

Hash Function Construction Based on RBFNN and Chaotic Mapping

Jun Chen^a, Chunxiao He^b, Pengcheng Wei^c

^aDean's Office Chongqing Education College Chongqing 400067, China

^bTeaching and Research Section of Computer Chongqing Education Management School Chongqing 400066, China

^cDepartment of Computer Science Chongqing Education College Chongqing 400067, China

Abstract

One-way Hash function is not only widely used in the aspects of the digital signature, identity authentication and integrity checking, etc. but also the research hotspot in the field of contemporary cryptography. In this paper, it firstly utilized neural network and practiced the chaotic sequences produced by one-dimensional nonlinear mapping. And then, it constructed Hash function with cipherkey by means of altering sequences. One of the advantages of this algorithm is that neural network hides the chaotic mapping relations and make it difficult to obtain mapping directly. Simulation experiment showed that the algorithm have good unidirectionality and weak collision, and stronger confidentiality than the tradition-based Hash function, as well as easy to achieve.

Index Terms: RBF neural network; Chaotic mapping; Hash function

© 2011 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science.

1. Introduction

Hash function is an aggregate transformation from the whole message aggregate to a summary aggregate with fixed-length messages, it can be divided into two categories [1,2]: the Hash function with cipherkey and without cipherkey. The Hash function without cipherkey is a key link in digital signature. It not only can greatly shorten the signature time, but also plays an important role in information integrity check and secure storage of accounts and passwords in operating system. The Hash function with cipherkey can be used for authentication, shared cipherkey and software protection, etc [2, 3]. Hash function contains three features.

- a) If given Message m and Hash function H , it is very easy to calculate the value of Hash $h = H(m)$;
- b) If given Hash value h , it is very difficult to calculate M according to $H(m) = h$, (also called unidirectionality);

* Corresponding author.

E-mail address: ^acj.cq@163.com; ^bhxcq@163.com; ^cwpc75@163.com

c) If given m , it is very difficult to find another message m' and meet $H(m') = H(m)$, (also called collision-resistance).

The traditional one-way Hash methods have several standards, such as MD2, MD5 and SHA, etc. It mostly gets Hash results based on the supplicated methods of XOR and EQV operations. It has low security and big computational burden, and difficult to find rapid and reliable encryption methods. In documents literature [4-7], it puts forward some Hash function construction operations based on chaos. However, these algorithms are design implementations based on a certain nonlinear mapping and the limitation to chaotic mapping parameter and state stimulation precision; also make chaotic sequences show the shortcomings such as short cycle, strong correlation and local linear. Therefore, the chaotic system implemented in the lower precision is not suitable to construct Hash function. Just aiming at this, the paper puts forward chaotic Hash algorithm based on neural network.

The characteristics, such as neural network nonlinear, associative memory, massively parallel distribution and high fault tolerance, are available to cryptographic communication. Only operate its parallel arithmetic method directly by integrated circuit. The paper utilized radial basis function neural network (RBFNN) to practice the known chaotic sequences, the resulting nonlinear sequences constructed Hash function with cipherkey. Through making full use of the flexibility of neural network, in the unified system structure, it can carry out that different chaotic system produce a variety of chaotic sequences by means of altering the network connection weight numbers. Meanwhile, it turns the chaotic mapping relations into implicit form, and makes it more concealed. This algorithm has high sensitivity to initial value, good unidirectionality and weak collision, and stronger confidentiality based on chaotic mapping Hash function, as well as easy to achieve.

2. The Study of RBF Neural Network versus Chaotic Sequence

2.1. RBF neural network (RBFNN)

The structure of RBF radial basis neural network is shown in **Fig.1**. It is a kind of neural network with simple structure, superior performance and the capacity of local approximation. It contains two layers: hidden layer and output layer. Assume the number of input nodes is n , the hidden layer node is m , and the output node is 1.

RBFNN Network input is:

$$X = [x_1, x_2, \dots, x_n]^T$$

Then, the network output is:

$$y_n = f(X_n) = b_0 + \sum_{i=1}^h w_i \Phi_i(X_i)$$

Select Gauss functions:

$$\Phi_i(X_n) = \exp(-\|x_n - C_i\|/2\sigma^2)$$

2.2. The Characteristics of Piecewise-nonlinear Chaotic Mapping

As shown in **Fig.2**, one-dimensional nonlinear mapping with good dynamic characteristics: $f: I \rightarrow I$, $I_i = [0,1]$, $I_i (i = 0,1, \dots, N)$ is defined as follows:[12,13]

$$F(x_{k+1}) = \begin{cases} \left(\frac{1}{I_{i+1} - I_i} + a_i \right) (x_k - a_i) - \frac{a_i}{I_{i+1} - I_i} (x_k - a_i)^2 & \text{if } x_k \in [I_i, I_{i+1}] \\ 0 & \text{if } x_k = 0.5 \\ F(x_k - 0.5) & \text{if } x_k \in (0.5, 1) \end{cases} \quad (1)$$

$$x_k \in [0, 1]$$

$$0 = I_0 < I_1 < \dots < I_i < \dots < I_{N+1} = 0.5, N \geq 2$$

$$a_j \in (-1, 0) \cup (0, 1), j = 0, 1, \dots, N$$

$$\sum_{i=0}^{N-1} (I_{i+1} - I_i) a_i = 0 \quad (2)$$

We found mapping $F(\cdot)$ has the following properties:

- i) Iterative system $x_{k+1} = F(x_k)$ ($k \geq 0$) is chaos;
- ii) Sequence $\{x_k\}_{k=1}^{\infty}$ is uniformly distributed in the interval $[0, 1]$, and distribution function $f(x) = 1$;
- iii) Sequence $\{x_k\}_{k=1}^{\infty}$ has the autocorrelation function of δ function

$$R_F(r) = \lim_{J \rightarrow \infty} \frac{1}{J} \frac{\sum_{k=1}^J x_k x_{k+r}}{\sum_{k=1}^J x_k^2}, \quad r \geq 0 \quad (3)$$

Demonstrate:

- i) Because $x \in [I_i, I_{i+1})$ and $0 \leq i \leq N$,

$$|F'(x)| = \left(\frac{1}{I_{i+1} - I_i} + a_i \right) - \frac{2a_i}{I_{i+1} - I_i} (x - I_i) > \frac{1}{I_{i+1} - I_i} - a_i > 1$$

Meanwhile, $F(x)$ is an even symmetry mapping, then, $|F'(x)| > 1$, so $x \in [0, 1]$, that is, Lyapunov index is:

$$\lambda = \lim_{J \rightarrow \infty} \frac{1}{J} \log_2 \left(\prod_{k=1}^J |F'(x_k)| \right) > 0 \quad (4)$$

According to the definition of chaotic system, Lyapunov index greater than zero implies that the iterative system is chaotic, so property(i) is tenable.

ii) Same as(i) , as $x \in [I_i, I_{i+1})$ and $0 \leq i \leq N$,

$$\begin{aligned} |F''(x)|/|F'(x)|^2 &= 2a_i(I_{i+1} - I_i)/[(1 + a_i(I_{i+1} - I_i)) - 2a_i(x - I_i)]^2 \\ &< 2a_i(I_{i+1} - I_i)/(1 - (I_{i+1} - I_i)a_i)^2 \\ &< 2/(1 - (I_{i+1} - I_i)a_i)^2 \end{aligned} \quad (5)$$

Noted that $F(x)$ is an even symmetry mapping, and get

$$|F''(x)|/|F'(x)|^2 < +\infty, x \in [0,1]$$

According to above theory,

$$P_r \circ f(x) = \sum |f(y_i)|(I_{i+1} - I_i)(1 + a_i x), \quad x \in [0,1]$$

P_r is the Frobenius-Perron operator of mapping $f(x)$, and is defined clearly in [kohda].

$$P_r \circ f(x) = \frac{d}{dy} \int_{F^{-1}([0,x])} f(x) dx \quad (6)$$

$$y_i = I_i + (I_{i+1} - I_i)(1 + a_i I_{i+1} + a_i I_i - 2a_i x)$$

According to equation (2), $f(x) = 1$ is the unique solution of the equation. So property (ii) is tenable. iii) As defined in [kohda], we rewrite the autocorrelation function,

$$\rho_F(r) = \frac{\int_0^1 x F^r(x) f(x) dx}{\int_0^1 x^2 f(x) dx} \quad (7)$$

As $r = 0$, $F^r(x) = x$, $\rho_F(r) = 1$, according to symmetry axis $x = 0.5$, as $r > 0$, x is an odd number and $F^r(x) = x$ is an even number, so property (iii) is tenable.

2.3. The RBFNN (GS-RBFNN) Produced Chaotic Sequences

RBF networks have good capacity to approach arbitrary nonlinear mapping, therefore, the network can possess chaotic state through studying on chaotic sequence and modeling. The model system structure, based on RBF neural network chaotic sequences, is shown in **Fig.3**. The database of network connection weight numbers and initial values are used to store the network connection weight numbers and the corresponding initial values trained by learning samples. The feedback from the network output end to input end shaped closed-loop structure, and make the output chaotic sequence feedback to input end as the initial value for next output

sequence, so as to output the chaotic sequences continuously. At last, it converts the simulated chaotic sequence produced by network into binary chaotic sequences.

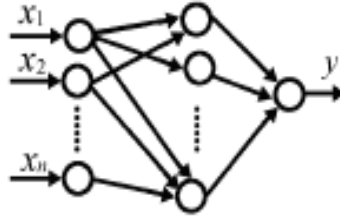


Fig. 1. RBF Neural Network

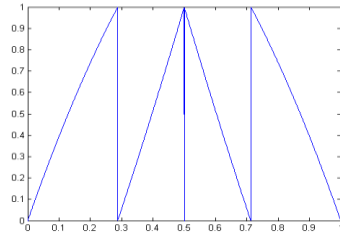


Fig. 2. Piecewise-nonlinear mapping

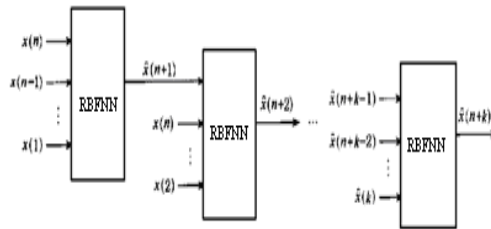


Fig. 3. RFB Neural Network Training Chaotic Sequence Model

Considering time sequence $x(1), x(2) \dots x(n)$ need predict $x(n+1)$, and then we can take the value of $x(n-k+1), \dots, x(n)$ as the input and output of neural networks $\hat{x}(n+1)$ (predictive value). The first step is to take N sets of sequences which match one-dimensional nonlinear mappings as the initial value fed into the neural network to produce new sequences, $\hat{x}(n+1), \hat{x}(n+2), \dots, \hat{x}(n+m)$, it is concluded how to use $x(1), x(2) \dots x(n)$ to predict the k step: $\hat{x}(n+k)$

We adopt recurrence method: it only predicts one step at a time, that is $\hat{x}(n+k)$, then input the predicted value which is increased continuously as the RBFNN. And then get the value for the next step, circulated in this way, and finally gets all the required values.

3. The Construction of Hash Function

The Hash function with cipherkey constructed by RBFNN is shown in **Fig.4**, described as follows:

- a) Assume the message M is a binary sequence, the length of Hash value is N ($N = 128 * i, i = 1, 2, \dots$), if the length of M isn't the integral multiples of N , it can connect the appropriate random sequence to meet requirements. Divide M into several groups according to the length of N , recorded as:

$$M = M_1, M_2, \dots, M_k \text{ and } M_i = M_i^1 M_i^2 \dots M_i^N$$

- b) Selecting random sequence N as the initialization vector H_0 , and calculated $H_0 \oplus M_1$.
- c) Take $x(0)$ and neural network parameter as a cipherkey, through CS - RBFNN training, it gets a pseudo random sequence with a length of N , then it encrypts $H_0 \oplus M_1$ and translates into a binary series to be the Hash value of M_1 , and then calculates $H_1 \oplus M_2$, encrypts $H_1 \oplus M_2$ and gets H_2 . Iterate as needed until the message ended, get the Hash value of M , H_k .

4. Simulation Experiment

4.1. The Analysis of Unidirectionality

Unidirectionality means just being able to get Hash value from the message, and it is very difficult to get original information from Hash value [8,10]. In order to describe this feature clearly, we make Hash experiment on below Plaintext 1 respectively:

Cryptologist the science of overt secret writing (cryptography), of its authorized decryption (cryptanalysis), and of the rules which are in turn intended to make that unauthorized decryption more difficult (encryption security).

In order to show the difference vividly between the plaintext and Hash value, we expressed by two-dimensional figure. Seen from ASCII distribution map of plaintext in **Fig.5**, ASCII code values are more concentrated, its code values are mainly distributed in a smaller range. Seen from the hexadecimal Hash value in **Fig.6**, operated by Hash function, the hexadecimal Hash value is very decentralized and homogenously distributed. In other words, by means of diffusion and confusion, in the Hash values, there isn't any information contained in the message (including statistical probability information of the messages). It is the Hash effect that we just desired.

4.2. Collision Analysis

The collision refers to that different initial values of Hash mapping have the same results. That is occurred many-to-one mapping [3, 5]. Select the initial text as a byte, which is 8 bit, and the corresponding value of ASCII code is from 0 to 255, the Hash result selected 8 bit, that is also from 0 to 255. In this way, the initial value space is the same as the final value space. Recording final value space to be k , which is the number of original image in initial space responding to the arbitrary value in the image space. Record the number of original images in final value space to be $n(k)$, the bigger $n(1)$, the smaller $n(0)$ and other items. It indicates that the less collision, the stronger scattering capacity of chaotic function. Use the measure ratio between the range space and domain space to quantitatively measure the collision incidence.

$$L = \frac{256 - n(0)}{56} \quad (8)$$

The closer the L value gets 1, the lower degree of collision. When equals to 1, it indicates that the collision did not occurred totally.

Based on RBF neural network and chaotic mapping, the collision distribution map of Hash function is shown in **Fig.7**. In **Fig.7**, from $n(0)$ to $n(8)$ is as follows: the 78,96,58,6,1,0, 0,0,0, $k > 8$, it all becomes 0, $P = 0.6953125$. Thus, it can be seen that the collision of the algorithm is lower. It is very difficult for the calculation of parameter L to compare with other algorithms, because the design of other algorithms mostly related to results length. If change result, it is very difficult to predict the change of Hash performance, which is equivalent to redesign algorithm. However, for the collision analysis on the application scale of 128 bit, the calculated quantities are too large and unrealistic. This algorithm can select any length as the result length, so as to make quantitative analysis on algorithm collision degree conveniently on small scale, which is just a unique advantage.

4.3. The Sensitive Dependence of Hash Value on Cipherkey

Based on RBF neural network and chaotic mapping, the Hash function construction has a sensitive dependence on cipherkey [11, 13]. Using the cipherkey with slight differences to perform Hash operation, it will produce completely different Hash value, which meets the requirements of definition exactly. In **Fig.8**, make changes on cipherkey with 10^{-3} , through Hash operation, it gets hexadecimal value distribution map. Seen from **Fig.5** and **Fig.8**, the slight change of cipherkey brings about the big changes with great probability, and gets distinctive Hash value. It can be seen that the algorithm has good unidirectionality and very high initial value sensitivity.

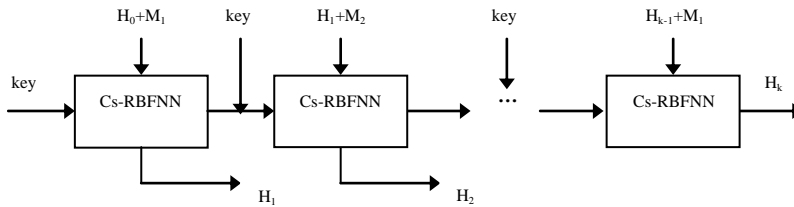


Fig. 4. Hash Algorithm Structure

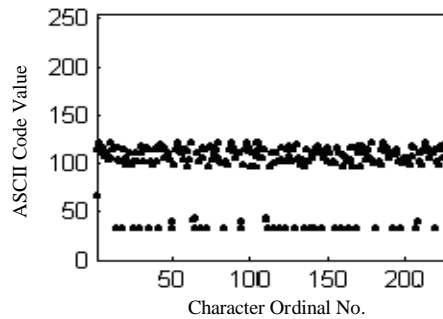


Fig. 5. ASCII Distribution Map of Plaintext

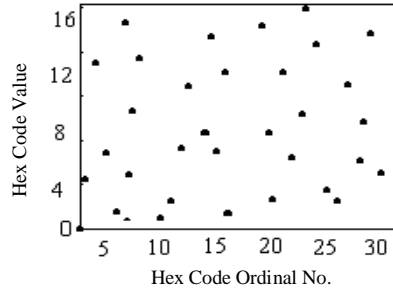


Fig. 6. Distribution Map of Hexadecimal Hash Value

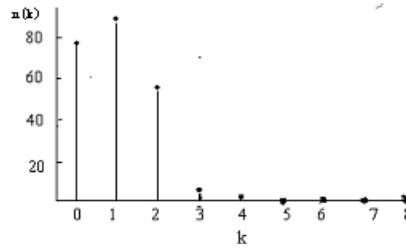
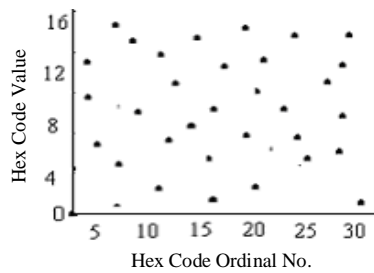
Fig. 7. The Distribution Map of $k-n(k)$ 

Fig. 8. The hexadecimal Hash Value Distribution Map as the cipherkey slightly mismatched

5. Conclusion

Based on RBF neural network and chaotic mapping, the algorithm of Hash function is simple and quick. Moreover, the algorithm utilizes the sensitivity of discrete chaotic system to the initial condition and the unidirectionality of iterative process to make each bit of Hash value related with message M , and this relation is sensitive to the slight change of message and cipherkey. Since the iterative process will increasingly enlarge the difference among initial values, and the iteration experienced the first round will make the difference become large enough to affect Hash value. Therefore, encrypting different cipherkey to the same message will get the completely different Hash value. The complex and sensitive non-linear relation between Hash values and messages can resist linear analysis effectively. Due to great key space, it can resist exhaustive attack effectively. The dynamic integration between neural network and chaotic sequence makes the attack method of chaos theory and analytical approach of traditional cryptography become difficult, so as to make Hash algorithm become highly safe.

References

- [1] H.D. Li and D.G. Feng, "Multiple discrete chaotic dynamic system and Hash function," *Chinese Journal of Computers*, Vol.26, pp.21-26, April 2003.
- [2] Hans Delfs, Helmut Knebl, "Introduction to cryptography principles and applications," Berlin: Springer-verlag, 2007.
- [3] Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud and Prashant Puniya, "Merkle-Damgård Revisited: How to Construct a Hash Function," *Lecture Notes in Computer Science*, 2005, Vol.3621, pp.430-448.
- [4] A. Swaminathan, Y.N. Mao and M. Wu, "Robust and secure image hashing," *Information Forensics and Security, IEEE Trans.*, Vol.1, pp.215-230, June 2006.
- [5] Stefan Lucks, "A Failure-Friendly Design Principle for Hash Functions," *Lecture Notes in Computer Science*, 2005, Vol.3788, pp.474-494.
- [6] Kwok-Wo Wong. A Combined Chaotic Cryptographic and Hashing Scheme[J]. *Physics Letters A* ,2003, Vol.307, pp.292–298.
- [7] X.Y. Wang, C.F. Duan and N.N. Gu, "A new chaotic cryptography based on ergodicity," *Int. J. Mod. Phys. B*, 2008, Vol.22, pp.901-908.
- [8] W. Zhang, J. Peng, H.Q. Yang, P.C. Wei. "A Digital Image Encryption Scheme Based on the Hybrid of Cellular Neural Network and Logistic[A][C] Map," *Advances in Neural Networks-ISNN2005*, Chongqing, China, May/June 2005 Proceedings, Part II, pp.860-867.
- [9] D. Xiao, X.F. Liao S.J. Deng. "One-way Hash Function Construction Based on the Chaotic Map with Changeable-parameter[J]," *Chaos Solitons & Fractals*, Vol.24, pp.65-71, January 2005.
- [10] Tao Sang, Ruli Wang and Yixun Yan. "Generating Binary Bernoulli Sequences Based on a Class of Even-Symmetric Chaotic Maps[J]," *IEEE Trans. on Communications*, Vol.49, pp.620-623, April 2001.
- [11] Guojie Hu and Zhengjin Feng, *Theoretical Design for a Class of New Chaotic Stream Cipher[J]*, *Communications Technology*, 2003, Vol.4, pp.73-74.
- [12] Shujun Li, Xuanqin Mou and Yuanlong Cai, *Improving Security of a Chaotic Encryption Approach. Physics Letters A[J]*, Vol.290, pp.127-133, April 2001.
- [13] Shujun Li, Xuanqin Mou and Yuanlong Cai, "Pseudo-random Bit Generator Based on Couple Chaotic Systems and its Application in Stream-ciphers[A][C] Cryptography," *In Progress in Cryptology-INDOCRYPT 2001*, *Lecture Notes in Computer Science*, 2001, Vol.2247, pp.316-329.