Entropy **2014**, *16*, 770-788; doi:10.3390/e16020770

OPEN ACCESS

entropy

ISSN 1099-4300

www.mdpi.com/journal/entropy

Article

A Symmetric Chaos-Based Image Cipher with an Improved Bit-Level Permutation Strategy

Chong Fu [1,*], Jun-Bin Huang [2], Ning-Ning Wang [1], Qi-Bin Hou [1] and Wei-Min Lei [1]

[1] School of Information Science and Engineering, Northeastern University, Shenyang 110004, China; E-Mails: bamboo7044@gmail.com (N.W.); andrewhoux@hotmail.com (Q.H.); leiweimin@ise.neu.edu.cn (W.L.)

[2] Department of Computer Science, University of Southern California, Los Angeles, CA 90089-0911, USA; E-Mail: pikaleize@163.com

* Author to whom correspondence should be addressed; E-Mail: fuchong@ise.neu.edu.cn; Tel.: +86-24-2338-8825.

Received: 22 November 2013; in revised form: 3 January 2014 / Accepted: 27 January 2014 / Published: 10 February 2014

Abstract: Very recently, several chaos-based image ciphers using a bit-level permutation have been suggested and shown promising results. Due to the diffusion effect introduced in the permutation stage, the workload of the time-consuming diffusion stage is reduced, and hence the performance of the cryptosystem is improved. In this paper, a symmetric chaos-based image cipher with a 3D cat map-based spatial bit-level permutation strategy is proposed. Compared with those recently proposed bit-level permutation methods, the diffusion effect of the new method is superior as the bits are shuffled among different bit-planes rather than within the same bit-plane. Moreover, the diffusion key stream extracted from hyperchaotic system is related to both the secret key and the plain image, which enhances the security against known/chosen plaintext attack. Extensive security analysis has been performed on the proposed scheme, including the most important ones like key space analysis, key sensitivity analysis, plaintext sensitivity analysis and various statistical analyses, which has demonstrated the satisfactory security of the proposed scheme.

Keywords: image cipher; bit-level permutation; 3D cat map; hyperchaotic system

1. Introduction

In recent years, various chaos-based image encryption algorithms have been proposed to meet the increasing demand for real-time secure image transmission over open channels. This is because conventional block ciphers, such as Triple-DES, AES and IDEA, do not have high performance in dealing with digital images which are mainly characterized by the bulk data capacity and high redundancy. Chaotic maps or systems, with the characteristics of sensitivity to initial conditions and control parameters, ergodicity, pseudo-randomness, *etc*., have drawn researchers' attention because such features naturally satisfy the essential design principles of a cryptosystem. Making use of these favorable characteristics, the algorithms based on chaotic systems have shown superior properties in security and complexity. In 1998, Fridrich [1] and Scharinger [2] proposed the first two chaos-based image encryption schemes with confusion-diffusion (or permutation-substitution) architecture, which are two essential properties of the operation of a secure cipher as identified by Claude Shannon in his masterpiece *Communication Theory of Secrecy Systems* [3]. Under this structure, the pixels of a plain image are firstly rearranged in a secret order with the purpose of estimating the strong relationship between adjacent pixels. Three typical area-preserving invertible chaotic maps □ baker map, Arnold cat map and Chirikov standard map □ are usually employed to fulfill this goal. Then in the diffusion stage, the pixel values are altered sequentially and the modification made to a particular pixel usually depends on the accumulated effect of all the previous pixel values, so as to diffuse the influence of each pixel over the whole cipher image. Various discrete and continuous chaotic systems such as logistic map, Chebyshev map, Lorenz system and hyperchaotic system can be employed to generate pseudorandom keystreams for diffusion. The permutation and the whole permutation-diffusion operations are usually iterated multiple times so as to achieve a satisfactory security level.

Following their pioneering work, a growing number of chaos-based image cryptosystems realized on variety of permutation-diffusion architectures utilizing different chaotic systems, their cryptanalysis, and improvements have been proposed [4–24]. We refer the readers to our recent contributions [21,23,25] for a brief description of those achievements. Very recently, bit-level permutation algorithms were suggested by some scholars [25–29]. As the permutation is performed on the bit-plane rather than the pixel-plane, the bit-level permutation has the effects of both confusion and diffusion. As a result, the workload of the time-consuming diffusion stage is reduced, and hence the cryptosystem performance is improved. However, those proposed schemes shuffle each bit-plane of an image independently. Accordingly, the bits distribution of a bit-plane significantly affects the diffusion effect, *i.e.*, if a bit-plane contains pixels that are nearly all 1s or 0s, the introduced diffusion effect will be negligible. To further enhance the diffusion effect introduced in the permutation stage, this paper proposes an improved bit-level permutation strategy which shuffles the bits among different bit-planes rather than within the same bit-plane.

Apart from performance considerations, security is the other essential issue. It has been reported that many proposed schemes have been successfully analyzed due to either the structural flaws or existence of weak keys in the algorithms. Table 1 summarizes some typical approaches to cryptanalysis of permutation-diffusion type image ciphers.

Table 1. Some typical approaches on cryptanalysis of permutation-diffusion type image ciphers.

Approaches	Cryptanalyzed by	Attacks employed
Fridrich (1998) [1]	Solak *et al.* (2010) [30]	chosen-ciphertext
Chen *et al.* (2004) [4]	Wang *et al.* (2005) [31]	chosen-plaintext
Pareek *et al.* (2006) [5]	Li *et al.* (2009) [32]	known/chosen-plaintext
Gao *et al.* (2008) [10]	Rhouma *et al.* (2008) [33]	chosen-plaintext/ciphertext
Tong *et al.* (2008) [11]	Li *et al.* (2009) [34]	chosen-plaintext
Patidar *et al.* (2009) [14]	Rhouma *et al.* (2010) [35]	known-plaintext
Zhu (2012) [22]	Ozkaynak *et al.* (2012) [36]; Li *et al.* (2013) [37]	known/chosen-plaintext

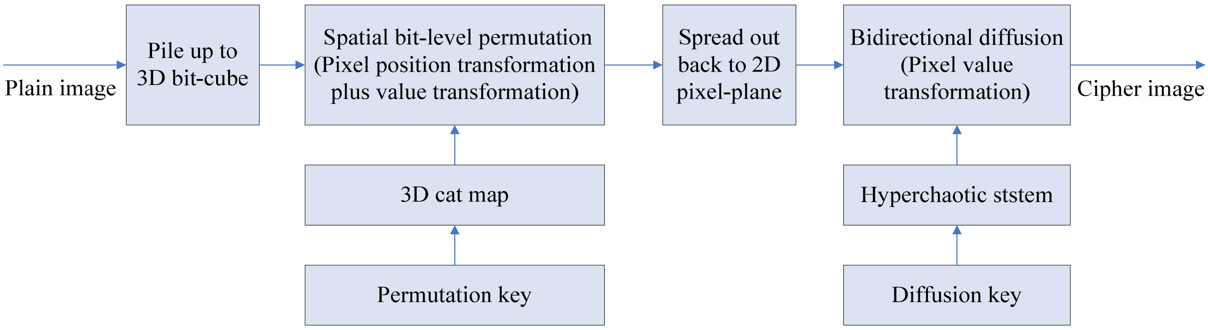
As can be seen from Table 1, these cryptanalysis works almost exclusively use the known/chosen plaintext attack. This is because the diffusion key stream used in most schemes is solely determined by the key, which means that the same key stream is used to encrypt different plain images unless a different key is used. Unfortunately, all these cryptosystems are non one-time pad. The key stream can be easily determined by encrypting some special images (*i.e.*, an all-white or all-black image) and then comparing them with the corresponding cipher images. To address this problem, Wang *et al.* [16] proposed a plain image related key stream generation scheme. In their scheme, the key stream elements are extracted from multiple times iteration of a chaotic map, and the iteration times is determined by plain pixel values. However, as the diffusion procedure, or more precisely, the key stream generation procedure is the highest cost of the whole cryptosystem, the extra iteration operation obviously degrades the performance. In the present paper, the key stream is associated with the plain image by circularly shifting each quantified element under the control of plain pixel. As the bitwise operations are extremely fast for the processor to handle, the execution time increment is negligible. Compared with ordinary chaotic systems, hyperchaotic systems, possessing more than one positive Lyapunov exponents, have more complex dynamical behaviors and number of system variables. This implies that cryptosystems built upon hyperchaotic system have stronger unpredictability and larger key space, which are essential for an effective cipher.

The remainder of this paper is organized as follows: in the next section, the architecture and diffusion mechanism of the proposed cryptosystem are introduced and discussed. The detailed 3D cat map-based spatial bit-level permutation strategy is described in Section 3, followed by the hyperchaotic system based diffusion algorithm in Section 4. In Section 5, we analyze the security of the proposed image cipher and evaluate its performance through key space, statistical, key sensitivity, plaintext sensitivity, and speed analysis. Finally, conclusions are drawn in the last section.

2. The Proposed Architecture

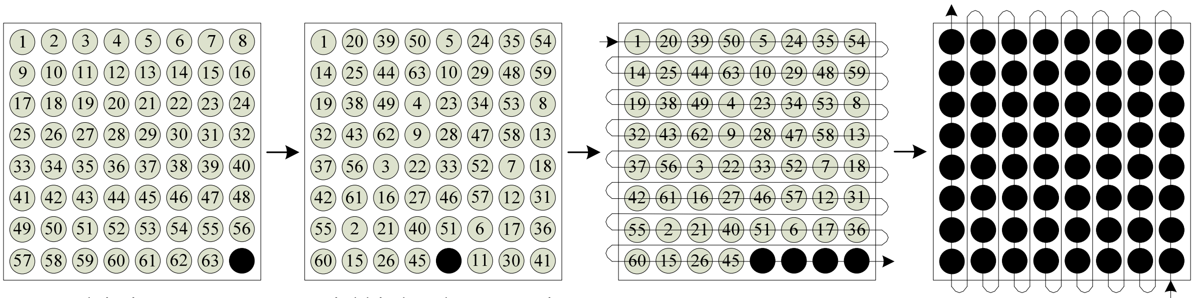
The architecture of the proposed image encryption scheme is shown in Figure 1. As illustrated in this Figure the proposed cryptosystem consists of a single round spatial bit-level permutation and bidirectional diffusion. The plain image is firstly extended and piled up to a bit-cube. Next in the permutation stage, all the bits in the bit-cube are shuffled spatially by using a 3D cat map. Then the shuffled bit-cube is spread out back to pixel plane. In the successive bidirectional diffusion stage, the bit-level shuffled image is sequentially masked by a key stream extracted from hyperchaotic system in order from left to right, top to bottom and bottom to top, right to left, respectively, and finally the output cipher image is produced.

Figure 1. Architecture of the proposed cryptosystem.


Plain image
Pile up to 3D bit-cube
Spatial bit-level permutation (Pixel position transformation plus value transformation)
Spread out back to 2D pixel-plane
Bidirectional diffusion (Pixel value transformation)
Cipher image
3D cat map
Permutation key
Hyperchaotic ststem
Diffusion key

The diffusion mechanism of our cryptosystem is illustrated by Figure 2. Here we assume a worst case that a slight change is made to the lower right corner of the plain image at pixel (M, N). In the permutation stage, the change in pixel (M, N) is shuffled to pixel (M', N') under the control of the 3D cat map. Next in the first stage of bidirectional diffusion, the change is spread out to all pixels subsequent to (M', N'). Then in the second stage, the diffused pixels produced in the previous step are spread out to the whole cipher image. The effectiveness of above diffusion mechanism will be qualitatively evaluated in Section 5.4.

Figure 2. The diffusion mechanism of the proposed scheme.



1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	●

plain image

1	20	39	50	5	24	35	54
14	25	44	63	10	29	48	59
19	38	49	4	23	34	53	8
32	43	62	9	28	47	58	13
37	56	3	22	33	52	7	18
42	61	16	27	46	57	12	31
55	2	21	40	51	6	17	36
60	15	26	45	●	11	30	41

spatial bit-level permutation

1	20	39	50	5	24	35	54
14	25	44	63	10	29	48	59
19	38	49	4	23	34	53	8
32	43	62	9	28	47	58	13
37	56	3	22	33	52	7	18
42	61	16	27	46	57	12	31
55	2	21	40	51	6	17	36
60	15	26	45	●	●	●	●

bidirectional diffusion (1st stage)

●	●	●	●	●	●	●	●
●	●	●	●	●	●	●	●
●	●	●	●	●	●	●	●
●	●	●	●	●	●	●	●
●	●	●	●	●	●	●	●
●	●	●	●	●	●	●	●
●	●	●	●	●	●	●	●
●	●	●	●	●	●	●	●

bidirectional diffusion (2nd stage)

3. Spatial Bit-Level Permutation Strategy Using 3D Cat Map

Before implementing spatial bit-level permutation, the plain image needs to be extended and piled up to a bit-cube. In digital imaging, color depth or bit depth is the number of bits used to indicate the color of a single pixel in a bitmapped image. For example, in 8-bit color mode, the color monitor uses 8 bits for each pixel, making it possible to display 2 to the 8th power (256) different colors or shades of gray. Therefore, a D-bit image of size $M \times N$ can be piled up to a bit-cube with side length $L_c = ceil(\sqrt[3]{M \times N \times D})$, where $ceil(x)$ returns the value of x to the nearest integers greater than or equal to x. The insufficient $R = L_c^3 - M \times N \times D$ bits are padded with pseudo-random binary numbers generated by chaotic logistic map. The logistic map is described by:

$$x_{n+1} = \mu x_n(1-x_n),\ \mu \in [0,4], \tag{1}$$

where μ and x are parameter and state variable, respectively, and the system shows chaotic behavior for $\mu \in [3.57, 4]$. The padding bits are quantified from the current state of the map with $\mu = 4$ according to:

$$p_n = \begin{cases} 0, & \text{for } x < 0.5, \\ 1, & \text{for } x \geq 0.5. \end{cases} \tag{2}$$

The quantified sequence p_n follows Bernoulli distribution because of the symmetric property of the invariant density for the logistic map with $\mu = 4$, which is described by:

$$\rho(x) = \begin{cases} \dfrac{1}{\pi\sqrt{x(1-x)}}, & \text{for } 0 < x < 1, \\ 0, & \text{otherwise}. \end{cases} \tag{3}$$

The padded bits are just discarded during the deciphering process. Then all the bits in the bit-cube are shuffled spatially by using 3D cat map. The so-call 3D cat map is a bijection of the unit cube $I \times I \times I$ onto itself, as described by:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \boldsymbol{A} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \ mod\ 1, \tag{4}$$

where:

$$\boldsymbol{A} = \begin{bmatrix} 1 + a_x a_z b_y & a_z & a_y + a_x a_z + a_x a_y a_z b_y \\ b_z + a_x b_y + a_x a_z b_y b_z & a_z b_z + 1 & a_y a_z + a_x a_y a_z b_y b_z + a_x a_z b_z + a_x a_y b_y + a_x \\ a_x b_x b_y + b_y & b_x & a_x a_y b_x b_y + a_x b_x + a_y b_y + 1 \end{bmatrix}$$

and $mod(x, y)$ divides x by y and returns the remainder of the division. The map is invertible and area-preserving as det $|\boldsymbol{A}| = 1$.

In order to incorporate the 3D cat map into the spatial permutation that operates on a 3D lattice of finitely many bits, it has to be discretized, while reserving some of its useful features such as the mixing property and the sensitivity to initial conditions and parameters. The discretized version 3D cat map can be obtained simply by changing the range of (x, y, z) from the unit cube $I \times I \times I$ to the discrete 3D lattice $N \times N \times N$, as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \boldsymbol{A} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \ mod\ N, \tag{5}$$

where N is the side length of a bit-cube. The combination of the six control parameters $(a_x, a_y, a_z, b_x, b_y, b_z)$ and the number of iterations m are used as the permutation key. As there only exist a linear transformation and *mod* function, it is very efficient to shuffle a bit-cube by using the 3D cat map.

The inverse transform of the 3D cat map used for deciphering is given by:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \boldsymbol{A}^{-1} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} mod\ N, \tag{6}$$

where:

$$\boldsymbol{A}^{-1} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ B_{11} & B_{12} & B_{13} \\ C_{11} & C_{12} & C_{13} \end{bmatrix} \Big/ D,$$

$$A_{11} = a_z b_z a_y b_y + a_z b_z + a_y b_y + 1 - b_x a_y a_z, A_{12} = -(a_z a_y b_y + a_z - b_x a_y), A_{13} = a_y(a_z a_z - a_z b_z - 1),$$
$$B_{11} = -b_z a_x a_y b_x b_y - b_z a_x b_x - b_z a_y b_y - b_z + a_z a_x a_y b_x b_y + a_z a_y b_y, B_{12} = a_x b_x + 1,$$
$$B_{13} = -(a_y a_z - a_x a_y a_z b_y b_z + a_x + a_x a_z a_z b_y a_y - a_y b_z),$$
$$C_{11} = b_x b_z - a_z b_z b_y - b_y, C_{12} = -b_x + a_z b_y, C_{13} = 1,$$
$$D = 1 - b_x a_y a_z - a_z b_z a_y b_y + b_z b_x a_y + b_y a_y a_z a_z.$$

The application of the proposed and conventional bit-level permutation methods is demonstrated in Figure 3. Figure 3(a) shows the 512 × 512 pixels Lena image with 256 gray levels. Figure 3(b) shows the results of applying the proposed permutation method once, and the permutation key is ($a_x = 20$, $a_y = 17$, $a_z = 42$, $b_x = 53$, $b_y = 5$, $b_z = 20$). The test images after applying the 2D cat map-based conventional bit-level permutation method once, two and three times are shown in Figures 3(c–e), respectively, and the key used for the eight bit-planes is {($a_1 = 40$, $b_1 = 9$), ($a_2 = 35$, $b_2 = 8$), ($a_3 = 30$, $b_3 = 7$), ($a_4 = 25$, $b_4 = 6$), ($a_5 = 20$, $b_5 = 5$), ($a_6 = 15$, $b_6 = 4$), ($a_7 = 10$, $b_7 = 3$), ($a_8 = 5$, $b_8 = 2$)}. As can be seen from Figure 3, after only applying the proposed permutation strategy once, the correlation among the adjacent pixels in the plain image is effectively eliminated and the resultant image is completely unrecognizable. Meanwhile, the randomness of the resultant image produced by the proposed permutation method with one iteration cycle is even superior to that of the resultant image produced by the conventional bit-level permutation method with three iteration cycles. A thorough quantitative comparison will be given in Section 5.2 to demonstrate the superior diffusion effect introduced by the proposed permutation method.

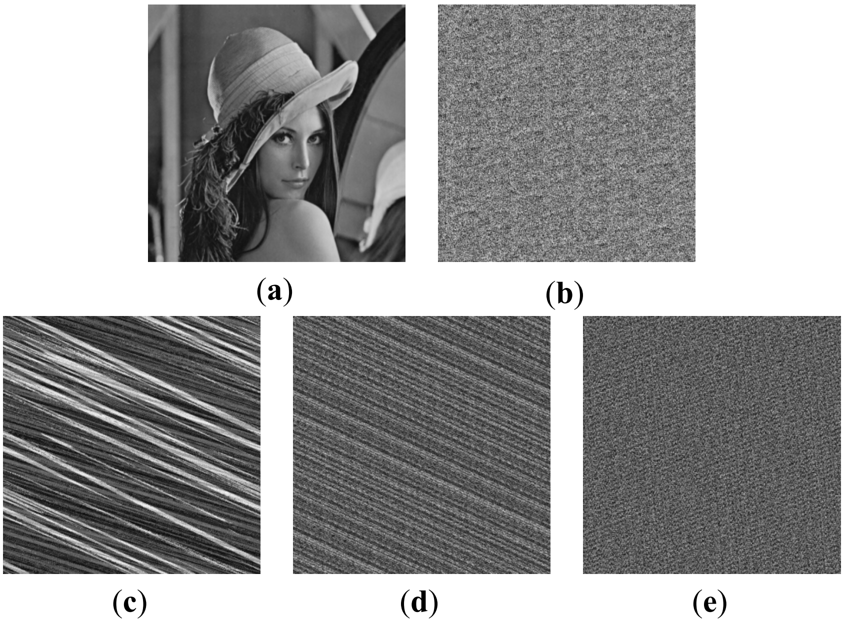
4. Image Diffusion Using Hyper-Chaotic System

In the present paper, a hyperchaotic system proposed by Jia [38] is employed to generate the key stream for diffusion. The system is described by:

$$\begin{cases} \dot{x} = -a(x-y) + u, \\ \dot{y} = -xz + rx - y, \\ \dot{z} = xy - bz, \\ \dot{u} = -xz + du, \end{cases} \tag{7}$$

where a, r, b are the system parameters, and d is the control parameter. When $a = 10$, $r = 28$, $b = 8/3$ and $0.85 < d < 1.3$, the system exhibits chaotic behavior, and the projections of its attractor in the six different planes are shown in Figure 4. The initial state values (x_0, y_0, z_0, u_0) are used as the diffusion key.

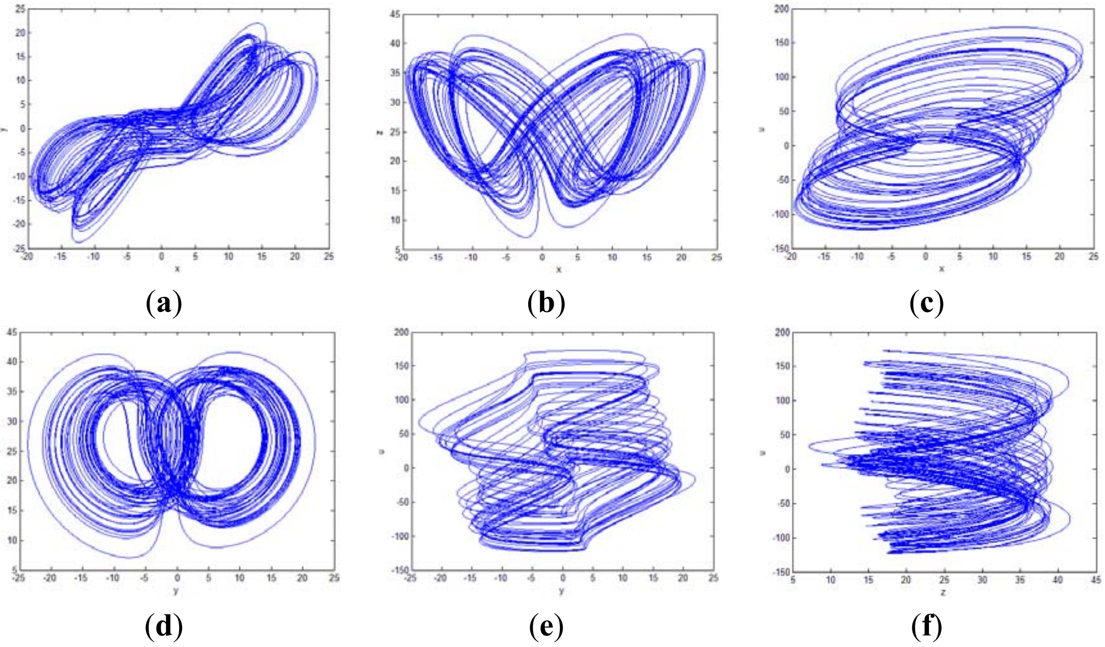
Figure 3. The application of the proposed bit-level permutation method and the conventional pixel-level permutation method. (**a**) The Lena image 512 × 512 pixels with 256 gray levels. (**b**) The Lena image after applying the proposed permutation method once. (**c**) The Lena image after applying the conventional bit-level permutation method once. (**d**) The Lena image after applying the conventional bit-level permutation method two times. (**e**) The Lena image after applying the conventional bit-level permutation method three times.



(**a**) (**b**)

(**c**) (**d**) (**e**)

Figure 4. Projections of the attractor of the employed hyperchaotic system. (**a**) *x*-*y* plane. (**b**) *x*-*z* plane. (**c**) *x*-*u* plane. (**d**) *y*-*z* plane. (**e**) *y*-*u* plane. (**f**) *z*-*u* plane.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

The detailed diffusion procedure is described as follows:

Step 1: The shuffled bit-cube is spread out back to pixel-plain and arranged to a vector $p=\{p_1, p_2, \ldots, p_{M\times N}\}$ in the order from left to right, top to bottom.

Step 2: Pre-iterate Equation (7) for N_0 times to avoid the harmful effect of transitional procedure, where N_0 is a constant. The equation is solved by using fourth-order Runge-Kutta method, as given by:

$$\begin{cases} x_{n+1} = x_n + (h/6)(K_1 + 2K_2 + 2K_3 + K_4), \\ y_{n+1} = y_n + (h/6)(L_1 + 2L_2 + 2L_3 + L_4), \\ z_{n+1} = z_n + (h/6)(M_1 + 2M_2 + 2M_3 + M_4), \\ u_{n+1} = u_n + (h/6)(N_1 + 2N_2 + 2N_3 + N_4), \end{cases} \tag{8}$$

where:

$$\begin{cases} K_j = -a(x_n - y_n) + u_n, \\ L_j = -x_n z_n + r x_n - y_n, \\ M_j = x_n y_n - b z_n, \\ N_j = -x_n z_n + d u_n, \\ (j = 1), \end{cases}$$

$$\begin{cases} K_j = -a\left[(x_n + hK_{j-1}/2) - (y_n + hL_{j-1}/2)\right] + (u_n + hN_{j-1}/2), \\ L_j = -(x_n + hK_{j-1}/2)(z_n + hM_{j-1}/2) + r(x_n + hK_{j-1}/2) - (y_n + hL_{j-1}/2), \\ M_j = (x_n + hK_{j-1}/2)(y_n + hL_{j-1}/2) - b(z_n + hM_{j-1}/2), \\ N_j = -(x_n + hK_{j-1}/2)(z_n + hM_{j-1}/2) + d(u_n + hN_{j-1}/2), \\ (j = 2,3), \end{cases}$$

$$\begin{cases} K_j = -a\left[(x_n + hK_{j-1}) - (y_n + hL_{j-1})\right] + (u_n + hN_{j-1}), \\ L_j = -(x_n + hK_{j-1})(z_n + hM_{j-1}) + r(x_n + hK_{j-1}) - (y_n + hL_{j-1}), \\ M_j = (x_n + hK_{j-1})(y_n + hL_{j-1}) - b(z_n + hM_{j-1}), \\ N_j = -(x_n + hK_{j-1})(z_n + hM_{j-1}) + d(u_n + hN_{j-1}), \\ (j = 4), \end{cases}$$

and the step h is chosen as 0.0005.

Step 3: The hyperchaotic system is iterated continuously. For each iteration, we can obtain four key stream elements from the current state of the hyperchaotic system according to:

$$k_{\varphi n} = mod[round((abs(\varphi_n) - floor(abs(\varphi_n))) \times 10^{14}), 2^D],\ \varphi \in \{x, y, z, u\}, \tag{9}$$

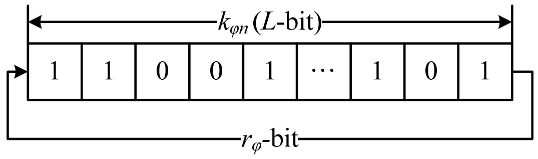
where *round*(*x*) rounds *x* to the nearest integers, *abs*(*x*) returns the absolute value of *x*, and *floor*(*x*) returns the value of *x* to the nearest integers less than or equal to *x*. In our scheme, all the variables are declared as 64-bit double-precision type, which has a 15-digit precision according to the IEEE floating-point standard, and therefore the decimal fractions of the variable is multiplied by 10^{14}.

Step 4: Let $p_{4\times(n-1)+m}$ (m = 1, 2, 3, 4) denote the currently operated pixel. Circularly shift $k_{\varphi n}$ left r_φ bits, as illustrated by Figure 5, where r_φ is determined by the previously operated plain pixel according to:

$$\begin{cases} r_x = mod(p_{4\times(n-1)}, 2^D), \\ r_y = mod(p_{4\times(n-1)+1}, 2^D), \\ r_z = mod(p_{4\times(n-1)+2}, 2^D), \\ r_u = mod(p_{4\times(n-1)+3}, 2^D). \end{cases} \tag{10}$$

One may set initial value p_0 as an arbitrary constant from 0 to 2^D.

Figure 5. The circular shift of key stream element.



Step 5: Calculate the cipher pixel value according to Equation (11):

$$\begin{cases} c_{4\times(n-1)+1} = k_{xn} \oplus \{[p_{4\times(n-1)+1} + k_{xn}]\ mod\ 2^D\} \oplus c_{4\times(n-1)}, \\ c_{4\times(n-1)+2} = k_{yn} \oplus \{[p_{4\times(n-1)+2} + k_{yn}]\ mod\ 2^D\} \oplus c_{4\times(n-1)+1}, \\ c_{4\times(n-1)+3} = k_{zn} \oplus \{[p_{4\times(n-1)+3} + k_{zn}]\ mod\ 2^D\} \oplus c_{4\times(n-1)+2}, \\ c_{4\times(n-1)+4} = k_{un} \oplus \{[p_{4\times(n-1)+4} + k_{un}]\ mod\ 2^D\} \oplus c_{4\times(n-1)+3}, \end{cases} \tag{11}$$

where $c_{4\times(n-1)+m}$ (m = 1, 2, 3, 4) are the output cipher pixels, and ⊕ performs bit-wise exclusive OR operation. The initial value c_0 may also be set as a constant.

Step 6: Return to ***Step 3*** until all the pixels in vector p are encrypted. Finally, the cipher pixel set $c = \{c_1, c_2, \ldots, c_{M\times N}\}$ is reshaped back into a $M \times N$ matrix and the cipher image is produced.

The decryption procedure is similar to that of the encryption process described above, and the inverse of Equation (11) is given by:

$$\begin{cases} p_{4\times(n-1)+1} = [k_{xn} \oplus c_{4\times(n-1)+1} \oplus c_{4\times(n-1)} + 2^D - k_{xn}]\ mod\ 2^D, \\ p_{4\times(n-1)+2} = [k_{yn} \oplus c_{4\times(n-1)+2} \oplus c_{4\times(n-1)+1} + 2^D - k_{yn}]\ mod\ 2^D, \\ p_{4\times(n-1)+3} = [k_{zn} \oplus c_{4\times(n-1)+3} \oplus c_{4\times(n-1)+2} + 2^D - k_{zn}]\ mod\ 2^D, \\ p_{4\times(n-1)+4} = [k_{un} \oplus c_{4\times(n-1)+4} \oplus c_{4\times(n-1)+3} + 2^D - k_{un}]\ mod\ 2^D. \end{cases} \tag{12}$$

The overall permutation-diffusion operations are usually performed for several rounds according to the security requirement. Obviously, the more rounds are processed, the more secure the encryption is, but at the expense of computations and time delays.

5. Security Analysis

A good cryptosystem should resist all kinds of known attacks, such as exhaustive search attack, statistical attack, known/chosen plaintext attack and differential attack. In this section, a thorough security analysis has been carried out to demonstrate the robustness of the proposed scheme, as discussed in the following.

5.1. Key Space Analysis

The key space is the total number of different keys that can be used in the encryption/decryption procedure. For an effective cryptosystem, the key space should be large enough to make the exhaustive search attack infeasible. As mentioned above, the key of the proposed cryptosystem is composed of two parts: permutation key *Key-P* and diffusion key *Key-D*. *key-P* consists of six integers $(a_x, a_y, a_z, b_x, b_y, b_z) \in [1, L_c]$ and the iteration times $m \in N+$, and therefore the size of *key-P* is $(L_c^6)^m$. *Key-D* is composed of four floating point numbers $(x_0, y_0, z_0, u_0) \in R$. As all the variables are declared as 64-bit double-precision type, the total number of possible values of *Key-D* is approximately $(10^{15})^4$. The two parts *key-P* and *key-D* are independent of each other, thus the key space of the proposed cryptosystem is:

$$Key_{total} = key\text{-}P \times key\text{-}D \approx (L_c^{\ 6})^m \times 10^{60}. \tag{13}$$

We take a 256 grayscale image of size 512 × 512 as an example, $L_c = ceil(\sqrt[3]{512 \times 512 \times 8}) = 128$. If we choose $m = 1$, the key space satisfies:

$$Key_{total} \approx 2^{241}, \tag{14}$$

which is far larger than that of most well-known block ciphers such as Triple-DES (168-bit), IDEA (128-bit) and AES (128-bit and 192-bit versions). Furthermore, this is just for one round of the several iterations, the increase of round numbers will further enlarge the key space. Therefore, it can be concluded that the proposed scheme is robust against exhaustive search attacks.

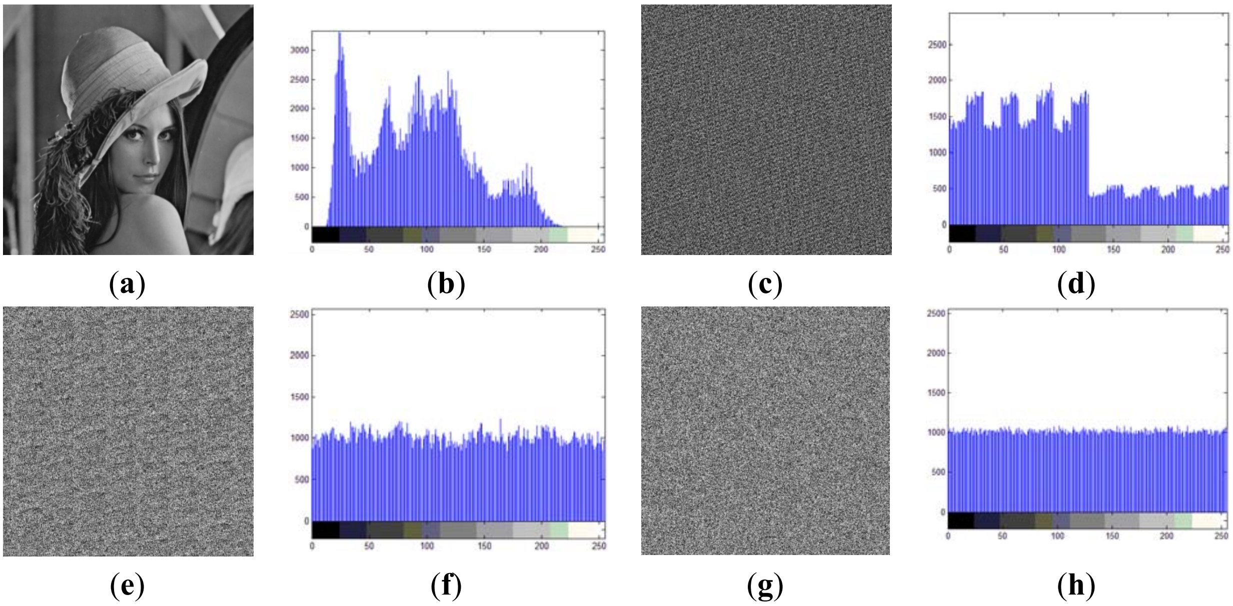
5.2. Statistical Analysis

Statistical analysis is a common and effective way to analyze a cryptosystem. Consequently, a good cipher should be robust against any statistical attack. In order to prove the security of the proposed image cryptosystem, the following statistical tests are performed.

5.2.1. Histogram

The frequency distribution of cipher pixel values is of great importance. It should hide the redundancy of the plain image and should not leak any information about the relationship between the plain image and the cipher image. Histogram, a key tool in image processing, is a graph showing the number of pixels in an image at each different intensity value found in that image. It can be visualised as if each pixel is placed in a bin corresponding to the colour intensity of that pixel. All of the pixels in each bin are then added up and displayed on a graph. Figure 6(a) shows the test image, and (c) and (e) show its shuffled images using the conventional bit-level permutation method and the proposed spatial bit-level permutation method, respectively, and (g) shows its output cipher image produced by the proposed cryptosystem. Their corresponding histograms are shown in Figures 6(b, d, f, h), respectively.

Figure 6. Histograms analysis. (**a**) plain image. (**b**) histogram of (**a**). (**c**) shuffled image using conventional bit-level permutation method. (**d**) histogram of (**c**). (**e**) shuffled image using the proposed spatial bit-level permutation method. (**f**) histogram of (**e**). (**g**) ciphered image produced by the proposed cryptosystem. (**h**) histogram of (**g**).



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

As can be seen from Figure 6, the histogram of the output cipher image is fairly evenly distributed over the scale, and therefore no information about the plain image can be gathered through histogram analysis. Meanwhile, though the histogram of the shuffled image produced by the proposed permutation method is not distributed in a perfectly uniform, its uniformity is much better than that of the image produced by the conventional bit-level permutation method owing to the superior diffusion effect introduced.

5.2.2. Information Entropy

Information entropy, the most important feature of randomness, is one of the fundamental criteria to measure the strength of a cryptosystem. To calculate the entropy $H(S)$ of a source s, we have:

$$H(S) = -\sum_{i=1}^{n} P(s_i)\log_2 P(s_i), \tag{15}$$

where S is a random variable with n outcomes $\{s_1, ..., s_n\}$ and $P(s_i)$ is the probability mass function of outcome s_i. Obviously, for a random image with 256 gray levels, the entropy should ideally be $H(S) = 8$. That is, if the entropy of a ciphered grayscale image is less than 8, there exists a certain level of predictability, which threatens its security.

Table 2 lists the entropies for the test image and its output cipher images produced by the proposed image cryptosystem, as well as its shuffled images using the conventional and proposed bit-level permutation methods. As can be seen from Table 2, the entropy of the output cipher image is very close to the theoretical value of 8. This means that there is no leakage of information from the proposed cryptosystem during its execution, and therefore the proposed cryptosystem is robust against entropy analysis. Furthermore, the entropy of the shuffled image using the proposed scheme is comparable with that of the output cipher image and significant superior to that of the shuffled image using the conventional scheme for the same reason discussed above.

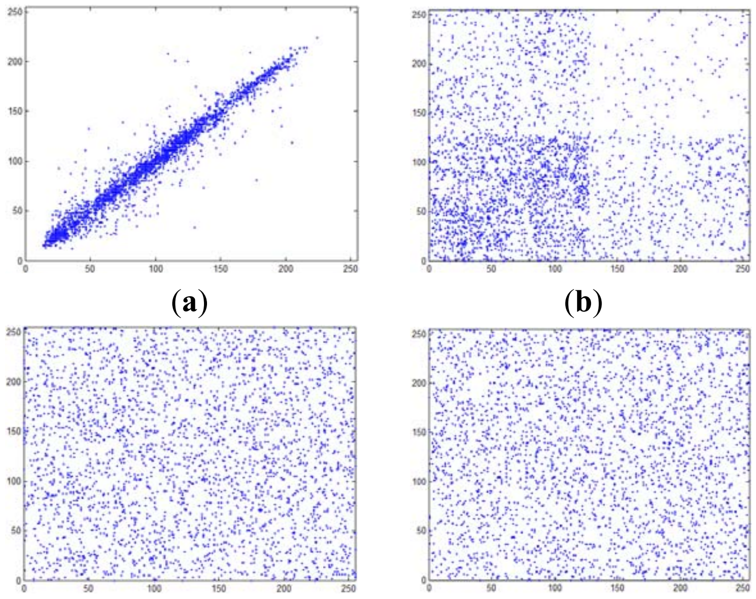
Table 2. Results of entropy analysis of the proposed image cryptosystem.

Plain image	**Shuffled image using conventional scheme**	**Shuffled image using proposed scheme**	**Output cipher image**
7.3640	7.7620	7.9959	7.9995

5.2.3. Correlation of Adjacent Pixels

Pixels in an ordinary image are usually highly correlated with their adjacent pixels either in horizontal, vertical or diagonal direction, but the correlation of the adjacent pixels in a cipher image should be as low as possible so as to resist correlation analysis. The correlation of adjacent pixels can be visually measured by the following procedure. First, randomly select P_0 pairs of adjacent pixels in each direction from the image, where P_0 is typically larger than 2,000. Then, plot the distribution of the adjacent pixels by using each pair as the values of the XY coordinate. The correlation distribution of two vertically adjacent pixels in the test image, its shuffled images using the conventional and proposed permutation methods, and its output cipher image produced by the proposed cryptosystem are shown in Figures 7(a–d), respectively. Similar results can be obtained for horizontally and diagonally adjacent pixels.

Figure 7. The visual testing of correlation of vertically adjacent pixels. (**a**) correlation of vertically adjacent pixels in the test image. (**b**) correlation of vertically adjacent pixels in the shuffled image using conventional permutation method. (**c**) correlation of vertically adjacent pixels in the shuffled image using the proposed permutation method. (**d**) correlation of vertically adjacent pixels in the output cipher image produced by the proposed cryptosystem.



(**a**) (**b**)

(**c**) (**d**)

To further quantify the correlations of adjacent pixels in an image, the correlation coefficient $r_{x,y}$ is calculated by using the following three formulas:

$$r_{xy} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2\right)\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - \overline{y})^2\right)}}, \tag{16}$$

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{17}$$

$$\overline{y} = \frac{1}{N}\sum_{i=1}^{N} y_i, \tag{18}$$

where x_i and y_i are grayscale values of the *i*th pair of adjacent pixels, and N denotes the total number of samples.

Table 3 lists the results of the correlation coefficients for horizontal, vertical and diagonal adjacent pixels in the four images. It's clear from Figure 7 and Table 3 that the strong correlation between adjacent pixels in the test image is completely eliminated in both the output cipher image and the shuffled image using the proposed permutation method. Moreover, though the correlation coefficients for the shuffled image produced by the conventional permutation method are comparable with that of the output cipher image, its visual testing results are relatively poor compared with the proposed permutation method.

Table 3. Results of correlation analysis of the proposed image cryptosystem.

Direction	Plain image	Shuffled image using conventional scheme	Shuffled image using proposed scheme	Output cipher image
Horizontal	0.9869	0.0232	0.0251	0.0201
Vertical	0.9768	0.0226	−0.0226	−0.0129
Diagonal	0.9679	−0.0511	0.0163	0.0057

5.3. Key Sensitivity Analysis

Another essential property required by a cryptosystem is key sensitivity, which ensures that no data can be recovered from the ciphertext even though there is only a minor difference between the encryption and decryption keys. To evaluate the key sensitivity property of the proposed cryptosystem, the test image (Figure 3(a)) is firstly encrypted using a randomly selected key ($a_x = 73$, $a_y = 40$, $a_z = 43$, $b_x = 74$, $b_y = 47$, $b_z = 63$, $x_0 = 5.00835323256446$, $y_0 = -5.76624087423199$, $z_0 = 6.72223702603131$, $u_0 = 5.26045903184283$), and the resultant cipher image is shown in Figure 8(a). Then the ciphered image is tried to be decrypted using eleven decryption keys, as listed in Table 4. The resultant deciphered images are shown in Figures 8(b–l), respectively, from which we can see that even an almost perfect guess of the key does not reveal any information about the plain image. Therefore, it can be concluded that the proposed image cryptosystem fully satisfies the key sensitivity requirement.

Table 4. Decryption keys used for the key sensitivity test.

Figure	Decryption key
8(b)	($a_x = 73$, $a_y = 40$, $a_z = 43$, $b_x = 74$, $b_y = 47$, $b_z = 63$, $x_0 = 5.00835323256446$, $y_0 = -5.76624087423199$, $z_0 = 6.72223702603131$, $u_0 = 5.26045903184283$)
8(c)	($\boldsymbol{a_x = 74}$, $a_y = 40$, $a_z = 43$, $b_x = 74$, $b_y = 47$, $b_z = 63$, $x_0 = 5.00835323256446$, $y_0 = -5.76624087423199$, $z_0 = 6.72223702603131$, $u_0 = 5.26045903184283$)
8(d)	($a_x = 73$, $\boldsymbol{a_y = 41}$, $a_z = 43$, $b_x = 74$, $b_y = 47$, $b_z = 63$, $x_0 = 5.00835323256446$, $y_0 = -5.76624087423199$, $z_0 = 6.72223702603131$, $u_0 = 5.26045903184283$)
8(e)	($a_x = 73$, $a_y = 40$, $\boldsymbol{a_z = 44}$, $b_x = 74$, $b_y = 47$, $b_z = 63$, $x_0 = 5.00835323256446$, $y_0 = -5.76624087423199$, $z_0 = 6.72223702603131$, $u_0 = 5.26045903184283$)
8(f)	($a_x = 73$, $a_y = 40$, $a_z = 43$, $\boldsymbol{b_x = 75}$, $b_y = 47$, $b_z = 63$, $x_0 = 5.00835323256446$, $y_0 = -5.76624087423199$, $z_0 = 6.72223702603131$, $u_0 = 5.26045903184283$)
8(g)	($a_x = 73$, $a_y = 40$, $a_z = 43$, $b_x = 74$, $\boldsymbol{b_y = 48}$, $b_z = 63$, $x_0 = 5.00835323256446$, $y_0 = -5.76624087423199$, $z_0 = 6.72223702603131$, $u_0 = 5.26045903184283$)
8(h)	($a_x = 73$, $a_y = 40$, $a_z = 43$, $b_x = 74$, $b_y = 47$, $\boldsymbol{b_z = 64}$, $x_0 = 5.00835323256446$, $y_0 = -5.76624087423199$, $z_0 = 6.72223702603131$, $u_0 = 5.26045903184283$)
8(i)	($a_x = 73$, $a_y = 40$, $a_z = 43$, $b_x = 74$, $b_y = 47$, $b_z = 63$, $\boldsymbol{x_0 = 5.00835323256447}$, $y_0 = -5.76624087423199$, $z_0 = 6.72223702603131$, $u_0 = 5.26045903184283$)
8(j)	($a_x = 73$, $a_y = 40$, $a_z = 43$, $b_x = 74$, $b_y = 47$, $b_z = 63$, $x_0 = 5.00835323256446$, $\boldsymbol{y_0 = -5.76624087423198}$, $z_0 = 6.72223702603131$, $u_0 = 5.26045903184283$)
8(k)	($a_x = 73$, $a_y = 40$, $a_z = 43$, $b_x = 74$, $b_y = 47$, $b_z = 63$, $x_0 = 5.00835323256446$, $y_0 = -5.76624087423199$, $\boldsymbol{z_0 = 6.72223702603132}$, $u_0 = 5.26045903184283$)
8(l)	($a_x = 73$, $a_y = 40$, $a_z = 43$, $b_x = 74$, $b_y = 47$, $b_z = 63$, $x_0 = 5.00835323256446$, $y_0 = -5.76624087423199$, $z_0 = 6.72223702603131$, $\boldsymbol{u_0 = 5.26045903184284}$)

Figure 8. Deciphered images using slightly different keys.

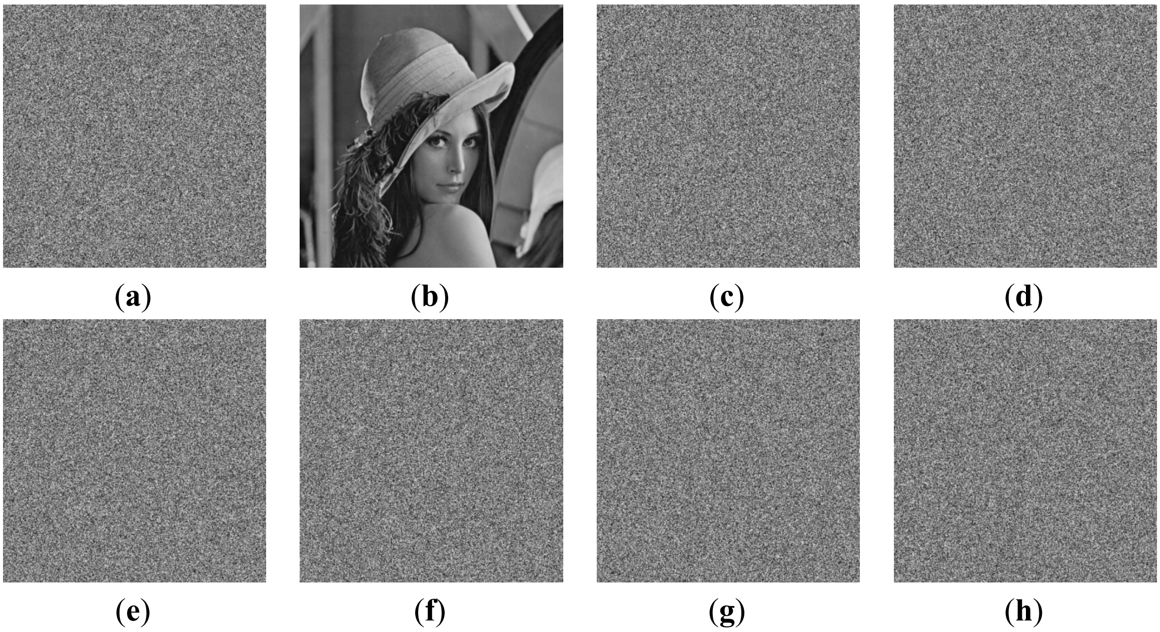
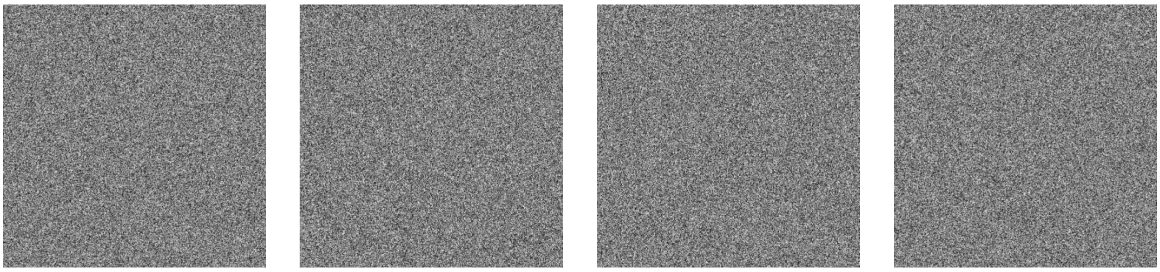


Figure 8. Cont.



(i) **(j)** **(k)** **(l)**

5.4. Plaintext Sensitivity Analysis

To implement plaintext sensitivity analysis, an opponent may try to establish a relationship between the plain image and its cipher image by observing the influence of a slight change on the overall encryption output. With the help of other analysis methods the secret key may be obtained. This kind of cryptanalysis becomes practically infeasible if such a slight change can be effectively diffused to the whole ciphered image. To measure the diffusion property of a cryptosystem, two criteria *NPCR* (number of pixel change rate) and *UACI* (unified average changing intensity) are commonly used.

The *NPCR* is used to measure the percentage of different pixel numbers between two images. Let $P_1(i, j)$ and $P_2(i, j)$ be the (i, j)th pixel of two images P_1 and P_2, respectively, the *NPCR* can be defined as:

$$NPCR = \frac{\sum_{i=1}^{W}\sum_{j=1}^{H} Diff(i,j)}{W \times H} \times 100\%, \tag{19}$$

where W and H are the width and height of P_1 or P_2, and $D(i, j)$ is set to 0 if $P_1\ (i, j) = P_2\ (i, j)$ and 1 otherwise. The *NPCR* value for two random images, namely the ideal value of the criterion, is given by:

$$NPCR_{\mathrm{exp}ected} = \left(1 - \frac{1}{2^D}\right) \times 100\%. \tag{20}$$

For instance, the expected *NPCR* for two random 8-bit grayscale images is 99.609%.

The second criterion, *UACI* is used to measure the average intensity of differences between the two images. It is defined as:

$$UACI = \frac{1}{W \times H}\left[\sum_{i=1}^{W}\sum_{j=1}^{H} \frac{|P_1(i,j) - P_2(i,j)|}{2^D - 1}\right] \times 100\%. \tag{21}$$

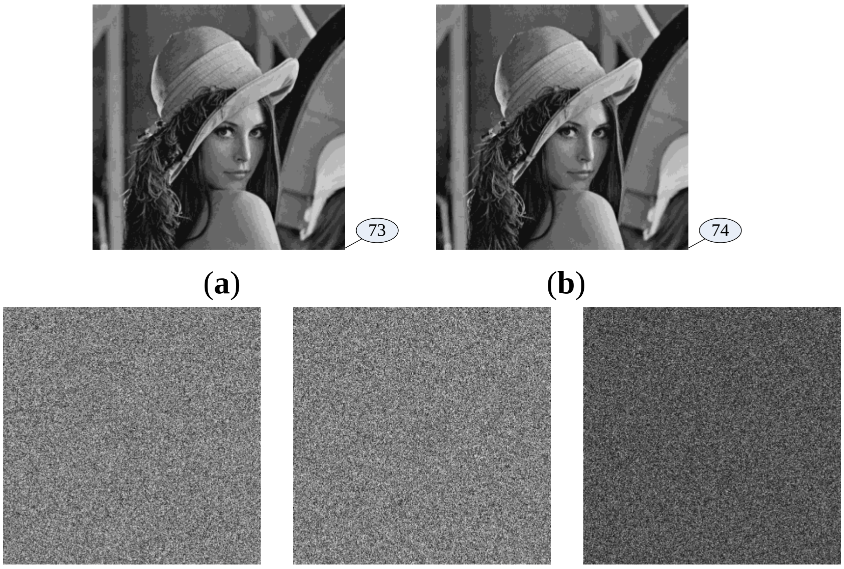
The *UACI* value for two random images is given by:

$$UACI_{\mathrm{exp}ected} = \frac{1}{2^{D^2}}\left(\frac{\sum_{i=1}^{2^D-1} i(i+1)}{2^D - 1}\right) \times 100\%. \tag{22}$$

For an 8-bit grayscale image, the expected *UACI* value is 33.464%.

To test the *NPCR* and *UACI* of the proposed cryptosystem, we assume a worst case that two plain images have only one bit difference at the lower-right pixel, as illustrated by Figures 9(a,b), respectively. The two images are encrypted with the same key and their corresponding cipher images are shown in Figures 9(c,d), respectively. The differential image between the two cipher images can be found in Figure 9(e). We obtain *NPCR* = 99.611% and *UACI* = 33.467%. The results show that a slight change in the original image will result in a significant change in the ciphered image, so the proposed scheme is robust against differential analysis.

Figure 9. *NPCR* and *UACI* test. (**a**) and (**b**) are two plain images with only one bit difference at the lower-right pixel. (**c**) cipher image of (**a**). (**d**) cipher image of (**b**). (**e**) differential image between (**c**) and (**d**).



(**a**) (**b**)

(**c**) (**d**) (**e**)

5.5. Speed Performance

Apart from the security considerations, computational efficiency is another important issue for a good cryptosystem, particularly for real-time Internet applications. Table 5 shows the time required for encrypting a 256 × 256 image with 256 grey levels by using the proposed and some typical block and chaos-based ciphers.

The number of permutation/diffusion rounds indicate the minimum number of iterations required to achieve a satisfactory diffusion effect, *i.e.*, *NPCR* > 0.996 and *UACI* > 0.334. As the operation mechanism of the chaos-based encryption algorithms is quite different from that of block algorithms, the comparison of iteration times is made only between chaos-based approaches. All the algorithms have been implemented using Code::Blocks and the tests have been done on a personal computer with an Intel Core i3-2100 CPU and 2 GB RAM. As we know, the speed performance of an algorithm may be influenced by many factors, including the compiler used and even the programming level. To estimate the efficiency of an image cryptosystem more precisely, a more objective criterion, number of basic assembly instructions needed to cipher a pixel, is employed, and the analysis results for the proposed and the comparative schemes are also listed in Table 5. As can be seen from Table 5, the proposed scheme outperforms other listed schemes with respect to either security or computational complexity. Therefore, our image cryptosystem is quite suitable for Internet applications over broadband networks, where the encryption and decryption time should be short relative to the transmission time.

Table 5. Comparison between the performance and security of the proposed and some typical block and chaos-based ciphers.

Approaches	Total cites	Permutation rounds	Diffusion rounds	Key size	Known/chosen-plaintext attack	Encryption time (*ms*)	Number of basic instructions per pixel
DES	N/A	N/A	N/A	56	Robust (CBC)	104.4	N/A
AES	N/A	N/A	N/A	128,192, 256	Robust (CBC)	75.5	N/A
Chen *et al* (2004) [4]	1024	4	4	128	Weak	54.7	2203
Wong *et al* (2008) [12]	176	2	2	256	Robust	52.3	1858
Patidar (2009) [14]	132	2	2	157	Weak	45.2	1629
Our scheme	N/A	1	2	241	Robust	30.7	1208

6. Conclusions

This paper has proposed an improved bit-level permutation approach for chaos-based image cipher with permutation-diffusion architecture. In the permutation stage, a significant diffusion effect is introduced through a 3D cat map-based spatial bit-level shuffling algorithm. As the pixel value mixing effect is contributed by both stages, the number of iteration rounds required by the time-consuming diffusion procedure is reduced, and hence the performance of the cryptosystem is improved. Compared with other recently proposed bit-level permutation algorithms, the diffusion effect of the proposed method is superior as the bits are shuffled among different bit-planes rather than within the same bit-plane. In the diffusion stage, the key stream elements extracted from the hyperchaotic system are circularly shifted under the control of plain pixel. As a result, the key stream is related to both the secret key and the plain image, which enhances the security against known/chosen plaintext attack. Moreover, compared with low dimensional chaotic maps, the hyperchatic system has more complicated dynamical property and number of state variables, which further enhance the security of the cryptosystem. Both theoretical analyses and experimental results indicate the new image cryptosystem has a high security level, which can effectively resist all common attacks such as brute force attacks, differential attacks, various statistical attacks, and known/chosen plaintext attacks. Therefore the proposed scheme has excellent potential for practical online image encryption applications.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61271350), and the Fundamental Research Funds for the Central Universities (No. N120504005).

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Fridrich, J. Symmetric ciphers based on two-dimensional chaotic maps. *Int. J. Bifurc. Chaos* **1998**, *8*, 1259–1284.
2. Scharinger, J. Fast encryption of image data using chaotic Kolmogorov flows. *J. Electron. Imaging* **1998**, *7*, 318–325.
3. Shannon, C.E. Communication theory of secrecy systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715.
4. Chen, G.R.; Mao, Y.B.; Chui, C.K. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos Solitons Fractals* **2004**, *21*, 749–761.
5. Pareek, N.K.; Patidar, V.; Sud, K.K. Image encryption using chaotic logistic map. *Image Vis. Comput.* **2006**, *24*, 926–934.
6. Kwok, H.S.; Tang, W.K.S. A fast image encryption system based on chaotic maps with finite precision representation. *Chaos Solitons Fractals* **2007**, *32*, 1518–1529.
7. Xiang, T.; Wong, K.W.; Liao, X.F. Selective image encryption using a spatiotemporal chaotic system. *Chaos* **2007**, *17*, 023115.
8. Behnia, S.; Akhshani, A.; Ahadpour, S.; Mahmodi H.; Akhavan A. A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps. *Phys. Lett. A* **2007**, *366*, 391–396.
9. Behnia, S.; Akhshani, A.; Mahmodi, H.; Akhavan A. A novel algorithm for image encryption based on mixture of chaotic maps. *Chaos Solitons Fractals* **2008**, *35*, 408–419.
10. Gao, T.G.; Chen, Z.Q. A new image encryption algorithm based on hyper-chaos. *Phys. Lett. A* **2008**, *372*, 394–400.
11. Tong, X.; Cui, M. Image encryption with compound chaotic sequence cipher shifting dynamically. *Image Vis. Comput.* **2008**, *26*, 843–850.
12. Wong, K.W.; Kwok, B.S.H.; Law, W.S. A fast image encryption scheme based on chaotic standard map. *Phys. Lett. A* **2008**, *372*, 2645–2652.
13. Tong, X.J.; Cui, M.G. Image encryption scheme based on 3D baker with dynamical compound chaotic sequence cipher generator. *Signal Process.* **2009**, *89*, 480–491.
14. Patidar, V.; Pareek, N.K.; Sud, K.K. A new substitution-diffusion based image cipher using chaotic standard and logistic maps. *Commun. Nonlinear Sci. Numer. Simul.* **2009**, *14*, 3056–3075.
15. Rhouma, R.; Meherzi, S.; Belghith, S. OCML-based colour image encryption. *Chaos Solitons Fractals* **2009**, *40*, 309–318.
16. Wang, Y.; Wong, K.W.; Liao, X.F.; Xiang T.; Chen G.R. A chaos-based image encryption algorithm with variable control parameters. *Chaos Solitons Fractals* **2009**, *41*, 1773–1783.
17. Wong, K.W.; Kwok, B.S.H.; Yuen, C.H. An efficient diffusion approach for chaos-based image encryption. *Chaos Solitons Fractals* **2009**, *41*, 2652–2663.
18. Mazloom, S.; Eftekhari-Moghadam, A.M. Color image encryption based on Coupled Nonlinear Chaotic Map. *Chaos Solitons Fractals* **2009**, *42*, 1745–1754.
19. Elashry, I.F.; Allah, O.S.F.; Abbas, A.M.; El-Rabaie S.; El-Samie F.E.A. Homomorphic image encryption. *J. Electron. Imaging* **2009**, *18*, 033002.
20. Borujeni, S.E.; Eshghi, M. Chaotic image encryption design using tompkins-paige algorithm. *Math. Probl. Eng.* **2009**, *2009*, 762652.

21. Amin, M.; Faragallah, O.S.; Abd El-Latif, A.A. A chaotic block cipher algorithm for image cryptosystems. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 3484–3497.
22. Zhu, C.X. A novel image encryption scheme based on improved hyperchaotic sequences. *Opt. Commun.* **2012**, *285*, 29–37.
23. Fu, C.; Chen, J.J.; Zou, H.; Meng W.H.; Zhan Y.F.; Yu Y.W. A chaos-based digital image encryption scheme with an improved diffusion strategy. *Opt. Express* **2012**, *20*, 2363–2378.
24. Seyedzadeh, S.M.; Mirzakuchaki, S. A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. *Signal Process.* **2012**, *92*, 1202–1215.
25. Fu, C.; Meng, W.H.; Zhan, Y.F.; Zhu Z.L.; Lau F.C.M.; Tse C.K.; Ma H.F. An efficient and secure medical image protection scheme based on chaotic maps. *Comput. Biol. Med.* **2013**, *43*, 1000–1010.
26. Fu, C.; Lin, B.B.; Miao, Y.S.; Liu X.; Chen, J.J. A novel chaos-based bit-level permutation scheme for digital image encryption. *Opt. Commun.* **2011**, *284*, 5415–5423.
27. Zhu, Z.L.; Zhang, W.; Wong, K.W.; Yu H. A chaos-based symmetric image encryption scheme using a bit-level permutation. *Inf. Sci.* **2011**, *181*, 1171–1186.
28. Liu, H.J.; Wang, X.Y. Color image encryption using spatial bit-level permutation and high-dimension chaotic system. *Opt. Commun.* **2011**, *284*, 3895–3903.
29. Zhang, G.J.; Shen, Y. A novel bit-level image encryption method based on chaotic map and dynamic grouping. *Commun. Theor. Phys.* **2012**, *58*, 520–524.
30. Solak, E.; Cokal, C.; Yildiz, O.T.; Biyikoglu, T. Cryptanalysis of Fridrich's chaotic image encryption. *Int. J. Bifurc. Chaos* **2010**, *20*, 1405–1413.
31. Wang, K.; Pei, W.J.; Zou, L.H.; Song A.G.; He Z. On the security of 3D Cat map based symmetric image encryption scheme. *Phys. Lett. A* **2005**, *343*, 432–439.
32. Li, C.Q.; Li, S.J.; Asim, M.; Nunez J; Alvarez G; Chen G.R. On the security defects of an image encryption scheme. *Image Vis. Comput.* **2009**, *27*, 1371–1381.
33. Rhouma, R.; Belghith, S. Cryptanalysis of a new image encryption algorithm based on hyper-chaos. *Phys. Lett. A* **2008**, *372*, 5973–5978.
34. Li, C.Q.; Li, S.J.; Chen, G.R.; Halang W.A. Cryptanalysis of an image encryption scheme based on a compound chaotic sequence. *Image Vis. Comput.* **2009**, *27*, 1035–1039.
35. Rhouma, R.; Solak, E.; Belghith, S. Cryptanalysis of a new substitution-diffusion based image cipher. *Commun. Nonlinear Sci. Numer. Simul.* **2010**, *15*, 1887–1892.
36. Ozkaynak, F.; Ozer, A.B.; Yavuz, S. Cryptanalysis of a novel image encryption scheme based on improved hyperchaotic sequences. *Opt. Commun.* **2012**, *285*, 4946–4948.
37. Li, C.Q.; Liu, Y.S.; Xie, T.; Chen M.Z.Q. Breaking a novel image encryption scheme based on improved hyperchaotic sequences. *Nonlinear Dyn.* **2013**, *73*, 2083–2089.
38. Jia, Q. Hyperchaos generated from the Lorenz chaotic system and its control. *Phys. Lett. A* **2007**, *366*, 217–222.