Vol 17 No 5, May 2008
1674-1056/2008/17(05)/1685-06

Chinese Physics B

© 2008 Chin. Phys. Soc.
and IOP Publishing Ltd

Hash function based on the generalized Henon map*

Zheng Fan(郑 凡), Tian Xiao-Jian(田小建)†, Li Xue-Yan(李雪妍), and Wu Bin(吴 斌)

College of Electronic Science and Engineering, Jilin University, Changchun 130012, *China*

(Received 9 October 2007; revised manuscript received 19 December 2007)

A new Hash function based on the generalized Henon map is proposed. We have obtained a binary sequence with excellent pseudo-random characteristics through improving the sequence generated by the generalized Henon map, and use it to construct Hash function. First we divide the message into groups, and then carry out the Xor operation between the ASCII value of each group and the binary sequence, the result can be used as the initial values of the next loop. Repeat the procedure until all the groups have been processed, and the final binary sequence is the Hash value. In the scheme, the initial values of the generalized Henon map are used as the secret key and the messages are mapped to Hash values with a designated length. Simulation results show that the proposed scheme has strong diffusion and confusion capability, good collision resistance, large key space, extreme sensitivity to message and secret key, and it is easy to be realized and extended.

Keywords: the generalized Henon map, Hash function, chaos
PACC: 0545

1. Introduction

With the rapid development of Internet and e-commerce technology, information security issues are attracting increasing attention. How to effectively identify authentication and ensure data security and integrity has become a hot spot. Hash function, with its basic properties such as compression, ease of computation, preimage resistance, 2nd-preimage resistance and collision resistance etc., has typically been used for data integrity and message authentication.[1] However, since the MD5 and SHA-1 and other traditional Hash functions are successfully attacked by Wang *et al*,[2,3] the construction of a new and more secure Hash function has become the focus of researches.

In recent years chaos has been studied by many researchers. The study of chaos concerns with several fields and are widely used in control, communication and laser[4] etc.. Chaos has good pseudo-random characteristics and unpredictable orbit. It is extremely sensitive to the initial states. Even if the same chaotic system runs with initial values of minute difference, the track will change dramatically soon. These characteristics can meet the requirements of Hash function, such as good diffusibility, forgery prevention, irreversibility and initial sensitivity. When the number of positive Lyapunov exponents is no less than 2, the chaotic system is defined as hyper-chaos. Because the hyper-chaotic system is more complex and changeable in chaotic behaviours, the high-dimensional hyper-chaotic system is suitable for constructing Hash functions. Recently, there has been considerable advance in the construction of Hash functions based on chaotic system.[5−17] Still, there are some disadvantages in some of them. References [5] and [6] use Logistic map and Henon map to construct Hash function, respectively. Logistic map and Henon map are both low-dimensional chaotic systems, therefore they are easy to be cracked.[18,19] Chaos-based keyed Hash function usually divides the original messages into several groups, each group having the same bit-length as the Hash value.[7−12] However, the approach was adopted by Refs.[7,8], that is, using definitive 0-1 sequences to fill the incomplete group, is not secure enough, in light of the conclusions in Ref.[13]. In this article, we use the high-dimensional discrete-time generalized Henon map to construct Hash function. This approach takes the secret key as the initial values of the generalized Henon map to iterate, and uses part of the generated sequence to fill the incomplete message group. The groups and part of the sequence interact with each other, resulting in a temporary value, which can be

*Project supported by the National Natural Science Foundation of China (Grant No 60372061).
†Corresponding author. E-mail: tianxj@jlu.edu.cn

used as the initial values of the next loop. Experimental results testify the outstanding security of the approach.

2. The generalized Henon map

Richter[20] created the generalized Henon map, which can be elaborated as follows:

$$\begin{pmatrix} x_1(j+1) \\ x_i(j+1) \end{pmatrix} = \begin{pmatrix} a - x_{w-1}(j)^2 - bx_w(j) \\ x_{i-1}(j) \end{pmatrix}, \quad (1)$$

where $i = 2, 3, \ldots, w$, $a > 0$, $b > 0$, and $x \in R^w$; w represents the dimension; a and b are control parameters.

In our another paper, we set $w = 4$, and the four initial values to 0.1, the control parameters $a = 1.76$, $b = 0.1$, then iterated the generalized Henon map for 50000 times, selected the middle 40000 numbers as test sequence, and the following conclusions were obtained:

(1) The sequence generated directly by the generalized Henon map does not fit the uniform distribution condition.

(2)A new algorithm should be designed (formula (2)) to improve the sequence, making it have a more uniform distribution.

$$X_i = \left| 10^k X_i - \mathrm{fix}(10^k X_i) \right|, \quad (2)$$

where fix$(\cdot)$ rounds the elements toward zero. Let the greatest accuracy be 10^{-r}, k donates integers between 2 and $r - \log(\mathrm{length}(X))$, while length$(\cdot)$ donates the length of the sequence.

(3)Use formula (3) to transform the improved sequence into binary sequence.

$$Y_i = \begin{cases} 0, & X_i \le \mathrm{median}(X), \\ 1, & X_i > \mathrm{median}(X), \end{cases} \quad (3)$$

where median(X) is the median of X (in statistics).

(4) A series of statistical tests testify to the excellent pseudo-random characteristics of the binary sequence.

In this article, we set $w = 20$, and all the twenty initial values to 0.1, and conduct simulation experiment following the above method. The sequence generated directly from the generalized Henon map and the sequence improved by formula (2) display their histograms in Figs.1(a) and 1(b), respectively. By comparing the two pictures, we can see that the uniform distribution characteristic is significantly improved in the new sequence. Then transform the improved sequence into binary one in accordance with formula (3). Figure 2 shows the auto-correlation function of the binary sequence; obviously, it is a more ideal δ-like function.

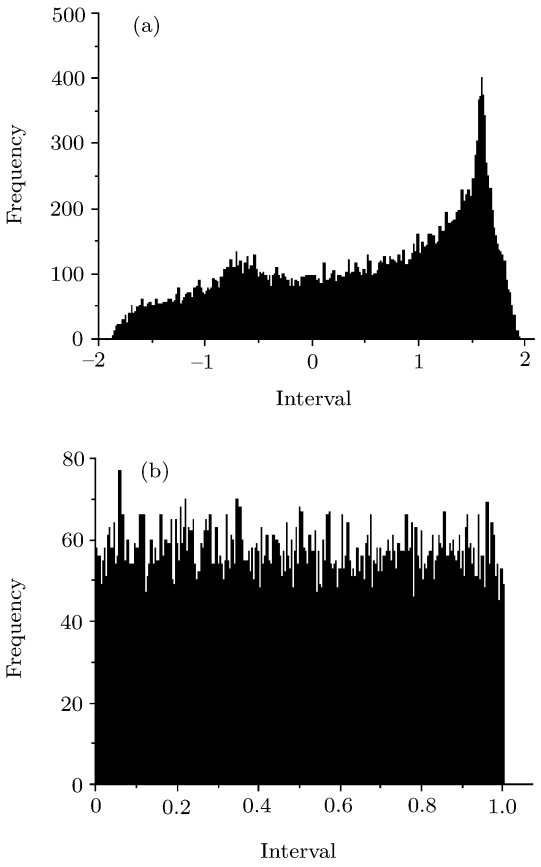


Fig.1. Histogram of chaotic sequences.

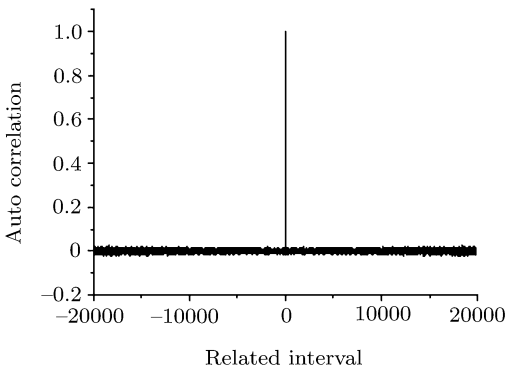


Fig.2. Auto-correlation function.

3. Realization of Hash function

Since the above binary sequence has excellent random statistic characteristics, it can be used to construct Hash function, then the message can be effectively diffused and confused. First, transform the message into ASCII, then map each ASCII into 8-bit binary, all of the binary are composed to sequence B. Suppose the final Hash value is N bit, where $N = 8 \times n$, $n = 1, 2, \ldots$. In consideration of security, we suggest that n should exceed 16. It is implied that $N \geq 128$. Define function T, and input two binary sequences $T1$ and $T2$, $T3 = T1 \oplus T2$, then map each 8 bit of $T3$ to a ASCII. Divide all the ASCII values with 256, return the values. The structure of Hash function is shown in Fig.3, which includes the following steps:

Step 1 Initialize the secret key key_0, $\text{key}_0 = \{K_i, K_i \in (0, 1), i = 1, 2, \ldots, n\}$.

Step 2 Iterate the generalized Henon map for 50000 times, where key_i are the initial values, $i = 0, 1, 2, \ldots$. Use elements between 5000 and 45000 in the generated sequence to form sequence X. Improve sequence X by formula (2), the amended sequence can be further translated into binary Y. Select N bits from the middle of Y, set them as H_i.

Step 3 Divide sequence B into groups. Each group has an N-bit binary. Set the last group as B_m, $m \geq \text{length}(B)/N$, if $\text{length}(B_m) < N$, then append it by the last $N - \text{length}(B_m)$ bits of Y. B can be signified by $B = B_1 B_2 \ldots B_m$.

Step 4 Set H_i and B_i as the input parameters of T, then calculate the next secret key key_{i+1}.

Loop Step 2 and Step 4 until all groups are processed. Finally, we obtain the final binary H_{m+1} as the Hash value.

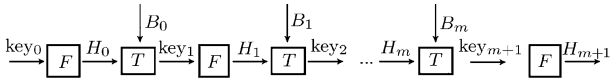


Fig.3. Hash algorithm structure.

4. Security analysis

For comparing with Ref.[8], this article has adopted text 1 in Ref.[8] as the test text: "Youth means a temperamental predominance of courage over timidity of the appetite, for adventure over the love of ease. This often exists in a man of 60 more than a boy of 20. Nobody grows old merely by a numbers of years. We grow old by deserting our ideals.". Suppose the final Hash value has 160 bits, therefore the length of the secret key is 20. Set the initial secret key $\text{key}_0 = \{K_i = 0.1, i = 1, 2, \ldots, 20\}$, the control parameters $a = 1.76$, $b = 0.1$.

4.1. Analysis of message and key

Simulation has been conducted under the following eight conditions:

C1: original text;

C2: replace initial letter Y with y;

C3: replace 20 with 10;

C4: replace the word ideals with ideal;

C5: insert a blank after the full stop of the second sentence;

C6: change the full stop of the first sentence into comma;

C7: change K_1 in secret key key_0 into $0.1+10^{-15}$;

C8: change K_2, K_4, K_6 in secret key key_0 into $0.1 - 10^{-15}$.

Compute the 0-1 sequences of the corresponding Hash values, shown in Fig.4, hexadecimal format is shown below:

C1:428ED652EF8988D339E71744108E0961438077EB
C2:BC73995F255D306E55110606B334A317C5A660CC
C3:B5CA7147717443A7EB5B24DFF1EFC7CAA8AD39E4
C4:348D061923C96DB69002B2EBE7CC6F0DA901E786
C5:A685EECB3173D09B02DC55A668F7E5F111197FE8
C6:C3225B0FCEBCE8917E84F1A349459E69C0D25170
C7:5AAD876C9421B0F2E95DBF5B7714257769144631
C8:72608EE3C3794559726BF933E78008643C1878AD

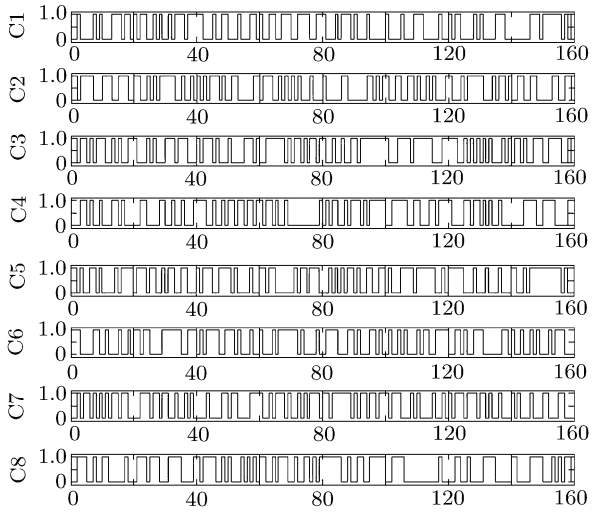


Fig.4. Comparison of Hash values.

The above data indicates that a slight change in the message will lead to a significant change in the Hash value; while under a fixed message, a slight change in the secret key will lead to a tremendous change in the result; therefore, we can say that the proposed algorithm has great sensitivity both in the message and in the secret key.

Set the length of the final Hash value to N, then the key's length should be $N/8$, namely, the secret key is composed of $N/8$ floats. Let the greatest accuracy be 10^{-r}, so the key space size is $10^{\frac{rN}{8}}$. When $r = 16$ and $N = 160$, the key space size equals to 10^{320}. It is obvious that this method produces a big enough space for the secret key to withstand attacks.

4.2. Statistic analysis of diffusion and confusion

Shannon[21] developed two major principles to deal with the message redundancy: diffusion and confusion. Hash function, like the traditional encrypted system, requires the message to diffuse its influence into the whole Hash space. This means that the correlation between the message and the corresponding Hash value should be as small as possible. The ideal Hash function should be like this: every bit of the Hash result could change with a 50% probability in response to each disturbance of the message. In order to be able to compare with Ref.[8], we adopt the same method and the same test text, then calculate the Hash value of the text. Change 1 bit at one time, and conduct four groups of tests: the first group consists of 256 tests, the second 512, the third 1024, and the fourth 2048. G_i represents the number of changed bits between the altered Hash value and the original Hash value in the ith test. The relevant changed bit probability is $P_i = G_i/160$. $G_{\max}$ and $G_{\min}$ represent the maximum and minimum changed bit number in each test group, respectively. M refers to the test times.

$\bar{G}$: mean changed bit number

$$\bar{G} = \frac{1}{M}\sum_{i=1}^{M} G_i.$$

$\bar{P}$: mean changed probability

$$\bar{P} = \frac{\bar{G}}{160} \times 100\%.$$

ΔG: standard variance of the changed bit number

$$\Delta G = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(G_i - \bar{G})^2}.$$

ΔP: standard variance of the changed probability

$$\Delta P = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(P_i - \bar{P})^2}.$$

Table 1 shows the test result. The mean changed bit number and the mean changed bit probability is 80.28 and 50.17%, respectively, which are very close to the ideal values 80 and 50%, which are better than those in Ref.[8] (79.55 and 49.72%). Figures 5(a) and 5(b) show the plot and the histogram of G_i when $M = 2048$. The changed bit number G_i gather together toward the ideal value 80, proving the advantages in such characteristics as stability, diffusion and confusion, of the Hash function under discussion. A small ΔP indicates a strong diffusion and confusion ability and a great stability of the algorithm. In Table 1, as compared with ΔP in Ref.[8], ours is smaller. In summary, our Hash function has diffusion and confusion characteristics superior to those in Ref.[8].

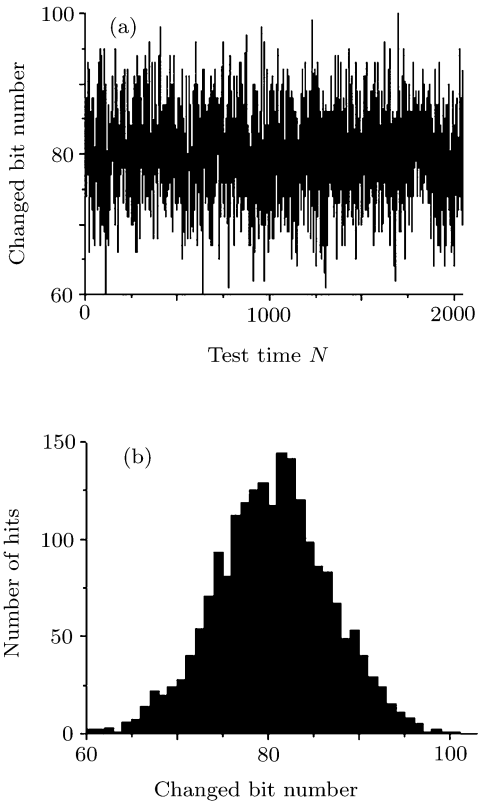


Fig.5. Distribution of changed bit number G_i.

Table 1. Statistics of number of changed bit G_i.

M	256	512	1024	2048	mean
$G_{\min}$	60	60	60	60	60
$G_{\max}$	95	98	98	100	97.75
$\bar{G}$	80.36	80.64	80.09	80.02	80.28
ΔG	6.31	6.12	6.36	6.24	6.26
$\bar{P}/\%$	50.22	50.40	50.05	50.01	50.17
$\Delta P/\%$	3.95	3.83	3.97	3.90	3.91
$\Delta P/\%$[8]	4.32	3.98	3.94	3.93	4.04

4.3. Analysis of collision resistance

Collision refers to the circumstance when a many-to-one mapping emerges between the space of the message and the space of the Hash value, in other words, different messages may share the same Hash value. Use the same test text to conduct collision test, following this method:[5,8,10] compute the Hash value of the message and transform it into decimal ASCII value; change 1 bit of the message at a time, and conduct 2048 consecutive tests, also, the altered Hash value should be transformed into ASCII. Comparing the original ASCII and the altered ASCII, we call the event that the same value appears in the same position a hit. Then count the hits. The distribution plot in Fig.6 shows the number of the same decimal ASCII values at a certain position of Hash value. In the 2048 tests, we have 147 tests hit once, and 7 tests hit twice, while in the 1894 tests, no hits occur, so we can say that the collision is at a low level, which means that the proposed Hash function holds a strong collision resistance capability.

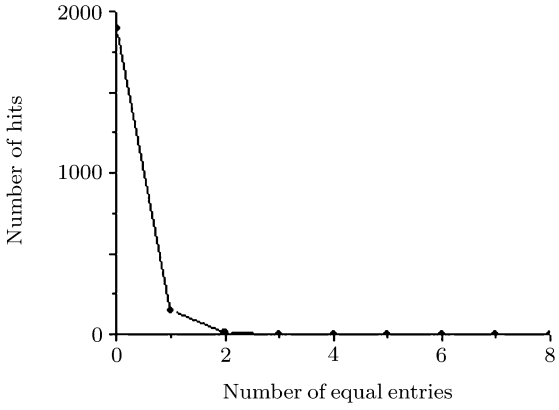


Fig.6. Distribution of the number of equal entries.

Define the absolute distance between the Hash values as

$$d = \sum_{i=1}^{N} |e_i - e_i'| , \tag{4}$$

where e_i and e_i' donate the ith decimal ASCII value of the original Hash value and the new Hash value, respectively.

Suppose the number of decimal ASCII corresponding to the Hash value is c, in our case, $c = 20$. In the test text, change "20" into "21", then repeat the experiment, an absolute distance can be obtained, which is donated by d'. Table 2 shows the maximum, minimum, mean and mean/c of d, d' and d in Ref.[8] which proposed a criterion to assess the collision characteristic of Hash function. However, by using this criterion, the mean/c of d is 83.75, greatly deviated from the ideal value 85.33 obtained in Ref.[8], and it is significantly weaker than that in Ref.[8]. While the mean/c of d' is very close to 85.33, and it is superior to that in Ref.[8]. Therefore, in our opinion, the criterion fails in all the circumstances. How to set the test conditions, and how to improve or perfect the theory to obtain an objective, generally applicable criterion for assessing the collision characteristic of Hash function, deserve further researches and discussions.

Table 2. Absolute distance between two Hash values.

	maximum	minimum	mean	mean/c
d	2524	872	1675	83.75
d'	2661	869	1715	85.73
d[8]	2520	960	1719	85.96

5. Conclusions

In this article, we have proposed an approach to construct Hash function, based on the generalized Henon map. This approach is easy to realize and has good extension characteristics: in a certain accuracy range, we can set N to any values (provided $\mathrm{mod}(N, 8) = 0$). In addition, the approach can transform messages of any length to Hash values whose length is designated by users. The experimental result shows that: (i) The Hash function constructed in our research is extremely sensitive to the message and the secret key, and possesses a large space of secret key. (ii) It has a strong, stable diffusion and confusion capability. (iii) It has an excellent collision resistance characteristic. Therefore, the proposed Hash function has great security. The less satisfactory part is the speed of operation. Since the turns of loop are proportional to the number of groups, longer messages will require longer time. It is expected that by improving the algorithm to parallel processing, the processing time can be shortened and the method will be faster and more widely applicable.

References

[1] Menezes A J, Oorschot P V and Vanstone S A 1996 *Handbook of Applied Cryptography* (New York : CRC Press) p321

[2] Wang X Y, Lai X J, Feng D G, Chen H and Yu X Y 2005 *EUROCRYPT 2005* (Berlin: Springer) p1

[3] Wang X Y and Yu H B 2005 *EUROCRYPT 2005* (Berlin: Springer) p19

[4] Fan W H, Tian X J, Chen J F, Zheng F, Yu Y L, Gao B and Luo H E 2007 *Chin. Phys.* **16** 2908

[5] Wong K W 2003 *Phys. Lett.* A **307** 292

[6] Liu J N, Xie J C and Wang P 2000 *J. Tsinghua Univ.* **40** 55 (in Chinese)

[7] Wang X M, Zhang J S and Zhang W F 2005 *Acta Phys. Sin.* **54** 5566 (in Chinese)

[8] Sheng L Y, Li G Q and Li Z W 2006 *Acta Phys. Sin.* **55** 5700 (in Chinese)

[9] Liu G J, Shan L, Dai Y W, Sun J S and Wang Z Q 2006 *Acta Phys. Sin.* **55** 5688 (in Chinese)

[10] Zhang J S, Wang X M and Zhang W F 2007 *Phys. Lett.* A **362** 439

[11] Wang X M and Zhang J S 2006 *Phys. Lett.* A **357** 323

[12] Li H D and Feng D G 2003 *Chin. J. Comput.* **26** 460 (in Chinese)

[13] Wang J Z, Wang Y L and Wang M Q 2006 *Acta Phys. Sin.* **55** 5048 (in Chinese)

[14] Wang X M, Zhang J S and Zhang W F 2003 *Acta Phys. Sin.* **52** 2737 (in Chinese)

[15] Peng F, Qiu S S and Long M 2005 *Acta Phys. Sin.* **54** 4562 (in Chinese)

[16] Zhang H, Wang X M, Li C H and Liu D H 2005 *Acta Phys. Sin.* **54** 4006 (in Chinese)

[17] Guo X F and Zhang J S 2006 *Acta Phys. Sin.* **55** 4442 (in Chinese)

[18] Short K M 1994 *Int. J. Bifurcation and Chaos* **4** 959

[19] Short K M 1996 *Int. J. Bifurcation and Chaos* **6** 367

[20] Richter H 2002 *Int. J. Bifurcation and Chaos* **12** 1371

[21] Shannon C E 1949 *Bell Syst. Tech. J.* **28** 656