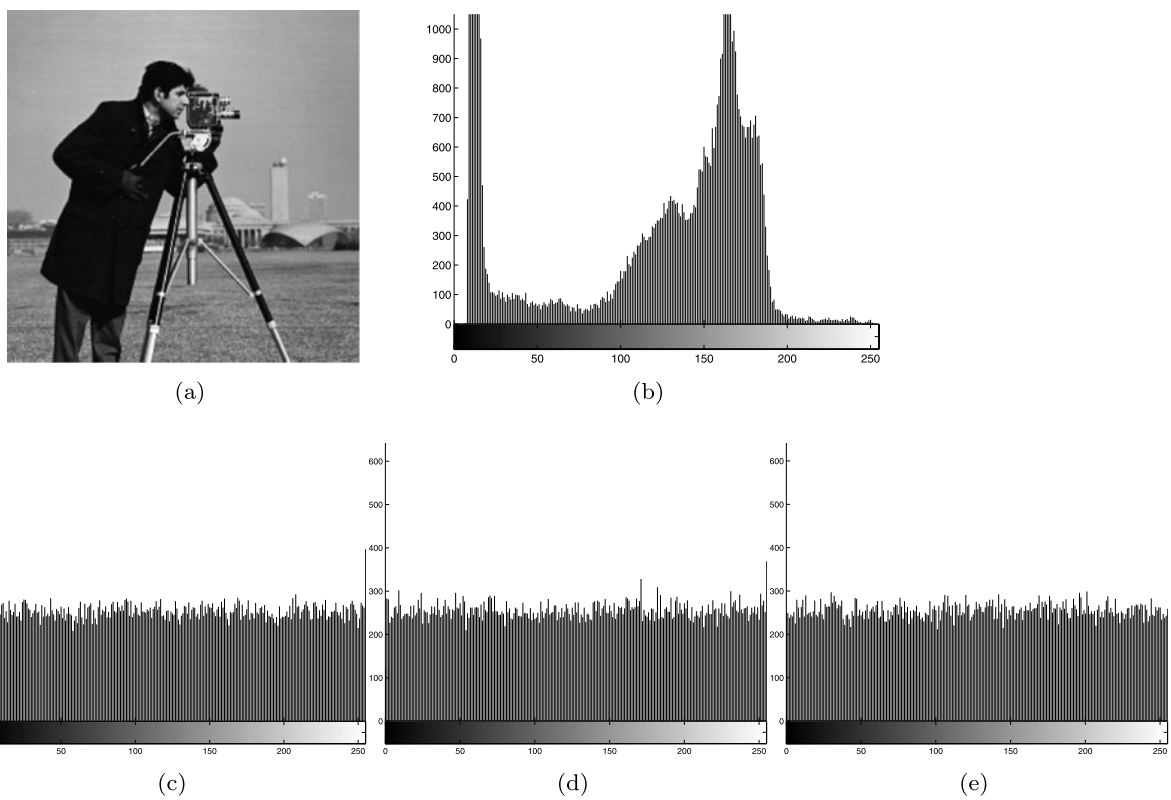

Fig. 6 Histogram: (**a**) Cameraman of size 256×256, (**b**) histogram of (**a**), (**c**) histogram of the encrypted image, (**d**) histogram of the encrypted image using a different key y_0, (**e**) histogram of the decrypted image using a wrong key y_0

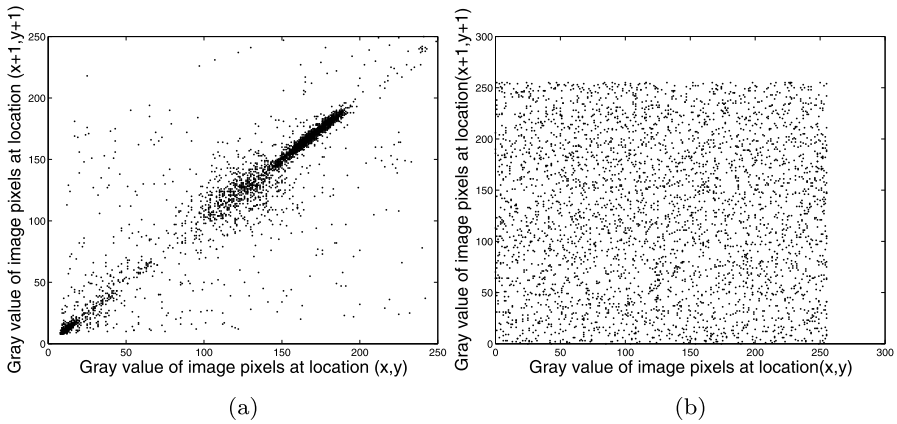

Fig. 7 Correlation of two diagonal adjacent pixels: (**a**) plain-image, (**b**) cipher-image

Table 2 UACI and NPCR results (in %) using different plain-images

Image	Lena	Barb	Cameraman	Boat
UACI	33.482	36.944	33.454	35.427
NPCR	99.619	99.739	99.617	99.681

close to the ideal values. The results for other plain-images can be found in Table 2 while the values at different number of encryption rounds using the Boat image of size 512×512 are listed in Table 3. In addition, the proposed scheme can well resist known-plaintext and chosen-plaintext attacks because the parameter key is related to the plain-image in the diffusion operation:

$$NPCR = \frac{\sum_{ij} D(i,j)}{M \times N} \times 100\ \% \tag{13}$$

$$UACI = \frac{1}{M \times N}\left[\sum_{i,j} \frac{|C_1(i,j) - C_2(i,j)|}{255}\right] \times 100\ \% \tag{14}$$

where $D(i,j) = 0$ if $C_1(i,j) = C_2(i,j)$; otherwise, $D(i,j) = 1$.

3.5 Speed analysis

In practical image transmission, the operation time must also be considered. We have measured the time cost in running the proposed encryption algorithm. It takes less than 0.14 s to encrypt an image of size 256×256 and 0.47 s for 512×512. Therefore, our algorithm is fast enough for practical applications.

Table 3 UACI, NPCR results (in %) and the running times at different number of rounds

Rounds	1	2	3	4	10
UACI	35.427	33.487	33.440	33.502	33.446
NPCR	99.681	99.625	99.617	99.620	99.618
Running time	0.47 s	0.57 s	0.67 s	0.77 s	1.36 s

4 Conclusion

A novel image encryption scheme based on time-delay and hyperchaotic system has been proposed. Time delay is introduced in the four chaotic sequences generated by a hyperchaotic system. Moreover, a position index permutation method is suggested which leads to a better permutation effect than existing methods. Together with the diffusion operation, our cryptosystem provides an effective and efficient way for protecting the privacy of digital images. These properties are justified by the experimental results on statistical, differential and running time analyses. Our encryption scheme also possesses the flexibility of using other hyperchaotic systems as the choice of the chaotic system is independent of the architecture of the cryptosystem.

Acknowledgement The work described in this paper was fully supported by a grant from CityU (Project No. 7008106) and the Natural Science Foundation of Guangdong Ocean University of P.R. China (No. 1212334).

References

1. Wang, X.Y., Wang, M.J.: A hyperchaos generated from Lorenz system. Physica A **387**, 3751–3758 (2008)

2. Yuen, C.H., Wong, K.W.: A chaos-based joint image compression and encryption scheme using DCT and SHA-1. Appl. Soft Comput. **11**, 5092–5098 (2011)
3. Francois, M., Grosges, T., Barchiesi, D., Erra, R.: A new image encryption scheme based on a chaotic function. Signal Process. Image Commun. **27**, 249–259 (2012)
4. Chen, G.R., Mao, Y.B., Chui, C.K.: A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Solitons Fractals **21**, 749–761 (2004)
5. Matthews, R.: On the derivation of a chaotic encryption algorithm. Cryptologia **4**, 29–42 (1989)
6. Niu, Y.J., Wang, X.Y., Wang, M.J., Zhang, H.G.: A new hyperchaotic system and its circuit implementation. Commun. Nonlinear Sci. Numer. Simul. **15**, 3518–3524 (2010)
7. Tong, X.J.: The novel bilateral-diffusion image encryption algorithm with dynamical compound chaos. J. Syst. Softw. **85**, 850–858 (2012)
8. Ye, R.S.: A novel chaos-based image encryption scheme with an efficient permutation diffusion mechanism. Opt. Commun. **284**, 5290–5298 (2011)
9. Mirzaei, O., Yaghoobi, M., Irani, H.: A new image encryption method: parallel sub-image encryption with hyper chaos. Nonlinear Dyn. **67**, 557–566 (2012)
10. Huang, X.L.: Image encryption algorithm using chaotic Chebyshev generator. Nonlinear Dyn. **67**, 2411–2417 (2012)
11. Gao, T.G., Chen, Z.Q.: A new image encryption algorithm based on hyper-chaos. Phys. Lett. A **372**, 394–400 (2008)
12. Zhu, C.X.: A novel image encryption scheme based on improved hyperchaotic sequences. Opt. Commun. **285**, 29–37 (2012)
13. Kanso, A., Ghebleh, M.: A novel image encryption algorithm based on a 3D chaotic map. Commun. Nonlinear Sci. Numer. Simul. **17**, 2943–2959 (2012)
14. Bigdeli, N., Farid, Y., Afshar, K.: A novel image encryption/decryption scheme based on chaotic neural networks. Eng. Appl. Artif. Intell. **25**, 753–765 (2012)
15. Hu, J.K., Han, F.L.: A pixel-based scrambling scheme for digital medical images protection. J. Netw. Comput. Appl. **32**, 788–794 (2009)
16. Wong, K.W., Kwok, B.S.H., Yuen, C.H.: An efficient diffusion approach for chaos-based image encryption. Chaos Solitons Fractals **41**, 2652–2663 (2009)
17. Pareek, N.K., Patidar, V., Sud, K.K.: Image encryption using chaotic logistic map. Image Vis. Comput. **24**, 926–934 (2006)
18. Fridrich, J.: Symmetric ciphers based on two-dimensional chaotic maps. Int. J. Bifurc. Chaos **8**, 1259–1284 (1998)
19. Wang, X.Y., Zhao, J.F., Liu, H.J.: A new image encryption algorithm based on chaos. Opt. Commun. **285**, 562–566 (2012)
20. Hussain, I., Shah, T., Gondal, M.A.: Image encryption algorithm based on $PGL(2, GF(2^8))$ S-boxes and TD-ERCS chaotic sequence. Nonlinear Dyn. (2012). doi:10.1007/s11071-012-0440-0
21. Shah, T., Hussain, I., Gondal, M.A., Mahmood, H.: Statistical analysis of S-box in image encryption applications based on majority logic criterion. Int. J. Phys. Sci. **6**, 4110–4127 (2011)
22. Hussain, I., Shah, T., Gondal, M.A.: An efficient image encryption algorithm based on S_8 S-box transformation and NCA map. Opt. Commun. (2012). doi:10.1016/j.optcom.2012.06.011
23. Hui, J.: Chaotic digital image encrypting method, involves implementing key coding treatment on processed array according to chaotic cryptography theory, and changing encrypted array into digital images, China Patent, CN101344960-A, 2008-8-20
24. Sobottka, M.: Encryption algorithm for digital representation of visual images, comprises chaotic sequence of digits of irrational roots based on chaotic dynamics of arrays of sequences, Brazil Patent, BR200703237-A, 2007-7-6