ATLANTIS PRESS International Journal of Networked and Distributed Computing, Vol. 5, No. 1 (January 2017) 37–44

A New Chaos-based Image Cipher Using a Hash Function

Chong Fu [1, *], Ou Bian [2], Hui-yan Jiang [3], Li-hui Ge [4], Hong-feng Ma [5]

[1] *School of Computer Science and Engineering, Northeastern University*
NO. 3-11, Wenhua Road, Heping District
Shenyang, 110004, China
E-mail: fuchong@mail.neu.edu.cn

[2] *The General Hospital of Shenyang Military Command*
NO. 83, Wenhua Road, Shenhe District
Shenyang, 110016, China
E-mail: bianou_nh@hotmail.com

[3] *Software College, Northeastern University*
Key Laboratory of Medical Image Computing of Ministry of Education
NO. 3-11, Wenhua Road, Heping District
Shenyang, 110004, China
E-mail: hyjiang@mail.neu.edu.cn

[4] *Anshan Central Hospital*
NO. 77, Nanzhonghua Road, Tiedong District
Anshan, 114001, China
E-mail: gelihui_ln@163.com

[5] *TeraRecon*
4000 East 3rd Avenue, Suite200
Foster, CA 94404, United States
E-mail: wind_ma621@hotmail.com

Abstract

This paper presents a new chaos-based image cipher using a plaintext-related permutation. The cat map and Lorenz system are employed to shuffle the positions of image pixels and generate the diffusion keystream, respectively. The control parameters of the cat map, i.e. the permutation key, are determined by the Murmur2 hash value of the original image. Owing to the avalanche property of hash functions, completely different shuffled images will be produced even if there is a tiny difference between the original ones, and it helps accelerate the diffusion process. Experimental results indicate that the proposed scheme requires only one and two cipher cycles to achieve an acceptable and a satisfactory diffusion properties, respectively, whereas two and three cipher cycles are needed by typical schemes to achieve the same properties. Thorough security analysis is carried out, and the results demonstrate the satisfactory security of the proposed scheme.

Keywords: image cipher; cat map; Murmur hash; Lorenz system

1. Introduction

Image encryption technologies have drawn much attention over the past decade or so to meet the increasing demand for secure image communications over public networks. Among the reported approaches, the chaos-based ones suggest a promising direction owing to their optimal trade-off between security and

* Corresponding author.

efficiency [1-2]. The iterative permutation-diffusion operations, suggested by Shannon for secure ciphers constructions, are widely adopted in chaos-based image ciphers. In each round of the cipher, the original image is firstly shuffled in a secret way, and then the pixel values are altered sequentially and the influence of each pixel is diffused to all its subsequent ones during the modification process. Generally, a satisfactory diffusion property can be achieved with 3-4 overall rounds of permutation and diffusion. Unlike modern block ciphers, such as Triple-DES, AES, whose keystreams are produced by key schedules or key expansion algorithms that work on integers, the chaos-based schemes produce keystreams by iterating chaotic systems/maps and quantifying their current values of state variables. As chaotic systems/maps are defined on real number field and arithmetic operations are significantly slower with floating-point types than with integral types, the computational cost of keystream generation is the main factor that affects the efficiency of a chaos-based cryptosystem. Consequently, a direct way to improve the efficiency of chaos-based image ciphers is reducing the number of iterations required by keystream generation, for instance, reducing the overall cipher cycles or the required keystream length. Several approaches have been proposed in accordance with this idea recently, and a brief overview is given below.

In [3], the feasibility of selective image encryption on a bitplane is investigated. It is concluded that only selectively encrypting 50% of the whole image data can gain an acceptable security. Therefore, the encryption time is substantially reduced. In [4-7], schemes with certain diffusion effect introduced in the permutation stage are proposed. As the pixel value mixing effect is contributed by both stages, the number of iteration rounds required by the diffusion procedure is reduced, and hence the performance of the cryptosystem is improved. In [8], Wong et al. proposed a more efficient diffusion mechanism using simple table lookup and swapping techniques as a light-weight replacement of the 1D chaotic map iteration. In [9], Wang et al. proposed a fast image encryption algorithm that combines the permutation and diffusion stages so that the pixel values are changed while the image blocks are being relocated. As a result, the image needs to be scanned only once in each encryption round, while conventional schemes separate the permutation and diffusion stages therefore require at least two image-scanning processes. In [10], a fast image cipher using a novel bidirectional diffusion is proposed. Theoretical analysis and simulation results indicate that a satisfactory level of security can be achieved with only one round of permutation and two rounds of diffusion operations.

A new chaos-based image cipher using cat map and Lorenz system is suggested in this paper. The control parameters of the cat map are determined by the Murmur2 hash value of the original image. Owing to the avalanche property of hash functions, completely different shuffled images will be produced even if there is a tiny difference between the original ones. Consequently, an acceptable diffusion property can be achieved with only one overall round and a satisfactory one with two overall rounds. The rest of this paper is organized as follows. Section 2 discusses the permutation and diffusion strategies of the proposed scheme, and their effectiveness and performance are evaluated in Section 3. In Section 4, the security of the proposed scheme is thoroughly analyzed. Finally, Section 5 concludes the paper.

2. The Proposed Scheme

The architecture of the proposed scheme is illustrated by Fig. 1. Under this structure, the original image is firstly shuffled by using Arnold cat map, whose control parameters, i.e. the permutation key, are given by the hash value of the original image. As is known, the essential property of a hash function is that it almost surely produces different hash values for different messages. This means different images are rearranged in different ways and it helps enhance the diffusion effect. In our scheme, the 64-bit version Murmurhash2 algorithm, created by Austin Appleby in 2008, is employed. The algorithm outperforms most other ones because of its excellent distribution, avalanche behavior, collision resistance and performance. In the diffusion stage, the shuffled data are masked by a keystream extracted from the orbit of Lorenz system. A large key space is ensured as the three state variables of the Lorenz system are used as the diffusion key. The detailed permutation and diffusion operations are discussed as follows.

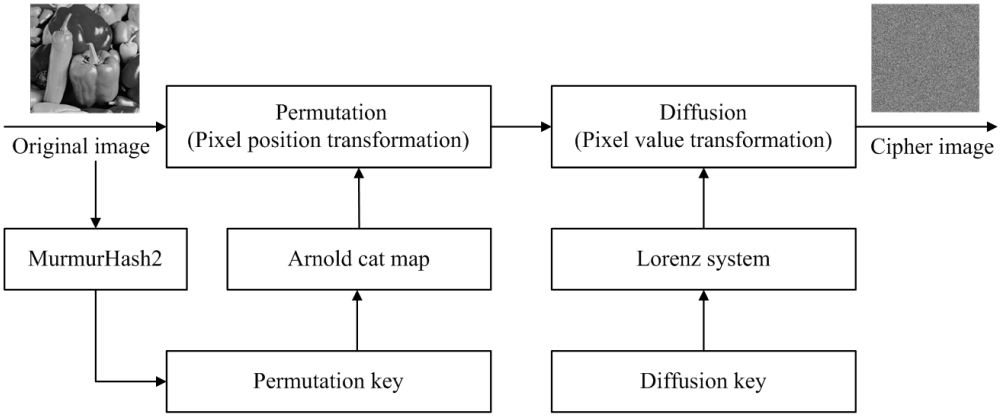


Fig. 1. Architecture of the proposed scheme.

2.1. Permutation process

The Arnold cat map, described by Eq. (1), is a chaotic bijection of a unit square onto itself.

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod 1, \tag{1}$$

where p and q are control parameters, and x mod 1 means the fractional part of x for any real number x. To incorporate the map into image permutation that operated on a lattice of finite number of pixels, it has to be discretized. This can be done simply by changing the range of (x, y) from the unit square to the lattice $N \times N$, as follows:

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & p \\ q & pq+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod N, \tag{2}$$

where N is the number of pixels in one row (column). The inverse transform of the map is easily found to be given by

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} pq+1 & -p \\ -q & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod N. \tag{3}$$

To determine the value of p and q, the 64-bit Murmur2 hash value of the original image is firstly divided into two 32-bit parts, which are denoted by $hash_l$ and $hash_r$, respectively. As the four-tuple $[1, (p+k_1N), (q+k_2N), (p+k_3N)(q+k_4N)+1]$ produce the same output as the four-tuple $[1, p, q, (pq+1)]$ for any $k_1, k_2, k_3, k_4 \in Z$, the two parameters are given by $mod(hash_l, N)$ and $mod(hash_r, N)$, respectively. The utilization of small parameters also speeds up the calculation. The pseudocode of the MrumurHash2 algorithm is listed below, where the argument *key* is a pointer that points to the image data.

pseudocode of MrumurHash2 algorithm

```
Murmur2_64(key, len, seed)
    m ← 0xc6a4a7935bd1e995;
    r ← 47
    hash ← seed ^ (len* m)
    for each eightByteChunk of key
        k ← eightByteChunk
        k ← k × m
        k ← k XOR (k >> r)
        k ← k × m
        hash ← hash XOR k
        hash ← hash × m
    end for
    with any remainingBytesInKey
        remainingBytes ←
SwapEndianOrderOf(remainingBytesInKey)
        // Note: Endian swapping is only
necessary on big-endian machines.
        hash ← hash XOR remainingBytes
        hash ← hash × m
    end with
    hash ← hash XOR (hash >> r)
    hash ← hash × m
    hash ← hash XOR (hash >> r)
```

As can be seen from the pseudocode, the algorithm uses bitwise and integer multiplication operations to manipulate and update the hash value. As is known, the two operations are very efficient for hardware implementation, and thereby the computation cost of the hash algorithm is much lower than that of one round of

diffusion operation, where the manipulations of real numbers are required.

2.2. Diffusion process

The well-known Lorenz system, developed by Edward Lorenz in 1963 for atmospheric convection, is described by

$$\begin{cases} \dot{x} = \sigma(y - x), \\ \dot{y} = x(\rho - z) - y, \\ \dot{z} = xy - \beta z, \end{cases} \tag{4}$$

where t is time and σ, ρ, β are the system parameters. The system is chaotic for the values of σ=10, ρ=8/3, β=28. The initial values of the state variables, (x_0, y_0, z_0), are used as the diffusion key.

The detailed diffusion process is described as follows:

Step 1: Arrange the pixels of the shuffled image to a vector $p=\{p_0, p_1, \ldots, p_{N\times N-1}\}$ in the order from left to right, top to bottom.

Step 2: Generate a keystream with length equal to p.

Step 2.1: Pre-iterate system (4) for I_0 times to the harmful effect of transitional procedure, where I_0 is a constant. The fourth-order Runge-Kutta method is employed for solving the equation [11].

Step 2.2: Iterate system (4) for t times, where t = ceil($N\times N$/3). For each iteration, the current values of the three state variables are appended to a vector $L_s=\{s_0, s_1, \ldots, s_{N\times N-1}\}$. Obviously, there are r = ($t\times 3 - N\times N$) redundant elements, which should be discarded.

Step 2.3: Qualify a keystream $k=\{k_0, k_1, \ldots, k_{N\times N-1}\}$ from L_s according to

$$k_n = mod[sig\_n((abs(s_n)), 2^L], \tag{5}$$

where L is the color depth of the original image, $abs(x)$ returns the absolute value of x, and $sig\_n(x)$ returns the n most significant decimal digits of x, where n is the precision of x. In our scheme, all the variables are declared as double-precision type, which has a precision of 15 decimal digits.

Step 3: Calculate the cipher-pixels value according to Eq. (6).

$$c_n = k_n \oplus \{[p_n + k_n] \ mod \ 2^L\} \oplus c_{n-1}, \tag{6}$$

where c_n and c_{n-1} are the output and previous cipher-pixels, respectively, and $\oplus$ performs bit-wise exclusive OR operation. One may set the initial value c_{-1} as a constant.

The decipher procedure is the same as that of the encipher process described above except that the inverse of Eq. (6), described by Eq. (7), is employed.

$$p_n = \left[k_n \oplus c_n \oplus c_{n-1} + 2^L - k_n \right] mod \ 2^L. \tag{7}$$

3. Analysis of Permutation and Diffusion Properties

3.1. Analysis of permutation property

To evaluate the effectiveness of the proposed permutation method, the following procedure is carried out. First, create an image with only 1-bit difference to the original one. The position of the differential pixel and the +1/–1 operation are randomly chosen. Next, generate the control parameters for the cat map according to the method discussed in Sec. 2.1. Finally, the two images are shuffled using their own parameters, and the differences between the resultant images are calculated. Table 1 lists the experimental results on five typical test images of size 512×512. Fig. 2 demonstrates the application of the proposed permutation strategy to the peppers test image and its modification. Figs. 2(a) and (b) show the peppers test image and its modification by a 1-bit change, respectively. Their corresponding shuffled images are shown in Figs. 2(c) and (d), respectively. Fig. 2(e) shows the differential images between the two shuffled images. As can be seen from Table 1 and Fig. 2, completely different outputs are produced even if there is only 1-bit difference between the two images.

3.2. Analysis of diffusion property

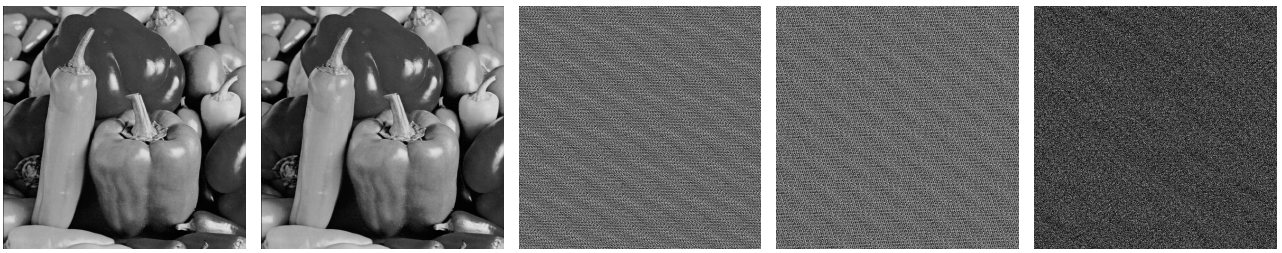
Diffusion serves to spread the influence of a single plaintext bit over as much of the ciphertext as possible. This is important because otherwise the cryptosystem will be vulnerable to chosen-plaintext attack. The diffusion property of an image cryptosystem is commonly measured by means of two criteria, namely, *NPCR* (the number of pixel change rate) and *UACI* (the unified average changing intensity). 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* and *UACI* are defined as:

$$NPCR = \frac{\sum_{i=1}^{W}\sum_{j=1}^{H} D(i,j)}{W \times H} \times 100\% \tag{8}$$

Table 1. Differences between shuffled images produced from original images with only 1-bit difference

Test image name	Position (x, y)	Pixel value		Control parameters (p, q)		Diff.*
		original	*modified*	*original*	*modified*	
avion	(334, 189)	199	198	(467, 231)	(174, 412)	98.56%
baboon	(400, 65)	131	132	(268, 359)	(192, 51)	99.31%
house	(232, 353)	112	111	(332, 91)	(37, 280)	99.10%
Lena	(144, 94)	46	45	(227, 362)	(248, 297)	99.36%
peppers	(368, 509)	161	162	(365, 218)	(115, 291)	99.42%

*Diff.: Differences between two shuffled images



(a) (b) (c) (d) (e)

Fig. 2. The application of the proposed permutation algorithm. (a) the original peppers test image; (b) the image with 1-bit difference from (a); (c) the shuffled image corresponding to (a); (d) the shuffled image corresponding to (b); (e) the differential image between (c) and (d).

Table 2. Results of *NPCR* and *UACI* tests

Test image name	No. of cipher rounds (proposed scheme)				No. of cipher rounds (typical schemes)					
	1		2		1		2		3	
	NPCR	*UACI*	*NPCR*	*UACI*	*NPCR*	*UACI*	*NPCR*	*UACI*	*NPCR*	*UACI*
avion	0.9960	0.3338	0.9961	0.3342	0.6909	0.0027	0.9956	0.3319	0.9960	0.3344
baboon	0.9961	0.3346	0.9960	0.3341	0.8347	0.0131	0.9954	0.3306	0.9961	0.3351
house	0.9961	0.3338	0.9961	0.3346	0.3904	0.0031	0.9964	0.3318	0.9962	0.3355
Lena	0.9960	0.3340	0.9961	0.3353	0.6437	0.0025	0.9962	0.3369	0.9960	0.3342
peppers	0.9960	0.3339	0.9962	0.3348	0.8182	0.0032	0.9960	0.3386	0.9962	0.3345

and

$$UACI = \frac{1}{W \times H}\left[\sum_{i=1}^{W}\sum_{j=1}^{H}\frac{|P_1(i,j) - P_2(i,j)|}{L-1}\right] \times 100\%, \quad (9)$$

respectively, where W and H are the width and height of P_1 or P_2 and $D(i, j)$ is defined as

$$D(i,j) = \begin{cases} 0 & \text{if } P_1(i,j) = P_2(i,j), \\ 1 & \text{if } P_1(i,j) \neq P_2(i,j). \end{cases} \quad (10)$$

The *NPCR* and *UACI* values for two truly random images, which are expected estimate for a good image cryptosystem, are 99.609% and 33.464%, respectively. To evaluate the *NPCR* and *UACI* of a cryptosystem under worst case conditions, two test images with only 1-bit difference at the last, lower-right pixel are usually constructed and encrypted with the same key. We test five such pairs of images, and the results are compared with that of typical schemes, as listed in Table 2. As can be seen from Table 2, the proposed scheme requires only one and two cipher cycles to achieve an acceptable and a

satisfactory diffusion properties, respectively, whereas two and three cipher cycles are needed by typical schemes to achieve the same properties. Therefore, the proposed scheme has a superior computational efficiency.

4. Security Analysis

4.1. Key space analysis

For an effective cryptosystem, the key space should be large enough to make brute-force attack infeasible. The key of the proposed cryptosystem is composed of two parts: the permutation key contributed by the control parameters (p, q) of the cat map and the diffusion key contributed by the initial conditions (x_0, y_0, z_0) of the Lorenz system. As (p, q) are integers range from 1 to N, the space of the permutation key is N^2. As mentioned above, the state variables of the Lorenz system are declared as double-precision type, and therefore the space of the diffusion key is approximately $(10^{15})^3$. The two parts of the key are independent of each other. Therefore, the total key space of the proposed cryptosystem is $N^2\times10^{45}$. We take images of size 512×512 as example, the total size satisfies $512^2\times10^{45} \approx 2^{185}$. Generally, cryptosystems with a key space greater than 2^{100} are considered to be "computational security", and therefore the proposed scheme is robust against brute-force attack.

4.2. Frequency distribution of pixel values

The frequency distribution of cipher-pixel values is of much importance to an image cryptosystem. It should hide the redundancy of plain image and should not reveal the relationship between plain image and cipher image. Histogram analysis is often used as a qualitative check for data distribution. An image histogram is a graphical representation showing a visual impression of the distribution of pixels by plotting the number of pixels at each grayscale level. The histograms of the peppers test image (Fig. 3(a)) and its output cipher image (Fig. 3(c)) are shown Figs. 3(b) and (d), respectively. It can be seen from Fig. 3(d) that the pixel values of the output cipher image are perfectly uniformly distributed over the whole intensity range.

The distribution of pixel values can be further qualitatively determined by calculating the information entropy of the image. The information entropy is usually expressed by the average number of bits needed to store or communicate one symbol in a message, as described by

$$H(X)=-\sum_{i=1}^{N}P(x_i)\log_2 P(x_i), \tag{11}$$

where X is a random variable with N outcomes $\{x_1, ..., x_N\}$ and $P(x_i)$ is the probability mass function of outcome x_i. It's obvious from Eq. (11) that the entropy for a random source emitting N symbols is $H(X)=\log_2 N$. For instance, for a ciphered image with 256 gray levels, the entropy should ideally be $H(X)=8$, otherwise there exists certain degree of predictability which threatens its security. The information entropy of the five test images and their corresponding cipher images produced by the proposed scheme are calculated, and the results are listed in Table 3. As can been seen from Table 3, the entropy of all the output cipher images are very close to the theoretical value of 8. This means the proposed scheme produces outputs with perfect randomness and hence is robust against frequency analysis.

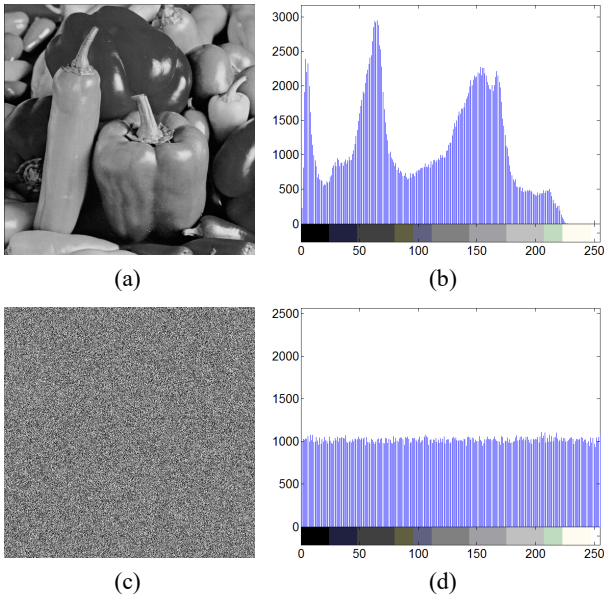


Fig. 3. Histograms of the peppers test image and its output cipher image. (a) the peppers test image; (b) histogram of (a); (c) cipher image corresponding to (a); (d) histogram of (c).

Table 3. Results of information entropy analysis

Test image name	Avion	Baboon	House	Lena	Peppers
Plain	6.6777	7.3579	7.2416	7.4456	7.5715
Cipher	7.9993	7.9993	7.9992	7.9994	7.9994

4.3. Correlation of adjacent pixels

For an ordinary image having definite visual content, each pixel is usually highly correlated with its adjacent pixels either in horizontal, vertical or diagonal direction. However, an efficient image cryptosystem should produce the cipher image with sufficiently low correlation in the adjacent pixels. Figs. 4(a) and (b) show the correlation distribution of two horizontally adjacent pixels in the peppers test image (Fig. 3(a)) and its output cipher image (Fig. 3(c)), respectively. Similar results can be obtained for vertically and diagonally adjacent pixels. As can be seen from Fig. 4(a), most points are clustered around the main diagonal, whereas those in Fig. 4(b) are fairly evenly distributed. The results indicate that the proposed scheme can effectively eliminate the correlation between adjacent pixels in an original image.

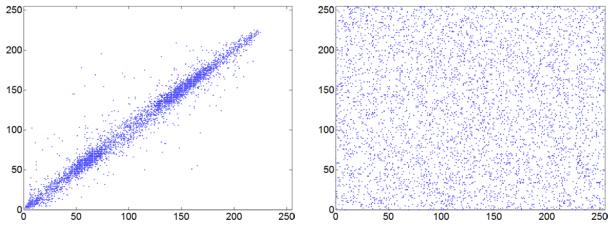


Fig. 4. (a) correlations of horizontally adjacent pixels in Fig. 3(a); (b) correlations of horizontally adjacent pixels in Fig. 3(c).

To further quantify and compare the correlations of adjacent pixels in the test image and its output cipher image, the following procedure is carried out. First, randomly select 5000 pairs of adjacent pixels in horizontal, vertical and diagonal direction from the shuffled image, respectively. Then, calculate the correlation coefficient $r_{x,y}$ of each pair by using the following three formulas:

$$r_{xy} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2\right)\left(\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2\right)}}, \tag{12}$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{13}$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i, \tag{14}$$

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

Table 4 lists the calculated correlation coefficients for adjacent pixels in five test images and their output cipher images. As can be seen from Table 4, the correlation coefficients for adjacent pixels in the output cipher images are very close to zero, and it further proves the conclusion drawn from Fig. 4.

Table 4. Correlation coefficients for adjacent pixels in five test images and their output cipher images

Test image name	Original			Ciphered		
	Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
avion	0.9631	0.9665	0.9397	–0.0197	–0.0148	–0.0143
baboon	0.7485	0.8674	0.7262	–0.0151	0.0303	–0.0010
house	0.9615	0.9480	0.9152	–0.0219	0.0026	–0.0165
Lena	0.9852	0.9720	0.9599	0.0010	0.0104	0.0087
peppers	0.9811	0.9766	0.9637	–0.0034	–0.0248	0.0037

5. Conclusions

This paper has proposed a fast chaos-based image cipher with a permutation-diffusion structure. The cat map and Lorenz system are employed to transform the pixel positions and generate the diffusion keystream, respectively. In the permutation stage, the Murmur2 hash value of the original image is calculated to determine the control parameters of the cat map. Owing to the avalanche property of hash functions, completely different shuffled images will be produced even if there is a tiny difference between the original ones, and it helps accelerate the diffusion process. Experimental results indicate that the proposed scheme requires only one and two cipher cycles to achieve an acceptable and a satisfactory level of security, respectively. Compared with typical schemes that require two and three cipher cycles to achieve the same level of security, the proposed scheme has a superior computational efficiency. Extensive security analysis has been carried out,

including the most important ones like key space analysis, statistical analysis, and plaintext sensitivity analysis, which have demonstrated the satisfactory security of the new scheme.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61271350), the Fundamental Research Funds for the Central Universities (No. N150402004), and the Online Education Research Fund of MOE Research Center for Online Education (Qtone Education) (No. 2016YB123).

References

1. G. Chen, Y. Mao, and C. K. Chui, A symmetric image encryption scheme based on 3D chaotic cat maps, *Chaos, Solitons & Fractals*. **21**(3) (2004) 749–761.
2. N. K. Pareek, V. Patidar, and K. K. Sud, Image encryption using chaotic logistic map, *Image and Vision Computing*. **24**(9) (2006) 926–934.
3. T. Xiang, K. W. Wong, and X. Liao, Selective image encryption using a spatiotemporal chaotic system, *Chaos: An Interdisciplinary Journal of Nonlinear Science.* **17**(2) (2007) article no. 023115.
4. K. W. Wong, B. S. H. Kwok, and W. S. Law, A fast image encryption scheme based on chaotic standard map, *Physics Letters A*. **372**(15) (2008) 2645–2652.
5. C. Fu, B. B. Lin, Y. S. Miao, X. Liu, and J. J. Chen, A novel chaos-based bit-level permutation scheme for digital image encryption, *Optics Communications.* **284**(23) (2011) 5415–5423.
6. C. Fu, W. H. Meng, Y. F. Zhan, Z. L. Zhu, F. C. Lau, K. T. Chi, and H. F. Ma, An efficient and secure medical image protection scheme based on chaotic maps, *Computers in biology and medicine.* **43**(8) (2013) 1000–1010.
7. C. Fu, J. B. Huang, N. N. Wang, Q. B. Hou, and W. M. Lei, A symmetric chaos-based image cipher with an improved bit-level permutation strategy, *Entropy*. **16**(2) (2014) 770–788.
8. K. W. Wong, B. S. H. Kwok, and C. H. Yuen, An efficient diffusion approach for chaos-based image encryption, *Chaos, Solitons & Fractals*. **41**(5) (2009) 2652–2663.
9. Y. Wang, K. W. Wong, X. Liao, and G. Chen, A new chaos-based fast image encryption algorithm, *Applied soft computing*. **11**(1) (2011) 514–522.
10. C. Fu, J. J. Chen, H. Zou, W. H. Meng, Y. F. Zhan, and Y. W. Yu, A chaos-based digital image encryption scheme with an improved diffusion strategy, *Optics Express*. **20**(3) (2012) 2363–2378.
11. C. Fu, G. Y. Zhang, O. Bian, W. M. Lei, and H. F. Ma, A novel medical image protection scheme using a 3-dimensional chaotic system, *PloS one*. **9**(12) (2014), article no. e115773.