Chaotic Video Encryption Algorithm Based on Baker Map

ZHANG Meng, WANG Fan-zhen, LIU Zhong-xin, SUN Qing-lin,
CHEN Zeng-qiang, YUAN Zhu-zhi
(*Department of Automation, NanKai University, Tianjin* 300071)

Abstract With the increase of Internet bandwidth, applications based on video stream have been developed rapidly. But the applications, such as digital video, video mail or videophone, are very vulnerable in network environment. Therefore video security has become an important research field. In recent years, a lot of image and video encryption algorithms and techniques have been used to solve the problem. However, most of them have drawbacks in security or speed. Combining with other algorithms' virtues, this article puts forward a method that uses both stream ciphers to generate pseudo-random numbers and block ciphers to do permutation. A 4-dimensional hyperchaotic function is chosen as stream ciphers, and Baker map as block ciphers. The improved Baker map can withstand known-plaintext attack in simple form. And the 4-dimensional hyperchaotic function can not only enhance encryption speed but also increase cipher complexity. This method, which can be simply realized by software, is independent of any kind of video compression algorithms, while providing high security for real-time digital video with fast encryption speed.

Keywords Baker map, hyperchaotic, video stream, encryption, stream cipher, block cipher
中图法分类号：TP309 TN918.74 **文献标识码**：A **文章编号**：1006-8961(2006)09-1327-07

基于 Baker 映射的视频流加密算法

张 萌 王繁珍 刘忠信 孙青林 陈增强 袁著祉
（南开大学自动化系，天津 300071）

摘 要 随着 Internet 带宽的不断增长，基于流媒体传输的视频应用得到了迅速的发展。然而，由于数字电视、视频邮件、可视电话等具体视频流应用在网络环境中很容易遭受人为的攻击，因此，视频流的网络安全成为当前亟待解决的重要研究课题之一。近年来，虽然出现了很多图像和视频的加密算法，然而很多算法在安全性或加密速度上存在缺陷。为此，结合其他算法的优点，提出了一种流密码与块密码相结合的加密算法。其中流密码用来产生伪随机序列，块密码用来置乱数据。由于 Baker 映射经改进后可以抵挡已知明文攻击，并且实现简单，而 4 维超混沌方程则不但可提高加密速度，还可增加密码的复杂度，因此可将 Baker 映射与 4 维混沌伪随机序列发生器相结合，前者用于块密码加密，后者用于流密码加密，其不但具有安全性高、速度快的特点，并且与视频压缩算法相独立，实现简单。实验结果表明，该算法可以进行实时视频传输和处理。

关键词 Baker 映射 超混沌 视频 加密 流密码 块密码

基金项目：国家自然科学基金项目(60374037, 60574036)；教育部新世纪优秀人才支持计划项目(2005-290)；高校博士学科点专项基金项目(20050055013)
收稿日期：2005-05-09；**改回日期**：2005-09-15

第一作者简介：张萌(1980～)，女。2003 年获南开大学学士学位，现为南开大学硕士研究生。主要研究方向为计算机网络与通信、图像与媒体加密。E-mail：mengah@mail.nankai.edu.cn

1 Introduction

Along with the development of network technology, more and more importance has been devoted to Multimedia application security. A lot of image and video encryption methods have been brought forward to solve the problem. However, most of them have defects in security or speed.

In video selective encryption algorithms, stream ciphers have been used widely. Shi and Bhargava encrypted sign bits of DCT(discrete cosine transform) coefficients or that of motion vectors in reference [1] for MPEG (moving picture experts group) video stream. Chun Yuan encrypted DC (direct current) coefficient and motion vectors by a chaotic sequence in reference [2]. These encryption algorithms have low complexity and high speed, but they can't withstand known plain-text attack. Suppose eavesdroppers know plain text and cipher text, they will get the keys easily by making XORs (exclusive OR) between corresponding bits(e. g. DC coefficient) of plain text and cipher text. Therefore, it is not safe enough to encrypt a video stream just by stream ciphers. To escape from known plain-text attack, some complicated modifications should be made, such as permutation.

Combining with other algorithms' virtues, we put forward a method that uses both stream ciphers to generate pseudorandom numbers and block ciphers to do permutation. The block cipher encryption can withstand known plain-text attack, while stream cipher encryption can increase cipher complexity. A 4-dimensional hyperchaotic function as pseudorandom sequence generator is used as stream cipher encryption. It can generate four hyperchaotic sequences at the same time, which not only enhances encryption speed but also increases cipher complexity. The block cipher encryption we choose is a 2-dimensional chaotic map, which is used for creating complex, key-dependent permutations. The map has properties of sensitivity to initial conditions and parameters, mixing, and bijection. This method, which can be simply realized by software, is independent of any kind of video compression algorithms, while providing high security for real-time digital video with fast encryption speed.

This paper is organized as follows. In Sect. 2, we firstly give a brief survey of Baker map including 2-dimensional continuous Baker map and 2-dimensional discrete Baker map. The algorithm scheme is described in detail in Sect. 3. Security analysis and test results are showed in Sect. 4. The last section is the conclusion of this paper.

2 Baker Map

There are a number of chaotic maps (e. g. the Baker map, the Cat map, the Standard map, etc.), which seem to be suitable for ciphering purpose. However, the maps we select must be simple so that the ciphering phase can be performed quickly. The maps should also have a large key space. Based on above, we select Baker map as block cipher encryption. Baker map has properties of sensitivity to initial conditions, mixing, and bijection. Furthermore, it's a geometrical map that can be parameterized naturally with a large number of keys.

2.1 Continuous Baker map

The Baker map $\boldsymbol{B}$, is described with the following formulas

$$B(x,y) = (2x, y/2) \quad \text{when } 0 \leqslant x < 1/2$$

$$B(x,y) = (2x-1, y/2+1/2) \quad \text{when } 1/2 \leqslant x < 1$$

The Baker map is a chaotic bijection of the unit square $I \times I$ onto itself. The square is divided into k vertical rectangles $[F_{i-1}, F_i] \times [0, 1]$, $i = 1, \cdots, k$, $F_i = p_1 + \cdots + p_i$, $F_0 = 0$ such that $p_1 + \cdots + p_k = 1$ (see Fig. 1). The generalized Baker map stretches each rectangle horizontally by the factor $1/p_i$. At the same time, the rectangle is contracted vertically by the factor pi. Finally, all rectangles are stacked on top of each other as in Fig. 1. Formally,

$$B(x,y) = \left(\frac{1}{p_i}(x - F_i), p_i y + F_i\right) \quad (1)$$

for $(x,y) \in [F_i, F_i + p_i) \times [0,1)$

2.2 Discrete Baker map

Follow the approach[3] below to discretize Baker

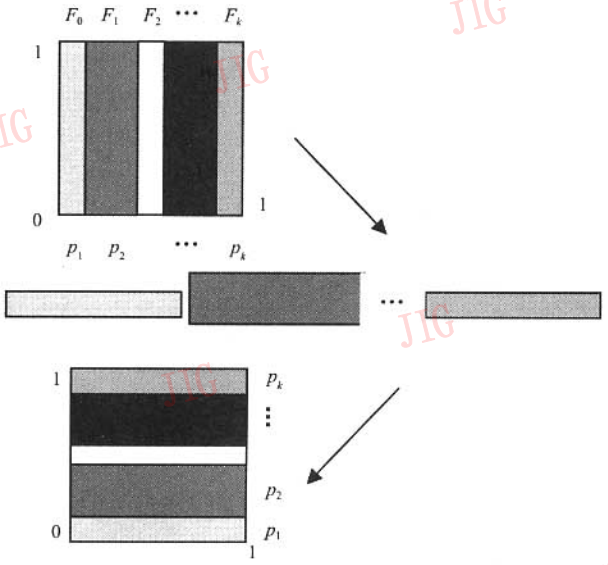


Fig. 1　Generalized Baker map

map.

(1) An $N \times N$ square is divided into vertical rectangles of height N and width n_i, where each integer n_i divides N.

(2) each vertical rectangle $N \times n_i$ is divided into n_i boxes $N/n_i \times n_i$ containing exactly N points (see Fig. 2).

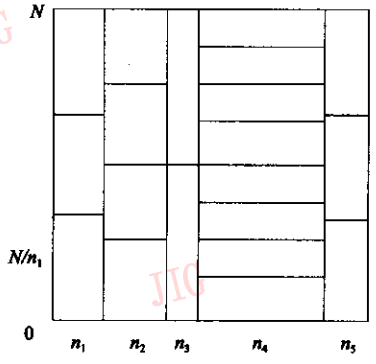


Fig. 2　Discretized version of the Baker map

(3) Each of these boxes is mapped to a row of pixels from bottom to top and left to right.

(4) All rows are stacked on top of each other.

The discrete generalized Baker map is denoted as $B_{(n_1,\cdots,n_k)}$, where $N_i = n_1 + \cdots + n_i, n_i \mid N, i = 1, \cdots, k$, such that $n_1 + \cdots + n_k = N$.

The pixel (r, s), with $N_i \leqslant r < N_i + n_i, 0 \leqslant s < N$ is mapped to

$$B_{(n_1,\cdots,n_k)}(r,s) = \left(\frac{N}{n_i}(r - N_i) + s \bmod \frac{N}{n_i}, \frac{n_i}{N}\left(s - s \bmod \frac{N}{n_i}\right) + N_i\right) \quad (2)$$

3　The Proposed Algorithm

Using stream ciphers or block ciphers alone will induce hidden trouble in security. Many stream ciphers can not withstand known plain-text attack. While block ciphers require to iterate many times to make the cipher-text independent of the plain-text, which will markedly reduce the encryption speed. Applying the two ciphers together, it not only improves security greatly, but decreases iteration times to enhance encryption speed. Therefore, we combine Baker map and hyperchaotic stream ciphers in our encryption algorithm.

Firstly, we extend Baker map from two dimensions to three dimensions in the following way[3]. Add a function $h(r,s)$, which could be any function about r and s, where r and s are coordinates of an image. Suppose that pixel (r, s) maps to a new location $(\hat{r}, \hat{s})$ after Baker map. Then the gray level of the new pixel is $\boldsymbol{new\_pixel}[\hat{r}][\hat{s}] = (\boldsymbol{pixel}[r][s] + \boldsymbol{h}(r,s)) \bmod L$, where $\boldsymbol{pixel}[r][s]$ denotes the gray level of the $\boldsymbol{pixel}(r,s)$ and L denotes gray levels. The 3-dimensional chaotic map leads to a substitution cipher that can create a random image with uniform histogram in a few iteration times.

Then, the map discussed above has no diffusion mechanism, which is potentially dangerous and makes the method vulnerable to a chosen plain-text type of attack. We insert a diffusion step into the encryption scheme after the permutation and gray level mixing. Make $\boldsymbol{pixel}[r][s] = \boldsymbol{pixel}[r][s] + D(p)$, where p is the gray level of the previous pixel and the function $D()$ is some arbitrary function of the gray level. Then one change of a pixel can influence the pixels behind and a tiny change is diffused.

Finally, we adopt a hyperchaotic method as pseudorandom sequence generator. Chaotic sequences have long cycle and high complexity. By changing the

initial states of the chaotic function, large numbers of chaotic sequences with good random quality can be acquired. To increase the level of security, here we use a 4-dimensional hyperchaotic function[4] as stream cipher encryption. Hyperchaotic system is usually classified as a chaotic system with more than one positive Lyapunov exponent, indicating that the chaotic dynamics of the system is expanded in more than one direction giving rise to a more complex attractor. The adoption of hyperchaotic systems is more advantageous than using simple chaotic ones. It has enormously improved cipher anti-attack quality.

The 4-dimensional hyperchaotic function[4] is as follows. Its 3D views are shown in Fig. 3.

$$\begin{cases} \dfrac{dx}{dt} = a(y - x) + u \\ \dfrac{dy}{dt} = dx - xz + cy \\ \dfrac{dz}{dt} = xy - bz \\ \dfrac{du}{dt} = yz + ru \end{cases} \tag{3}$$

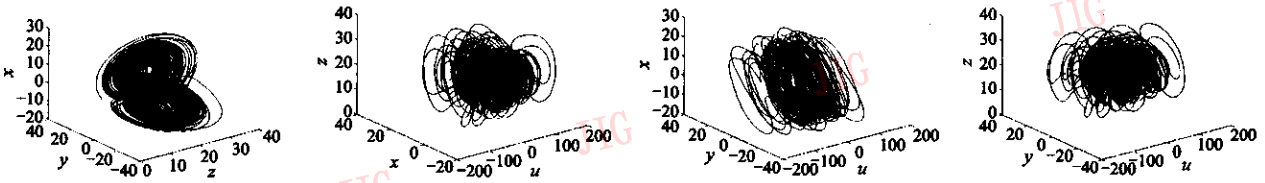


Fig. 3 3D views of system (3)

$a = 35, b = 3, c = 12, d = 7, r = 0.58$

Detailed encryption steps (see Fig. 4):

(1) Get $N \times N = N^2$ bytes of video data

(2) Make the N^2-byte data do Baker map with a key $\boldsymbol{K}_1$ according to formula (2)

(3) Encrypt the data by the 4-dimensional hyperchaotic function showed as formula (3)

If all data has been done, then finish, else go to (1).

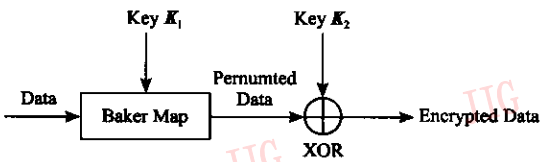


Fig. 4 The structure of the encryption algorithm

4 Security Analysis and Test Results

Comparing with other encryption algorithms, this method provides high security with fast encryption speed. Here, some security analysis results on the scheme are described, including the most important ones like key space analysis, statistical analysis, and differential analysis[5].

The testing platform is Intel PIV1. 7G, 256 RAM, operating system is Windows 2000 and program language is Visual C ++ 6. 0. Let Baker map 's key $\boldsymbol{K}_1 = \{C, n_1, \cdots, n_k, p_1, \cdots, p_l, D\}$, where C denotes iteration numbers, $n_1, \cdots, n_k$ denote the parameters of Baker map, $p_1, \cdots, p_l$ denote the parameters of function h and D denotes the parameter of diffusion function.

Parameters are set as below: $N = 64, C = 10$, $\{n_1, n_2, n_3, n_4, n_5\} = \{4, 8, 16, 32, 4\}$, $h(r, s) = rs$. Diffusion function D is defined as

$$\boldsymbol{pixel}[r][s] = \boldsymbol{pixel}[r][s] \text{ XOR } p.$$

The chaotic initial state, namely $\boldsymbol{K}_2$, is

$$\boldsymbol{K}_2 = (x_1, x_2, x_3, x_4) = (1.123\,456\,789, 2.123\,456\,789, 3.123\,456\,789, 4.123\,456\,789).$$

4.1 Key Space Analysis

(1) Baker Map 's Key

To test the key space of Baker map, we use block ciphers to encrypt an image only. Here the cipher excludes function h and D. When $C = 10$, the key number of Baker map for Known plain-text is calculated in the Tab. 1.

(2) Chaotic Function 's Key

To test the key space of the hyperchaotic function,

Tab. 1　Key number of the Baker map

	# of keys	# of clusters
N	$k(N)$	Known plain-text
64	1.8e19	2.0607e10
128	3.4e38	1.9751e21
256	1.2e77	1.8144e43
512	1.4e154	1.5312e87
1 024	1.8e308	1.0905e175

$k(N)$：the number of the keys

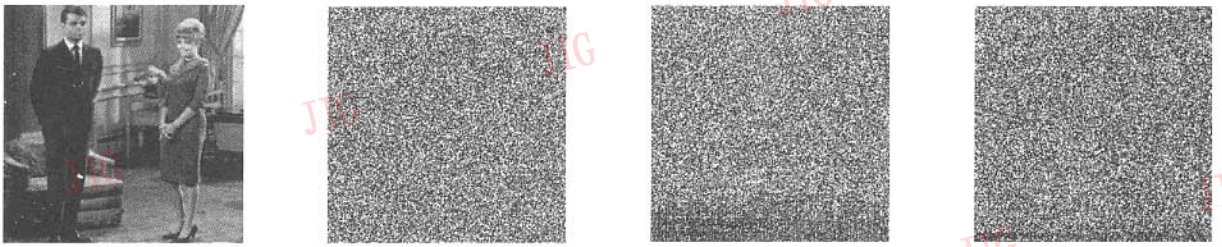
cluster：similar keys. Consult reference[3] for detailed analysis.

we use stream ciphers to encrypt an image only. First we encrypt the image with $\boldsymbol{K}_2$ to obtain the cipher-image. Then we decrypt it with different keys $\boldsymbol{K}_3$ and $\boldsymbol{K}_4$ separately. In Fig. 5 we can see that though the keys are very similar, the plain-image can't be obtained by different keys.

$$\begin{aligned}\boldsymbol{K}_2 &= (x_1, x_2, x_3, x_4)\\ &= (1.123\,456\,789, 2.123\,456\,789, 3.123\,456\,789,\\ &\quad 4.123\,456\,789).\end{aligned}$$

$$\begin{aligned}\boldsymbol{K}_3 &= (x_1, x_2, x_3, x_4)\\ &= (1.123\,456\,798, 2.123\,456\,789, 3.123\,456\,789,\\ &\quad 4.123\,456\,789).\end{aligned}$$

$$\begin{aligned}\boldsymbol{K}_4 &= (x_1, x_2, x_3, x_4)\\ &= (1.123\,456\,789, 2.123\,456\,789, 3.123\,456\,78,\\ &\quad 4.123\,456\,789).\end{aligned}$$



(a) Plain-Image　(b) Encrypt with $\boldsymbol{K}_2$　(c) Decrypt with $\boldsymbol{K}_3$　(d) Decrypt with $\boldsymbol{K}_4$

Fig. 5　Decryption with different key

As discussed above, when $C = 10, N = 64$, the computing complexity of a known plain-text attack of Baker map is $O(10^{10})$ and the key space of $\boldsymbol{K}_2$ is $O(10^{9\times4})$. So the key space of the encryption method is $O(10^{10} \times 10^{9\times4}) = O(2^{152})$. It indicates this algorithm has high security.

4.2　Statistical Analysis

Shannon suggested two methods of diffusion and confusion for frustrating the powerful statistical analysis. The above-described encryption algorithm has good confusion and diffusion properties. This is shown by tests below.

(1) Histograms of ciphered images

In Fig. 6, we give the comparison of a plain-image and its cipher-image. We can see the plain-image is encrypted to a cipher-image with uniform histogram, which implies the perfect cryptographic properties of the algorithm.

(2) Correlation of two adjacent pixels

To test the correlation between two vertically adjacent pixels and two horizontally adjacent pixels in a ciphered image, respectively, the procedure is performed as follows：First, randomly select 4096 pairs of two adjacent pixels from an image. Then, calculate their correlation coefficients using the following formulas：

$$r_{x,y} = \frac{COV(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}$$

$$COV(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y))$$

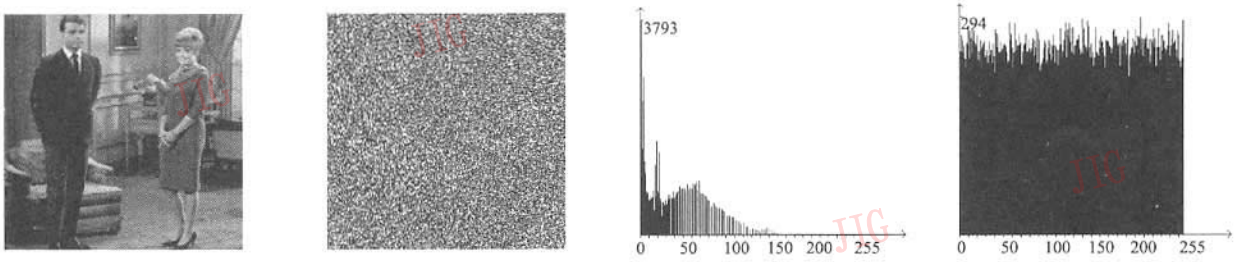
$$E(x) = \frac{1}{N}\sum_{i=1}^{N}x_i$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$$

Where x and y are gray-scale value of two adjacent pixels in the image. The comparison is shown in Fig. 7 and Tab. 2.

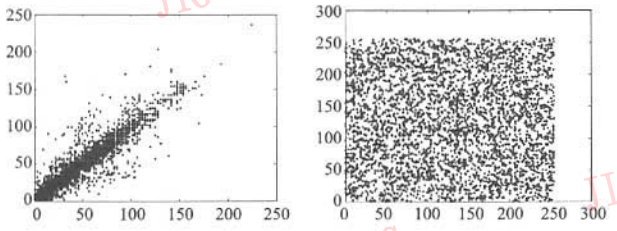
4.3　Differential Attacks

Generally, an eavesdropper may make a slight change of a plain-image in order to observe the change of the cipher-image. In this way, he may be able to find out a meaningful relationship between the plain-image and the cipher-image. This is known as the



(a) Plain-Image (b) Cipher-Image (c) Histogram of the Plain-Image (d) Histogram of the Cipher-Image

Fig. 6 Image encryption effect



(a) (b)

Fig. 7 Correlations of two horizontally adjacent pixels in the plain image and in the ciphered image

Tab. 2 Correlation coefficients of two adjacent pixels in two images

	Plain-Image	Ciphered-Image
horizontal	0.942 191	-0.020 407
vertical	0.956 797	-0.024 611

differential attack. However, if one minor change in the plain-image can cause a significant change in the cipher-image, with respect to diffusion and confusion, then the differential attack would become very inefficient and useless.

To test this property of the algorithm, we use two measures described in reference [5]: Number of pixels change rate (NPCR) and unified average changing intensity (UACI). Let two cipher-images, whose corresponding plain-images have only one pixel difference, be $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. Label the grey values of the pixels at grid (i, j) in $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$ by $C_1(i, j)$ and $C_2(i, j)$, respectively. Define a bipolar array $\boldsymbol{D}$, with the same size as image $\boldsymbol{C}_1$ or $\boldsymbol{C}_2$. Then, $D(i, j)$ is determined by $C_1(i, j)$ and $C_2(i, j)$, namely, if $C_1(i, j) = C_2(i, j)$ then $D(i, j) = 0$; otherwise, $D(i, j) = 1$. The *NPCR* is defined by

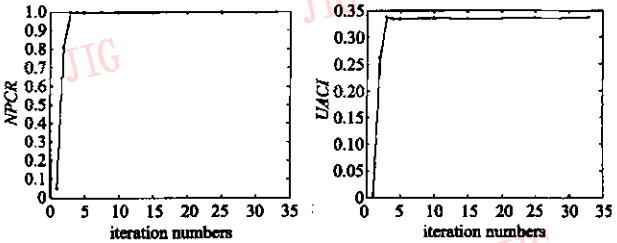
$$NPCR = \frac{\sum_{i,j} D(i, j)}{WH} \times 100\%$$

where W and H are the width and height of both $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$, and *NPCR* measures the percentage of different pixel numbers between the two images. The *UACI* is define by

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

which measures the average intensity of differences between the two images.

One performed test is on the one-pixel change influence on a 256 grey-level image of size 256×256. The test results are shown in Fig. 8. Generally, the two measures keep stable when Baker map takes 5 rounds or more.



(a) NPCR (b) UACI

Fig. 8 NPCR vs. iteration numbers and UACI vs. iteration numbers

4.4 Encryption Speed

The testing object is a 4-second long video file which is 128×128 24-bit color video sequence. We put the algorithm into MPEG2. The speed of the encryption algorithm achieves 13.6Mbps, which satisfies real-time process. The results are displayed in Tab. 3.

Tab. 3 Encryption speed

Method	Processing Time(s)
Only MPEG2	4.203
MPEG2 + Encryption	4.385

5 Conclusion

Using stream ciphers or block ciphers alone is not safe enough. So we combine both stream ciphers and block ciphers in our encryption algorithm. We adopt a 4-dimensional hyperchaotic function as stream cipher encryption, which not only enhances encryption speed, but also increases cipher complexity. Block cipher encryption is based on Baker map, which have a large key space and can be performed quickly. This method, which can be simply realized by software, provides high security for real-time digital video with fast encryption speed. Moreover it is independent of any kind of video compression algorithms. But we have not analyzed detailedly in hyperchaotic function security. And we used Runge-Kutta integration method to resolve the function, which might reduce the random quality. These problems will be solved in our future work.

References

1 Shi C, Bhargava B. A fast MPEG video encryption algorithm[A]. In: Proceedings of the sixth ACM international conference on Multimedia[C], Bristol, United Kingdom(UK), 1998:81~88.

2 Yuan Chun, Zhong Yu-zhuo, He Yu-wen. Chaos based encryption algorithm for compressed video[J]. Chinese Journal of Computers, 2004, **27**(2):257~263.

3 JIRI Fridrich. Symmetric ciphers based on two-dimensional chaotic maps[J]. International Journal of Bifurcation and Chaos, 1998, **8**(6):1259~1284.

4 Li Yu-xia, Tang Wallace K S, Chen Guan-rong. Generating hyperchaos via state feedback control[J]. International Journal of Bifurcation and Chaos, 2005, **15**(10):3367~3375.

5 Mao Yao-bin, Chen Guan-rong. Chaos-based image encryption[A]. In: E. Bayro(ed.), Handbook of Computational Geometry for Pattern Recognition, Computer Vision, Neural Computing and Robotics[M], Berlin: Springer-Verlag, 2005:231~265.