Acta Technica *62* (2017), No. 4A, 225–234 

# A new image encryption approach based on three-dimensional chaotic maps

Feng Huang[2,3], Huiming Deng[3]

**Abstract.** Since 1990s some two-dimensional chaotic maps, act as baker map and cat map, were used in image encryption. But the encryption using two-dimensional chaotic maps only shuffled the positions of the image pixels and hadn't changed the value of the pixel. It couldn't against statistical cryptanalysis. Here it proposed a new encryption approach based on three-dimensional chaotic maps which were used in encryption. Firstly, it introduced eight different two-dimensional chaotic maps which was realized by processing image stretch and fold. Secondly a plain image was extended to three-dimensions one by the value of pixels. Here 8-bit grayscale images could be divided into eight layers. Lastly it used the two-dimensional chaotic maps to encrypt each layer of the map. More detail studies proved that high correlation among adjacent pixels was rapidly charged and the value of pixel was changed at the same time. The encryption had good diffusion and confusion. It could satisfy fast security requirements in some public network.

**Key words.** Image encryption, three-dimensional map, chaotic map, map patterns.

## 1. Introduction

For personal privacy, image security need be satisfied before transmitted over some public network. People used some encryption technology to protect image from attacks. Among them, encryption based on chaos had certain advantages[1-4]. Some characteristics in chaos could be connected with the "confusion" and "diffusion" property in encryption. Such as chaos system was sensitive dependence on initial conditions or parameters, broadband power spectrum, randomness in the time domain[5].

Image had space redundancy for biggish relativity about neighborhood pixels.

[1]Acknowledgment - Authors gratefully acknowledge the Projects Supported by Scientific Research Fund of Hunan Provincial Education Department of China(15K032), the Projects Supported by National Natural Science Foundation of Hunan Province of China (16JJ6025).

[2]Workshop 1 - The Cooperative Innovation Center of Wind Power Equipment and Energy Conversion, Hunan Institute of Engineering, Xiangtan, Hunan, China

[3]Corresponding Author:Feng Huang; e-mail: hf7825@qq.com

Image permutation was an important part in encryption which could charge the position of pixels randomly. Some typical chaotic maps, the cat map, the baker map and the tent map, were used for image permutation. In Fridrich[6], a symmetric image encryption scheme was designed. The image permutations induced by the baker map behave as typical random image permutations. In [7,8], symmetric image encryption schemes based on three-dimensional chaotic maps were proposed. Image permutation employed by chaotic maps was an important part of encryption. In [9] a large chaotic permutation matrix was designed to achieve the high performance of pseudorandom permutation.

The chaotic maps shuffled the positions of image pixels and convert plain image into a new image not recognizable by its attackers. The permutation could protect image and against statistical cryptanalysis. But it couldn't resist plaintext attack. At the same time, there were a lot of week keys and duplicate keys in permutation based on those maps. In [6], the key was not sensitive enough. A bit change of key could not completely affect the results of encryption and decryption. In [10], a new chaotic map was proposed. Unfortunately, there were some security risks. Act as the number "0" was a weak key. A good way was to use chaotic maps to both change the pixel position and pixel value. The encryption had good diffusion and confusion.

SCAN patterns [11,12] could be used in image encryption. But it required the plain image must be square and its size be even. By the idea of patterns, the paper proposed eight different chaotic maps. Those maps were used to encrypt each lays of an image. An image encryption approach was realized by them. The high correlation among adjacent pixels was rapidly charged and the value of pixel was changed. The approach had good diffusion and confusion. It could satisfy fast security requirements in some public network.

## 2. The Principle of the Chaotic Maps

Suppose that a square image consists of $N \times N$ pixels with $L$ gray levels. The chaotic map was realized by processing image stretch-and-fold. Firstly, the square image was divided into two isosceles triangles along the one diagonal, utilizing the difference of the pixel numbers of two adjacent columns of the triangles, each pixel in one column was inserted to the next adjacent column. Then, the plain image could be stretched to a new pixel's line. Finally, the line was folded over to a new square image whose size was the same as the plain image. It shuffled the positions of image pixels. The process was invertible which is seen in Figure.1. For the different diagonal direction, there were two maps, map a and map b here.

Examples were given here. The process of the map a permutation was shown in Figure.2. The image with $4 \times 4$ pixels, that was $N=4$. The pixels join to another part of line. Then it connected two parts to a line. Lastly it was from a pixel's line to a square image different from the originally one. The map b was symmetric with the map a.

Supposing the dimension of a square image is $N \times N$, where $N$ is an integer. A($i$, $j$) is the matrix of a square image, in which each element corresponds to a gray-level value of the pixel ($i$, $j$); $L(i)$, $i=0, \ldots, N-1$, $j=0, \ldots, N-1$. $N$ is a one-dimensional

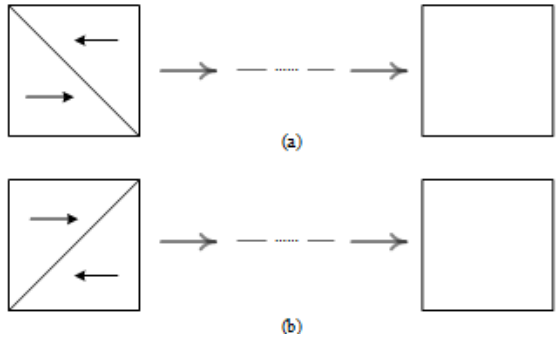


Fig. 1. The principle of the maps

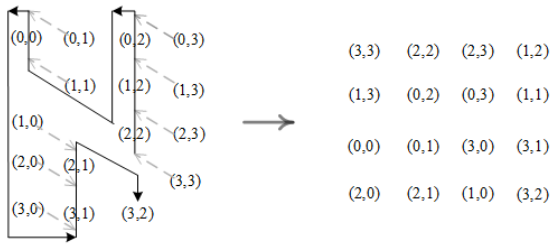


Fig. 2. The process of the map a with a 4×4 image

vector mapped from A.

The map a is shown in Figure.1 (a). The algorithm of the image is described with the following formula:

$$L[\frac{(N+j+2)(N-j-1)}{2}+2(j-i)]=A(i,j) \tag{1}$$

while $j$ $i$, $N$- $j$ is odd number, $i$=0, ..., $N$-1, $j$=0, ..., $N$-1.

$$L[\frac{(N+j+3)(N-j-2)}{2}+2(j-i)+1]=A(i,j) \tag{2}$$

while $j$ $i$, $N$- $j$ is even number, $i$=0, ..., $N$-1, $j$=0, ..., $N$-1.

$$L[\frac{N^2+N+(2N-j-1)\times j}{2}+2(N-i-1)]=A(i,j) \tag{3}$$

while $j$<$i$, $j$ is even number, $i$=0, ..., $N$-1, $j$=0, ..., $N$-1.

$$L[\frac{N^2+N+(2N-j)\times (j-1)}{2}+2(N-i)-1]=A(i,j) \tag{4}$$

while $j$<$i$, $j$ is odd number, $i$=0, ..., $N$-1, $j$=0, ..., $N$-1.
The map b is shown in Figure.1 (b). The algorithm of the image is described with

the following formula:

$$L[\frac{(N+j+2)(N-j-1)}{2}+2(j-i)]=A(i,N-1-j) \tag{5}$$

while $j$ $i$, $N$- $j$ is odd number, $i$=0, ..., $N$-1, $j$=0, ..., $N$-1.

$$L[\frac{(N+j+3)(N-j-2)}{2}+2(j-i)+1]=A(i,N-1-j) \tag{6}$$

while $j$ $i$, $N$- $j$ is even number, $i$=0, ..., $N$-1, $j$=0, ..., $N$-1.

$$L[\frac{N^2+N+(2N-j-1)\times j}{2}+2(N-i-1)]=A(i,N-1-j) \tag{7}$$

while $j<i$, $j$ is even number, $i$=0, ..., $N$-1, $j$=0, ..., $N$-1.

$$L[\frac{N^2+N+(2N-j)\times (j-1)}{2}+2(N-i)-1]=A(i,N-1-j) \tag{8}$$

while $j<i$, $j$ is odd number, $i$=0, ..., $N$-1, $j$=0, ..., $N$-1.
The line of $N\times N$ pixels $L$ is further mapped to a same size $N\times N$ square image, B. Here there are four methods to do this. It can be seen in Figure.3. Method one is

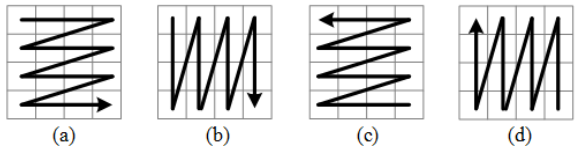


Fig. 3. Four folding algorithms

shown in Figure.3(a). While $i$=0, ..., $N$-1, $j$=0, ..., $N$-1, the map from line $L$ to image $B$ is described with the following formula:

$$B(i,j)=L(i\cdot N+j) \tag{9}$$

Another method is shown in Figure.3(b). The map from line L to image B is described with the following formula:

$$B(i,j)=L(j\cdot N+i) \tag{10}$$

Another method is shown in Figure.3(c). The map from line L to image B is described with the following formula:

$$B(i,j)=L((N-1-i)\cdot N+N-1-j) \tag{11}$$

Another method is shown in Figure.3(d). The map from line L to image B is described with the following formula:

$$B(i,j)=L((N-1-j)\cdot N+N-1-i) \tag{12}$$

Using the maps and folding methods, it could get eight complete chaotic maps. Act as the map A may means using the map b stretched the square image to a line and using formula (9) fold the line to another image, shown in Table.1.

Table 1. The map patterns

| | Map Patterns | Form of map Patterns |
|---|---|---|
| 1 | map A | The map b + formula(??) |
| 2 | map B | The map a + formula(??) |
| 3 | map C | The map b + formula(??) |
| 4 | map D | The map a + formula(??) |
| 5 | map E | The map b + formula(??) |
| 6 | map F | The map a + formula(??) |
| 7 | map G | The map b + formula(??) |
| 8 | map H | The map a + formula(??) |

## 3. Extension to three-dimension map

The square image consists of $N \times N$ pixels with $L$ gray levels. The gray level value of each pixel A is in decimal which can be expressed as a binary number.

It can split the plain image into eight layers. As shown in Figure.4, the first layer is composed by the lowest coefficients of the binary number of image values; the second layer is composed by the second... and so on.

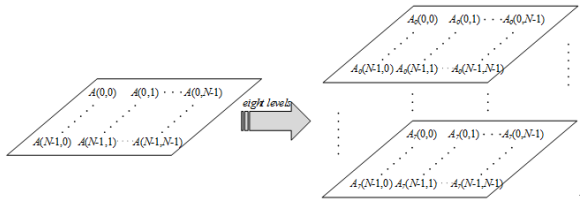


Fig. 4. Delamination of image

## 4. A image encryption approach base on the maps

Here, the plain image was shown in Figure.5. It has 256×256 pixels with 8-bit grayscale. It could be divided into eight layers by formula (13). The eight complete chaotic maps in Table.1 could be used to permute every layer's pixels. At the same time, it also charged the value of pixels in plain image. The method of key design could be seen in Table.2. The number of maps of each lay was used as the key. The total number of maps was 100. Act as the key is "0123", the "0" means the number of map A was 20, map B was 20, map C was 10 and so on, the "1" means the number of map A was 20, map B was 10, map C was 20 and so on....

An image encryption approach was carried out based on the map patterns. The plain image was encrypted using the maps by the *key* "0" and "0123". By Figure.5, it could be seen that the plain image has been encrypted. Figure.6 showed the value of image was changed. The time of encryption by *key* "0" was 0.0997s; the time of decryption by *key*"0" was 0.1012s. (the CPU of PC was Intel's core i5 1.6Ghz, the ram was 4G, and the operating system was Windows 10).

Table 2. Key design

| number of map key | map A | map B | map C | map D | map E | map F | map G | map H |
|---|---|---|---|---|---|---|---|---|
| 0 | 20 | 20 | 10 | 10 | 10 | 10 | 10 | 10 |
| 1 | 20 | 10 | 20 | 10 | 10 | 10 | 10 | 10 |
| 2 | 10 | 20 | 20 | 10 | 10 | 10 | 10 | 10 |
| 3 | 10 | 120 | 10 | 20 | 10 | 10 | 10 | 10 |
| 4 | 10 | 10 | 20 | 20 | 10 | 10 | 10 | 10 |
| 5 | 10 | 10 | 20 | 10 | 20 | 10 | 10 | 10 |
| 6 | 10 | 10 | 10 | 20 | 20 | 10 | 10 | 10 |
| 7 | 10 | 10 | 10 | 20 | 10 | 20 | 10 | 10 |
| 8 | 10 | 10 | 10 | 10 | 20 | 20 | 10 | 10 |
| 9 | 10 | 10 | 10 | 10 | 20 | 10 | 20 | 10 |

*Key space.* Since the length of the key in encryption had no limit, its key space could be calculated according to the length of the key. Suppose the key was represented in decimal. The relationship between the key space size and the key length was shown in Table.3. In theory, security key could be any long integer to satisfy the different security requirements. But the encryption speed was related to the length of the key. So it was better to made arrangements on key length before encryption.

Table 3. Key space size vs keys length

Fig. 5. Plain image and cipher image

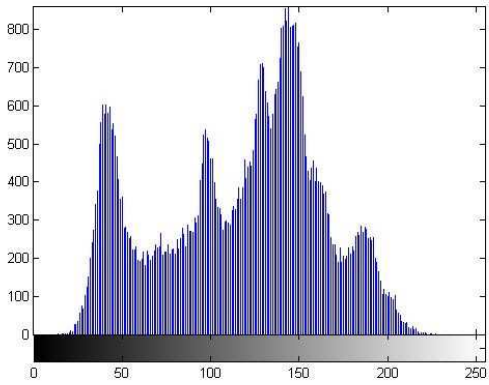


Fig. 6. the histogram between plain image and cipher one

| Key length (digit) | 6 | 7 | 8 | 9 |
|---|---|---|---|---|
| Key space size | $10^6$ | $10^7$ | $10^8$ | $10^9$ |

*Correlation.* Correlation of two adjacent pixels in a cipher image, , , Where x and y were gray-scale values of two adjacent pixels in the image.

Figure.7 showed the correlations of two horizon-tally adjacent pixels in plain image and cipher image (key was "0123"): the correlation coefficients were 0.9442,

0.0006. Similar results for diagonal and vertical directions were shown in Table.4.

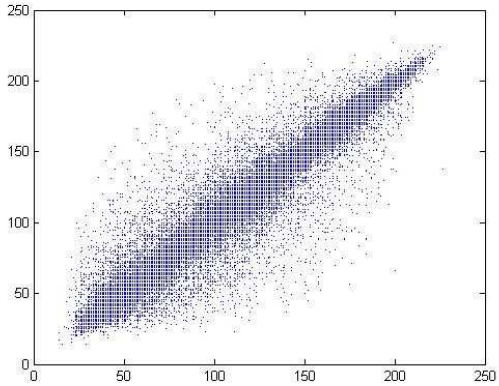


Fig. 7. Correlations of adjacent pixels

Table4. Correlation coefficients of two adjacent pixels

| | Plain image | Cipher image |
|---|---|---|
| horizontal | 0.9442 | 0.0006 |
| vertical | 0.9711 | 0.0007 |
| diagonal | 0.9187 | 0.0002 |

*Fixed point ratio.*Where *key* was "0123" BD=0.45%. Those meant the positions of the 99.55% plain image pixels were charged.

*Change of the gray.*Where *key* was "0123" GAVE = 71.41. Those meant the average values of the pixels were charged by 27.7%.

*R-m self-relevance.* Where $r$=1, the $r$-$m$ self-relevance could see in Table.5, here *keys* was "0123". It could be proved the self-relevance of cipher image significantly reduced compared with the plain image. In Table.5, the value of self-relevance was even smaller than the value when $m$=1. Those mean the effect of permutation was very good.

Table 5. Self-correlation of images

| *m* | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *lena* | 0.41 | 0.41 | 0.46 | 0.50 | 0.54 | 0.57 | 0.60 | 0.62 | 0.64 | 0.66 | 0.68 | 0.69 | 0.70 |
| *keys* | 0.13 | 0.13 | 0.14 | 0.14 | 0.14 | 0.15 | 0.15 | 0.16 | 0.16 | 0.16 | 0.17 | 0.17 | 0.18 |

## 5. Summary

Some chaotic maps were used in image permutation for only shuffling the positions of image pixels. Here the 8-bit grayscale images were divided into eight layers. It used some different chaotic map to permute every layer's pixels. The process had good diffusion and confusion. It proposed a three-dimensional encryption approach based on new eight chaotic maps. The chaotic maps realized by processing image stretch-and-fold. It used the eight-different chaotic map to permute every layer's pixels. The process permuted the position of pixels and also change the value of image. The advantages of the approach could be described as follows: 1) the approach was quite simple, enough safe and fairly fast. 2) the encryption had no message loss. 3) the key space could be enough big to satisfy security requirements.

### References


[1] J. M. Zheng, W. Z. Gao, N. C. Jain: *Color image encryption algorithm based on chaotic map.* Comp. Eng. Design *32* (2011), No. 9, 2934–2937.

[2] Y. Wang, K. Wong, X. Liao, G. Chen: *A new chaos-based fast image encryption algorithm.* Applied Soft Comput *11* (2011), No. 1, 514–522.

[3] J. X. Chen, Z. L. Zhu, C. Fu, H. Yu, L. B. Zhang: *A fast chaos-based image encryption scheme with a dynamic state variables selection mechanism.* Communications in Nonlinear Science and Numerical Simulation *20* (2015), No. 3, 846–860.

[4] X. J. Tong: *The novel bilateral - Diffusion image encryption algorithm with dynamical compound chaos.* Journal of Systems & Software *85* (2012), No. 4, 850–858.

[5] A. S. Meno, K. S. Sarila: *Image encryption based on chaotic algorithms: An overview.* Int. J. Science, Engineering and Technology Research *2* (2013), No. 6, 1328–1332.

[6] J. Fridrich: *Symmetric ciphers based on two-dimensional chaotic maps.* Int. J. Bifurcat Chaos *8* (1998), No. 6, 1259–1284.

[7] Y. B. Mao, R. G. Chen, G. S. Lian: *A novel fast image encryption scheme based on 3D chaotic baker maps.* Int. J. Bifurcat. Chaos *14* (2004), No. 10, 3613–3624.

[8] R. G. Chen, Y. B. Mao, C. K. Chui: *A symmetric image encryption scheme based on 3D chaotic cat maps.* Chaos Solitons Fractals *21* (2004), No. 3, 749–761.

[9] J. W. Yoon, H. Kim: *An image encryption scheme with a pseudorandom permutation based on chaotic maps. Commun.* Nonlinear Sci. Numer. Simul *15* (2010), No. 12, 3998–4006.

[10] F. Huang, Y. Feng: *A symmetric image encryption scheme based on a simple novel two-dimensional map.* Int J Innovative Computing, Information and Control *3*, (2007), No. 6, 591–1600.

[11] S. S. Maniccam, N. G. Bourbkis: *Image and video encryption using SCAN patterns. Pattern Recogn.* Pattern Recogn *37* (2004), No. 4, 725–737.

[12] D. Radu, D. Ioana, K. Hyongsuk: *Chaotic Scan: A low complexity video transmission system for efficiently sending relevant image features.* IEEE Transaction on circuits and systems for video technology *20* (2010), No. 2, 317–321.


Received November 16, 2016