Chaos-Based Image Encryption Algorithm Using Decomposition

Hongyao Deng*[1,2], Qingxin Zhu[1,] Xiuli Song[3], Jingsong Tao[4]
[1]School of Information & Software Engineering, University of Electronic Science and Technology of China, No. 2006, Xiyuan Ave., West Hi-Tech Zone, Chengdu, 611731, Sichuan China, Ph/Fax: +86 28 83201864
[2]School of Mathematics & Computer Science, Yangtze Normal University, No. 98 Julong Rd., Lidu Fuling District, Chongqing, 408000, China. Ph: +86 23 07290088
[3]Dept. of Computer Science and Technology, Chongqing University of Posts and Telecommunications, No. 2 Chongwen Rd., Nan-an District, Chongqing, 400065, China, Ph: +86 23 62461404
[4]School of Electrical Engineering, Wuhan University, Luojiashan,Wuchang, Wuhan,430072, China, Ph: +86 27 68770776
Corresponding author, e-mail: hydeng_2004@163.com[1,2], qxzhu@uestc.edu.cn[1], songxl@cqupt.edu.cn[3], jamson_tao@163.com[4]

Abstract

In this paper, a chaos-based image encryption algorithm was proposed. It consists of four stages: decomposition, shuffle, diffusion and combination. Decomposition is that an original image is decomposed to components by some rule. Shuffle and diffusion are the essential processions of image encryption. The purpose of the shuffle is to mask original organization of the pixels in images and diffusion is to mask their values. Combination is not necessary for real-time Internet applications. However, the components must be labeled before sended, and the assemblage of components would be done on the receiving terminal. There were two methods to improve security. One was to enlarge the key space, and another was to strong the nonlinear map. About the speed, the scheme had good characteristics of parallel. More than 50 images were tested to evaluate the algorithm. Experiment results and security analysis demonstrate that the encryption algorithm not only is robust and flexible, but also can withstand common attacks such as statistical attacks and differential attacks.

Keywords: *Chaos encryption, Component images, Feature values, Decomposition*

Copyright © 2014 Institute of Advanced Engineering and Science. All rights reserved.

1. Introduction

Digital image encryption is one of the technologies in secure image transmission over the Internet and wireless networks with the rapid development of network. Compared with traditional encryption for textual data, the approach to encrypting images is different due to its intrinsic data features of images. Generally, there are two types of cryptographic scheme, namely private-key and public-key schemes. Classical approaches such as DES, AES are classified as private-key cryptographic scheme, while RSA and elliptic curve cryptography (ECC) are classified as public-key cryptographic scheme. For chaos-based image encryption, its cryptography with chaos falls into the category of private-key scheme. The secret keys are usually the chaos system parameters and the initial conditions. Therefore, sensitivity to parameters and initial conditions is one of characteristic for a chaos-based cryptosystem.

So far, many image encryption schemes have been proposed. It can be divided to two kinds, namely spatial domain and frequency domain image encryptions. For the former, generally the main methods are to mix the position of pixels by choosing a map such as Baker, Arnold or Standard map. In [1], Mao et al. proposed a chaos-based image encryption scheme by Baker maps. In [2], Wong et al. introduced a certain diffusion effect in the substitution stage by simple sequential add-and-shift operations, using these operations to reduce the overall encryption time as fewer rounds are required. Gao et al. [3] presented a nonlinear chaotic algorithm (NCA) for image encryption, which used power function and tangent function instead of linear parts in the Logistic map. Chen et al. [4] designed a symmetric encryption scheme. The scheme employed a 3D cat map to shuffle image pixels and used Logistic map to confuse the relationship between the plain-image and cipher-image. In [5], a chaotic cryptographic scheme iterating a Logistic map was proposed, and the look-up table used in the cryptographic process

Received April 27, 2013; Revised July 12, 2013; Accepted August 6, 2013

was updated dynamically. In [6], Krikor et al. presented a method of image encryption by selecting specific higher frequencies of DCT coefficients taking as the characteristic values. In [7], the original image first was dealt with DWT, then was divided into two parts. Encryption for approximate part and Compression for detailed part, both of them final form the cipher-image. For the methods in frequency domain, the decrypted image is not equal to the original image absolutely. The decrypted image contains small distortion which is acceptable due to human perception. Besides the two approaches to encryption image, some researcher uses multiple technologies to encrypt images. For example, in [8], Patel et al. encrypted image using different techniques. There are many methods for image decomposition such as bit-plane decomposition, color-based decomposition. Literature [9, 10] proposed bit-plane decomposition based encryption approaches. In this paper, we propose chaos-based image encryption algorithm using decomposition, the encryption algorithm is simplified using decomposition under ensuring the security.

2. Our Image Encryption Algorithm

The architecture of our image encryption algorithm is shown in Figure 1. It consists of four processing stages: decomposition, shuffle, diffusion and combination. The decoposition decomposes the original image, namely plain-image, to sub-image, namely component image or components. For each component, the shuffle displaces the positions of the pixels and the diffusion diffuses the values of the pixels. We call them position and values mask respectively. The combination combines all the components to final encrypted image.

As a specific instance, we gived a encryption scheme for RGB images, as can been seen in Figure 2. A RGB image is decomposed to red, green and blue component according to the three channels. Each component is processed in parallel during the processing of the shuffle and diffusion stages, witch improved encryption speed. The shuffle masks the original organization of the pixels and the diffusion masks the original values of the pixels. Shuffle and diffusion are crucial to the security of whole encryption algorithm. The decomposition can be taken as the preparation of encryption processing and the combination generates an encrypted image with a permutation order.

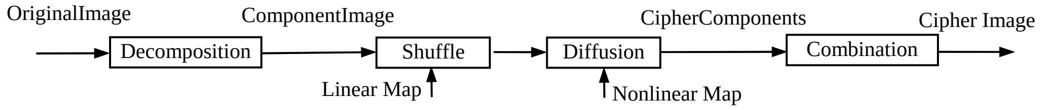

Figure 1. Architecture of image encryption algorithm

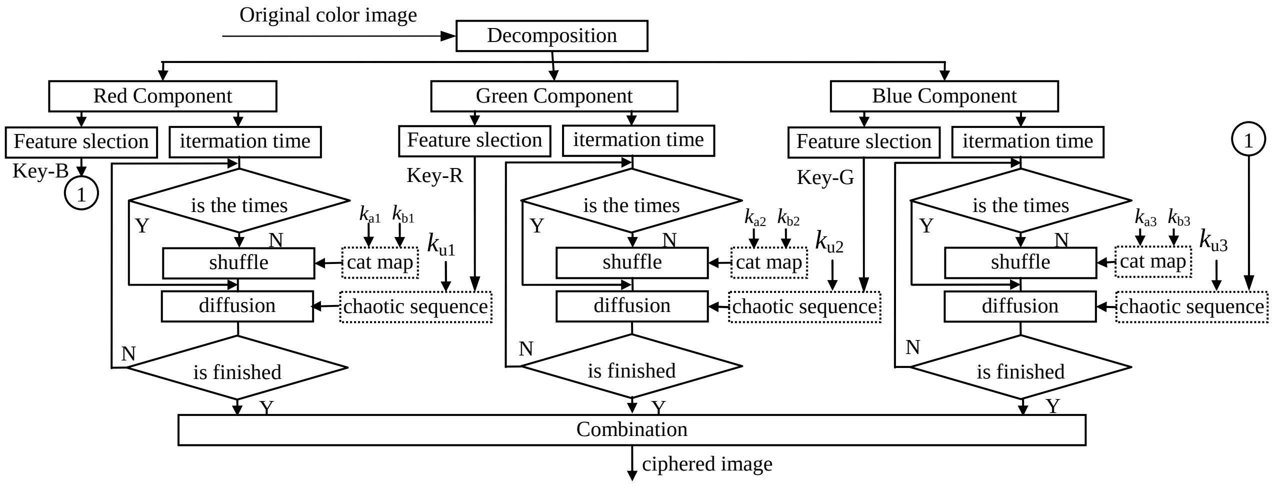

Figure 2. Encryption details of a color image

2.1. Decomposition

Any component that comes from decomposition is a two-dimension matrix, $H_{m\times n}$. Each element of this array is an element or pixel in component images. It can stand for any of types such as grey scale, color component, bit-plane or the part of the original image. For example, a RGB image before encrypted can be decomposed three components which are called red, green, and blue components, as can be seen from Figure 3.

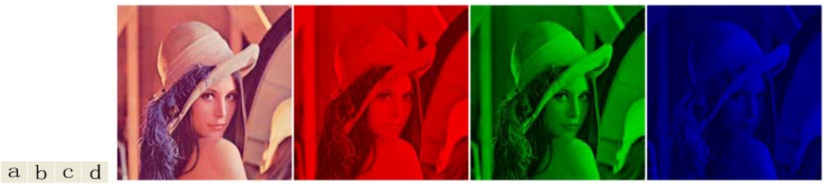

Figure 3. A method of decomposition for RGB images. (a) Original Lena image; (b) Red component; (c) Green component; (d) Blue component.

2.2. Shuffle

Shuffle is to mask original organization of the pixels of the image. Specifically, it permutes all the pixels of the image without changing their values. We exploited the known cat map to solve the problem (Figure 2). The classical cat map is a two-dimensional invertible chaotic map [4]. It is

$$\begin{bmatrix} x_{n+1} \\ y_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab+1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \bmod M \tag{1}$$

where the two parameters a and b, namely K_a and K_b in Figure 2, are positive integers, M is the width or height of the images. The map is area-preserving since the determinant of its linear transformation matrix is equal to 1. Let the eigenvalues of the transformation matrix be σ_1 and σ_2, given by

$$\sigma_1 = 1+\frac{1}{2}\left(ab+\sqrt{4ab+a^2b^2}\right)>1,\ \sigma_2 = 1+\frac{1}{2}\left(ab-\sqrt{4ab+a^2b^2}\right)<1 \tag{2}$$

According to the Lyapunov characteristic exponent theory [11], the map is chaotic. If $a=1$ and $b=1$, it is the classical Arnold cat map. However, the original component image reappears if it iterates enough times. Let the period of the map (2) is p. We determine the $\lfloor p/2 \rfloor$ as the rounds of permutation. The reason is that the correlation of two adjacent pixels in different directions, including horizontal, vertical and diagonal direction, is the weakest. For eample, the period is 11 for the Cat image, and the variation of these correlation coefficients can be seen from Table 1, whose computing method can be seen from the section 3.2.

Table 1. Correlation coefficients of two adjacent pixels under different rounds in the Cat image

Rounds	0	1	2	3	4	5	6	7	8	9	10	11
Horizon	84.0	75.8	59.4	28.0	-6.2	0.7	-5.1	-0.6	1.8	42.9	68.9	80.0
Vertical	83.7	75.2	58.9	27.1	-3.5	0.6	-5.7	-0.7	2.0	43.5	67.9	81.0
Diagonal	82.3	74.8	58.1	26.9	3.1	0.5	5.3	-0.5	1.7	42.5	66.2	79.6

2.3. Diffusion

If an image encryption algorithm only has the shuffle, its security is weak because the cat map is an invertible discrete map without mixing the pixels' values. In other words, the map does not change the statistical properties of the plain-text such as the intensity distribution of the pixels. As a remedy, we thereby resort to the diffusion.

It is very necessary for diffusion to choice a chaotic map. The Logistic map is a decent option. Its mathematical expression is

$$x_{n+1} = \mu x_n(1 - x_n) \tag{3}$$

where $\mu \in (0,4)$ is a parameter and $x \in (0,1)$ and $n = 0,1,2\cdots$. The parameter μ and the initial value x_0 may be taken as the key (Namely, μ means k_μ and x_0 means the feature value extracted from the component in Figure 2). In order to explain the chaotic properties, we divide the parameter μ into eight segments (Table 2). Note that the Logistic map is not chaos until the value of the parameter μ is between 3.5699 and 4. For Lyapunov exponents of Logistic map, let λ be its value. Then it can be obtained by the function given as

$$\lambda = \lim \frac{1}{n}\sum_{n=0}^{n-1} \ln\left|\frac{df(x_n,\mu)}{dx}\right| \tag{4}$$

Therefore, the sequence of Logistic map can be used to encrypt images when μ is larger than 3.5699 and smaller 4.

Table 2. The numbers of periodic orbits varies with the different parameter μ

μ	0-1	1-3	3-3.4995	3.4995-3.5441	3.5441-3.5644	3.5644-3.5688	…	3.5699-4
Orbits	0	1	2	4	8	16	…	Chaos

In [12], however, the authors demonstrated that chaotic encryption systems can be easily attacked and suggested the adoption of nonlinear functions to change the key continuously. There are two methods can improve the security. One is to enlarge the key space, and another is to strong the nonlinear map. For the former, we use two Logistic maps to encrypt the component image passed down from the shuffle. One map encrypts the odd rows of the component and another map encrypts the even rows. The equation is

$$\begin{cases} x_{n+1}^{odd} = \mu^{odd} x_n^{odd}(1 - x_n^{odd}) \\ x_{n+1}^{even} = \mu^{even} x_n^{even}(1 - x_n^{even}) \end{cases} \tag{5}$$

Additionally, we can take $x_0 = (x_0 + k)/2$ to change the initial condition. The notation k denotes the aided key which is the feature value selected from the component image (Figure 2). Let the component intensity vary from 0 to 255, then k is equal to $[(sum(A) \bmod 256) * 0.1/255]$, where notation function $sum(A)$ denotes the sum of the component image intensity because the initial condition value x_0 amended must be varied from 0.1 to 1. The keys have μ^{odd}, μ^{even}, x_0^{odd}, x_0^{even}, k^{odd}, k^{even} instead of μ and x_0. The key space, therefore, enlarges three times. For the later, we use a transformation to improve the nonlinear map. It is

$$y_n = (\arcsin\sqrt{x_n})/\pi \tag{6}$$

where π denotes the circle index pi. The range of dependent variable x_i is 0 to 1, so the value of independent variable y_i is 0 to 1. Note that the sequence of independent variable y_i has a uniform probability density function. It is obvious the two methods increase the security level.

We digitized the chaotic sequence by amplifying it with a proper scaling and sampling once it was obtained by Equation (7). Let τ_i be the digitized value, then it was $\lfloor 10^{15} y_i \rfloor \bmod N$. where the notation $\lfloor \bullet \rfloor$ denotes that it rounds the elements to the nearest integers less than or equal to $\bullet$, and N is the intensity level of pixels. Then the output of encryption can be generated by the formula:

$$C(i) = \tau(i) \oplus I(i) \oplus C(i-1) \tag{7}$$

where I_i is the currently operated pixel and C_{i-1} is the previously output cipher-pixel, the initial condition C_0 is equal to $\tau_0 \oplus I_0$, and the notation $\oplus$ denotes bitwise XOR.
Obviously the inverse transformation of the above equation is given by

$$I(i) = C(i) \oplus \tau(i) \oplus C(i-1) \tag{8}$$

2.4. Combination

After the preceding three stages have been completed, we can obtain each cipher component image. The next tasks are combining them. The simplest method is to organize the entire cipher component image to a cipher image in their original order. Of course, we should design a permutation of the cipher component images sequence to further improve their security.

Let the component image sequence is $\{1,2,3,\cdots,N\}$ which is called input sequence, and its permutation is $\{T_1,T_2,T_3,\cdots,T_N\}$, called output sequence. Then a transformation can be defined as

$$T_j = ((F_p(i)+\varepsilon)\cdot j) \bmod F_p(i+1) \quad \text{s.t. } F_p(i)+\varepsilon < F_p(i+1) \text{ and } N = F_P(i+1)-1 \tag{9}$$

where $F_p(i)$ is the P-Fibonacci sequence and the non-negative integer i is the index location of it, $F_p(i)$ and $F_p(i+1)$ are two consecutive elements in the P-Fibonacci sequence, the constant ε is a minimal integer offset such that the greatest common divisor of $F_p(i)+\varepsilon$ and $F_p(i+1)$ is 1, $\{j\}$ is the input sequence and $\{T_j\}$ is the output sequence [13,14]. Note that there are two important constraints for the transformation. One is $F_p(i)+\varepsilon < F_p(i+1)$ that is a limitation for choosing the minimal offset ε and another is $N = F_P(i+1)-1$ which specifies the maximum value of the input sequence. For example, if $p=2$, $F_p(i)=9$, then $F_p(i+1)=13$, $N=12$ and $\varepsilon=0$ can be easily got.

Obviously, the output sequence $\{T_1,T_2,T_3,\cdots,T_N\}$ is the permutation of an input sequence $\{1,2,3,\cdots,N\}$. For example, for $p=2$, the output sequence will be $\{9,5,1,10,6,2,11,7,3,12,8,4\}$ if input sequence is $\{1,2,3,4,5,6,7,8,9,10\}$. In other words, given a specific p, the output sequence is not different for the same input sequence according to Equation (9). Therefore, p can act as a key parameter by whitch we can easily assemble all the component images to a reconstructed image. However, the assemblage is not necessary for real-time Internet applications. The sender only transmits all the cipher component images labeled and the value p. The receiver can take advantage of them to recover the original structure according to Equation (9).

3. Tests and Security Analysis

As can been seen from the section 2, the more the number of component-images is, the larger the key space will become. In the section, we will analyze the security of our encryption algorithm in key space, correlation of neighboring pixels, pixels intensity distribution, and sensitivity, and take Figure 4(left) as test images with different size.

3.1. Key Space

Different number of component-images has different key space in the encryption algorithm. The simplest case is only a component image such as grey-scale images (of course, it may be split to blocks). A RGB image may be decomposed to 3 components. Specifically for a multi-layered image such as TIFF image with 27 frames, it may be decomposed to 51 components. Let the number of components be N and the notation key be the key to encrypt an image. Then the key can be taken a representation by Equation (10)

The right side of this equation is a $8 \times N$ matrix which each column vector is taken as acomponent encryption key. Namely, each component image has eight keys in which (k_{i1}, k_{i2}) are two parameters of cat map (a,b) in Equation (1), $(k_{i3}, k_{i4}, k_{i5}, k_{i6}, k_{i7})$ are the parameters

$$key = \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} & k_{17} & k_{18} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} & k_{27} & k_{28} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ k_{N1} & k_{N2} & k_{N3} & k_{N4} & k_{N5} & k_{N6} & k_{N7} & k_{N8} \end{bmatrix}^T \tag{10}$$

and initial conditions of Logistic map （$\mu^{odd}, \mu^{even}, x_0^{odd}, x_0^{even}, k)$ in Equation (5), k_{i8} is the parameter p in Equation (9). For an instace, each component image is encrypted by using the key vector (1, 1, 4.0, 4.0, 2.0, 7.0, k, 2). The encryption test results of original images (Figure 4(left)) have been shown in Figure 4(right). Visually the textures of encrypted images are different though all of them like noise images

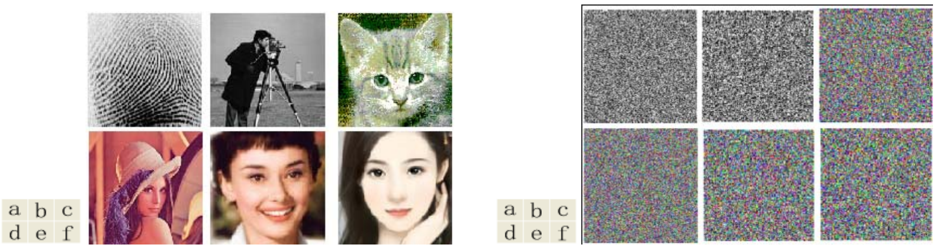

Figure 4. Original images with different size (left) and their encrypted images. (a) Fingerprint, 170*170; (b) Cameraman, 140*140; (c) Cat, 144*144*3; (d) Lena, 256*256*3; (e) Smile, 120*120*3; (f) Lady, 128*128*3.

3.2. Correlation of Neighboring Pixels

The following issues will be discussed on the basis of a component because all the properties are similar for the components. The neighboring pixels are also called adjacent pixels that indicate two horizontally, vertically and diagonally neighboring pixels. The information of the adjacent pixels correlations is very important for the opponent who may discover some useful information to take statistical attacks. If the adjacent pixels have a strong linear correlation, the cryptosystem is easy to be attacked. Therefore, this section analyzes the correlation of the adjacent pixels in both the original image and the corresponding encrypted image.

Figure 5 plots the pixel intensity distributions of two adjacent pixels at different directions in the original and encrypted Lena image. As can be seen from the Figure 5 that the correlation of the adjacent pixels is highly strong at different directions in the original image, however, in the cipher image it is so weak that their distributions are random.

In another way, the correlation coefficient of the adjacent pixels at different directions can assess their correlation quantitatively. It can be calculated by

$$r_{xy} = \left(M \sum_{i=1}^{M} x_i y_i - \sum_{i=1}^{M} x_i \sum_{i=1}^{M} y_i \right) \Bigg/ \sqrt{\left(M \sum_{i=1}^{M} x_i^2 - \left(\sum_{i=1}^{M} x_i \right)^2 \right) \left(M \sum_{i=1}^{M} y_i^2 - \left(\sum_{i=1}^{M} y_i \right)^2 \right)} \tag{11}$$

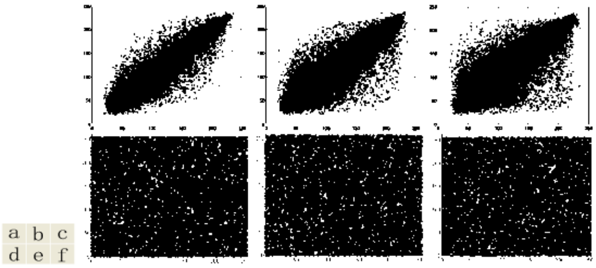

Figure 5. The pixel intensity distributions of two adjacent pixels at different directions in the original and encrypted Lena image. (a) Horizontal, original image; (b) Vertical, original image; (c) Diagonal, original image; (d) Horizontal, encrypted image; (e) Vertical, encrypted image; (f) Diagonal, encrypted image.

where r_{xy} is the value of the correlation coefficient, x, y are the intensity values of two adjacent pixels and M is the total number of pixels from the image. The test results of original and encrypted Lena image can be seen from Table 3. The same conclusion has been obtained because the value of r_{xy} can indicate the relationship between two adjacent pixels.

Table 3. Correlation coefficients of two adjacent pixels in both original and encrypted image

	Original image	Encrypted image
Horizontal	0.91675	0.01182
Vertical	0.95431	0.00014
Diagonal	0.90203	0.01478

3.3. Probability Density Function of Pixels

Apart from the correlation of neighboring pixels, the opponent can also take advantage of the probability density function of pixels to attack the cryptosystems statistically. For resistance to this statistic attack, the probability of the pixels in the cipher image should be a uniform invariant function. This section further demonstrates how make the presented algorithm robust when it is faced with statistic attacks.

Figure 6 shows the graph of the probability density function of pixels in the original image and the corresponding encrypted image. As can be seen from the Figure 6, the distribution of pixels (histogram) in the original images is completely different from the corresponding encrypted image. Moreover, different original images have different the distribution of pixels, but different cipher images preserve uniform invariant function. These demonstrate that the algorithm can withstand the statistic attacks.

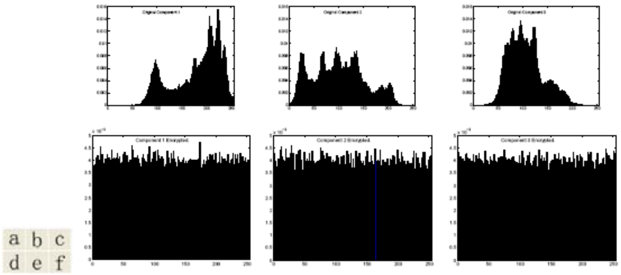

Figure 6. The probability density distribution of pixels in the components by the Lena's decomposition. (a) Original red component; (b) Original green component; (c) Original blue component; (d) Encrypted red component; (e) Encrypted green component; (f) Encrypted blue component.

3.4. Sensitivity

Sensitivity test to the security key and initial conditions is extremely important for an image cryptosystem. An encryption algorithm with good sensitivity can make differential attacks [4] infeasible. There are two common measures to assess the sensitivity. One is the number of pixels change rate (NPCR), another is the unified average changing intensity (UACI) [1]. NPCR means the change rate of pixels in cipher image when the value of one pixel in the original image is changed. It measures the percentage of different pixel numbers between the cipher images while their corresponding original images have only one different pixel. Let the cipher images be C_1 and C_2. Define a bipolar array D, with the same size as image C_1 or C_2. If $C_1(i,j)=C_2(i,j)$ then $D(i,j)=1$; otherwise, $D(i,j)=0$. The NPCR is defined by

$$NPCR=\left(\left(\sum_{i=1}^{H}\sum_{j=1}^{W}D(i,j)\right)\Big/\left(W\times H\right)\right)*100\% \tag{12}$$

where W and H are the width and height of both cipher images. UACI measures the average intensity of difference between the two cipher images. It is defined by

$$UACI=\frac{1}{W\times H}\left[\sum_{i=1}^{H}\sum_{j=1}^{W}\frac{\left|C_1(i,j)-C_2(i,j)\right|}{255}\right]\times 100\% \tag{13}$$

The six images have been tested and the experimental results are listed in Table 4. By analyzing these data, we can take a conclusion that about the half number of the pixels in cipher image change while only one pixel changes in the original image. These demonstrate that the proposed algorithm is so sensitive to initial condition that the differential attack would become very inefficient.

Table 4. NPCR and UACI

	Fingerprint	Cameraman	Cat	Lena	Smile	Lady	[Average]
NPCR(%)	49.23	50.87	46.68	45.59	53.37	52.74	49.7467
UACI(%)	0.243	0.257	0.248	0.231	0.269	0.266	0.2523

4. Conclusions

Our chaos-based image encryption algorithm contains four stages: decomposition, shuffle, diffusion and combination. In decomposition stage, an original image is decomposed to some components. Shuffle and diffusion stages are crucial to the security of whole encryption algorithm. Shuffle stage masks the original organization of the pixels and diffusion stage masks the original values of the pixels. The former does not change the probability distribution of pixels in image. The later make the probability distribution of pixels into uniform distribution. Note that the final stage is not necessary. We discuss the map's sensitivity to initial conditions and parameters. To evaluate the algorithm performance, we give out the test results and analyze the security of the encryption algorithm. Experiment data and analysis results demonstrate that the encryption algorithm meets the different security requirement and has the ability to withstand common attacks such as brute force, statistic, and differential attacks.

Aknowledagement

This work is partially surpported by Visiting Scholarship of State Key Laboratory of Power Transmission Equipment & System Security and New Technology (Chongqing University) under grant No. 2007DA10512711412.

References

[1] Mao YB, Chen GR. A Novel Fast Image Encryption Scheme Based on 3D Chaotic Baker Maps*, International Journal of Bifurcation and Chaos.* 2004; 14(10): 3613-3624.

[2] Wong KW, Kwok BSH, Law WS. A fast image encryption scheme based on chaotic standard map.

Physics Letters A. 2008; 372: 2642-2652.

[3] Gao HJ, Zhang YS, Liang SY, Li DQ. A new chaotic algorithm for image encryption*, Chaos, Solitons and Fractals*. 2006; 29: 393-399.

[4] Chen GR, Mao YB, Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons and Fractals.* 2004; 21: 749-761.

[5] Wong KW. A fast chaotic cryptographic scheme with dynamic look-up table, *Physical Letters A*. 2002; 298: 238-242.

[6] Krikor L, Baba S, Arif T, Shaaban Z. Image Encryption Using DCT and Stream Cipher, *European Journal of Scientific Research*. 2009; 32(1): 47-57.

[7] Al-Maadeed S, Al-Ali A, Abdalla T. A New Chaos-Based Image-Encryption and Compression Algorithm. *Journal of Electrical and Computer Engineering.* 2012; doi:10.1155/2012/179693

[8] Patel KD, Belani S. Image Encryption Using Different Techniques: A Review*, International Journal of Emerging Technology and Advanced Engineering*. 2011; 1(1): 30-34.

[9] Zhou YC, Panetta K, Agaian S, Chen CLP. Image encryption using P-Fibonacci transform and decomposition. *Optics Communications*. 2012; 285(5): 594-608.

[10] Zhou YC. Multimedia Security System for Security and Medical Applications. A dissertation for the degree of Doctor of Philosophy in Electrical Engineering. 2010.

[11] Wolf A, Swift JB, Swinney HL, et al. Determining Lyapunov exponents from a time series, *Physica D: Nonlinear Phenomena*. 1985; 16(3): 285-317.

[12] Sobhy MI, Shehata AER. *Methods of attacking chaotic encryption and countermeasures.* Proceedings of ICASSP '01. 2001; 1001-1004.

[13] Agaian S, Astola J, Egiazarian K, Kuosmanen P. Decompositional methods for stack filtering using Fibonacci p-codes. *Signal Processing*. 1995; 41: 101-110.

[14] Zhou YC, Panetta K, Agaian S. *P-recursive Sequence and Key-dependent Multimedia Scrambling.* Proceedings of SPIE, Mobile Multimedia/Image Processing, Security, and Applications. 2008; 6982: 69820H.