http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 10 (3): 686-690, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.686.690

Encryption Algorithm of RSH (Round Sheep Hash)

Chaoqun Liu, Yang Zhou, Yunhua Xiao and Guang Sun
Department of Information Management, Hunan University of Finance and Economics,
No. 139, Fenglin 2nd Road, Changsha 410205, China

Abstract: According to the characteristics of hash function, the one-way hash encryption algorithm based on high diophantine equation (RSH) is proposed. RSH not only can be used for password encryption, but also can be used for Data integrity check and the digital signature of message digest. These algorithms hash the arbitrary length message into 128 bits, then make the multi-iterations by high diophantine equation, finally produce the numeral string of 128 bits. In the conversion process, because we do not know the high indefinite equation iterative power law, so the resulting string of numbers is very reliable and safe.

Key words: Encryption algorithm, hash function, RSH algorithm, high diophantine equation, digital signature

INTRODUCTION

Now the common method is encrypting data by hash function. Hash function can changes the input message string of arbitrary length into the fixed-length output string. Output string is called hash value (Mironov, 2005) of input message. It generally is used to generate message digest and Key encryption. The existing hash function divides into password and the no password. Now the study is the following two:

Modification Detection Codes (MDCs): MDCs is discussed more than other no-password encryption. The common MDCs are: One-way Hash Functions (OWHFs) (Benson *et al*., 1994; Katz and Koo, 2008; Barak, 2008) and Collision Resistant Hash Functions (CRHFs) (Simon, 1998).

Message Authentication Codes (MACs) (Black, 2000): MACs is discussed more than other password encryption. For the last decades, with the rapid development of network, Many signature algorithm appeared. There are the following:

RSA: RSA algorithm is currently the best known and most widely used digital signature (Rabah, 2006).

The Digital Signature Algorithm (DSA): DSA can only be used for signature and can not be used for encryption (Hwang *et al*., 2001).

Elliptic Curve Cryptogram (ECC): Based on Elliptic Curve Digital Signature system (Toorani and Beheshti Shirazi, 2009). ECC were more research, including the algorithm optimization (Rabah, 2005a), secure application (Rabah, 2005b; Khalique *et al*., 2010), Protocol (Nikooghadam *et al*., 2008) and process optimization (Hwang and Yangz, 2002; Shen *et al*., 2006). Based on Elliptic Curve Digital Signature with the key short, fast and safe operation, etc., so In cryptography, it occupies a very important position, applications continue to expand. Another problem is that people are concerned about the security of digital signature verification (Rabah, 2005c) and application problems. In addition wavel algorithm (Chen *et al*., 2003) and digital signature agent (Khan *et al*., 2008) is relatively popular research.

When the learn and speak platform based on the campus network is developed, we need encrypt and validate the data as related to single sign-on and message digest. In the process of development, the above two Hash function can't meet the requirements of system development.

According to the research of hash function, a secure hash function should meet the following conditions at least (Rivest, 2005):

- Input string may is any length
- The length of input string is fixed. It take 128 bits at least according to the current computer technology in order to resist attacks from the network
- It is very easy that calculating hash value of the output for each given input
- Don't find that any two different input messages can hash to the same value

Hash function is mainly used for integrity check and improving the validity of digital signatures. The

Corresponding Author: Chaoqun Liu, Department of Information Management, Hunan University of Finance and Economics, No. 139, Fenglin 2nd Road, Changsha 410205, China

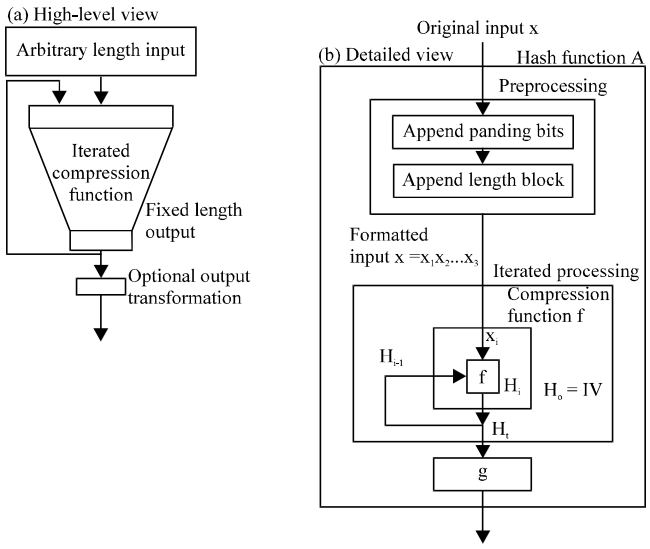

Fig. 1: General model for an iterated hash function

process of hash function is generally shown in Fig. 1(Meneze *et al*., 1997). In this iterative process the arbitrary finite-length input message X is divided into many blocks of fixed length. The preprocessing which is typically known as padding involves appending extra bits as necessary to attain an overall bit-length which is a multiple of the block length. The length of the original message before padding is also included in the last block of the padded input for security reasons. The iterative processing starts with a predefined initial value; using hash function f turn to the message digest; at the final, an optional output transformation g is often used to harden the message digest further.

Based on the existing methods and after the project team to find the information again, the based on a high diophantine equation one-way hash encryption algorithm is proposed. It will be named RSH (Round Sheep Hash) hash algorithm. Repeated practice by more than one year, RSH function be realized and has been verified in our developed system. It is used to meet the system's single sign-on password encryption and meet the request of the digital signature and verifying the file transfer process integrity when users use system.

ENCRYPTION PROCESS OF RSH HASH ALGORITHM

A designed algorithm should have the function of encrypting arbitrary length string. This algorithm can be used for integrity check of data or digital signature. Furthermore, the speed of encryption algorithm is faster. This study presents a one-way hash encryption algorithm

Fig. 2: The source string is transformed into the string

based on high diophantine equation. The length of encrypted password is 128 bits. In the process of encrypted algorithm, we will select 6 iterations as an optional mid-point. The following paper will introduce the encryption algorithm in details.

Step 1: Generate a 128 bits decimal digit from the source string. At first, the source string is transformed into ASCII decimal digit. If the encrypted source string is a12, the converted string is 974950. Figure 2 shows the general procedure.

When the lengths of number string is less than 128 bits, copy and connect the source string to the string end until the string length is greater than or equal to 128 bits, that is:

974950974950.....

The above is the case that the length of the number string is less than 128 bits. But if the length of number string is greater than or equal to 128 bits, the number string is divided into the group of 16 bits. If the last group is less than 16 bits, 0 is filled until to 16 bits.

Then retain 16 bits after each two group is added (the highest 16 bits or the lowest 16 bits, or make by turns). By this time the group numbers are reduced by half, Fig. 3

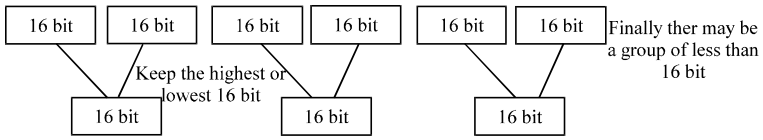

Fig. 3: Each two group of 16-bit folded into a group of 16-bit

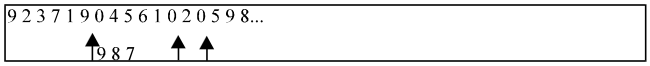

Fig. 4: Fill position 0 of the source string with the fixed series

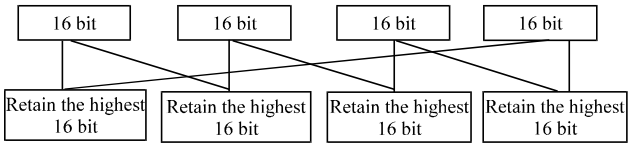

Fig. 5: The 8 groups data of 16 bits cross multiply and retain the highest(lowest) 16 bits

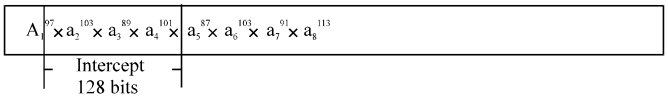

Fig. 6: Calculate and intercept the higher powers

shows the general procedure. If the total length is greater than or equal to 256, repeat these operations until the string length is less than 256.

A string that length is between 128 and 256 is left by extending or folding the string. If the string length is still greater than 128, the string is refolded. Add the front 128 bits to the back data that greater than 128 bits. The result reserve 128 bits.

Step 2: Filled the position 0 in the 128 bits with the fixed series, get a non-zero number string of 128 bits by filling. Such as: Source string is 92371904561020598…, fill position 0 of the source string with the sequence of numeric string (such as: 9876543219876543 21…), get a new string is 92371994561827598…. Fig. 4 shows the general procedure.

Step 3: Divided the string of 128 bits into 8 groups data of 16 bits (a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8). The adjacent two groups data is multiplied and retain the highest 16 bits (or retain the lowest 16 bits). The result after multiplying may overflow. Take the absolute value if the result is negative. The method of multiplication is shown in Fig. 5.

- Take the highest 16 bits from the result of $a_1 \times a_2$ into a_1' and get new a_1
- Take the highest 16 bits from the result of $a_2 \times a_3$ into a_2' and get new a_2

……

- Take the highest 16 bits from the result of $A_8 \times a_1$ into a_8' and get new a_8

Setp 4: There is still a group of 16 bits (is separately a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8). Calculate the higher powers. Abnegate the highest array component in the results after calculating (bit size is between 1 and 16). Retain 8 array components that are from 2 to 9, namely 16*8=128 bits.

If the power of the first iteration is the fixed 8 prime numbers (97, 103, 89, 101, 87, 107, 91, 113) start from the second iteration, the selected power is take the second, the fourth, the sixth, the eighth, the tenth from the group number a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8. Figure 6 shows the general procedure. This method cannot determine the power of algorithm operation. So it can increase the security of the algorithm.

Setp 5: Repeat iteration until finish six iterations.

The final cipher-text is a number string of 128 bits after computing. Namely 16128 address spaces. Conflict rate is very low.

THE PERFORMANCE TEST OF RSH ALGORITHM

Security: We can see the step 4 is the key step from the above encryption process. The calculate need 20 msec by an Inter dual-core 1.6 GHz CPU. Because there have 6 iterations, the time of violence declassification is:

6*16128/A(8,8)*20 m sec = 158548959918822932521562658 5489.6(year) (1)

We can see the declassification is unrealistic from formula (1). If we known that the text is the mixed 5 bits string of lowercase letters and numbers, the time of violence declassification is:

$$6*(26+16)^5/A(8,8)*20ms=3.4992(day) \quad (2)$$

It may reduce the computing time if we use fast exponentiation and strassen matrix multiplication. Now the step 4 needs to change into this algorithm, the numbers of iteration*20 ms ec/fast algorithm time (or add power). Then the time of declassification is still invariability. So the strength of this algorithm is still relatively high.

Complexity: The encryption time is about 100 msec that include the time of read encrypted string by test. Speed and encryption strength is a contradiction certainly. Algorithm selects 6 iterations as an optional mid-point after repeated test.

The complexity of algorithm: Set the source number string length len=2^n (Source number string length is about 2 times for the source string length, because it is 0~255 that a byte is converted to decimal digit).There remains the string of $2^{(n-1)}$ lengths after an iteration. Therefore the total addition number is approximately:

$$2^{(n-1)}+2^{(n-2)}+\ldots+2^6=2^n-2^6=\log(len)-64 \quad (3)$$

Namely it is the same magnitude to the log of the source string length. The speed is very fast. At the same time it also guarantees that the final result is related to each bit of the source string (even).

CONCLUSION

The security of the most hash function used commonly is seriously threatened as the researchers is more? On the base of studying fully the characteristics of hash function, this paper presents a fixed 128-bit output variable hash algorithm by increasing the computational complexity of iterative with high-indefinite equation and using parallel iterative structure. The realization of RSH algorithm is very simple. It has a higher complexity and more advantage of execution efficiency and wider application compared with the traditional Hash function. This algorithm can be applied to the following by the validation of actual system development:

The verification of File transfer (Bakhtiari *et al*., 1995). Calculate the RSH value of the target file and compare with the RSH value of Source file. If the two RSH value is consistency, we can assure that this two files is complete same statistically. So this algorithm can not only test whether produce error but also ensure the file is not been tampered maliciously in the process of file transfer.

The digital signature (Stallings, 2003; Sun and Sun, 2010). Signer first calculates RSH hash value of this data file and makes the digital signature by asymmetric algorithms. When other sides verify the signature, he first calculates RSH hash value of this data file and makes the digital signature by asymmetric algorithms too.

Password Encryption. Calculate the RSH value of password and store in the database. When user use password, he first calculate the RSH value of password. Then compare with the RSH value stored in the database. If the value is same, the password is correct. Otherwise incorrect. This method can also be used for authentication protocols

Make 6 iterations for RSH algorithm; we can further improve the security of the algorithm by increasing the number of iterations in practice. We can also improve the fixed prime power of iteration, the power of iteration extracts from 128 bits string before iteration. This method can ensure the randomness and unpredictability of iterative power. Hash arbitrarily long message into a fixed 128-bit hash value; change the message hash value or any length message hash value that can be set to 128, 160, 192, 224 or 256-bit.

ACKNOWLEDGMENTS

This Study was supported by Hunan Provincial Education Department of China (XingJiaoTong (2009) Document No. 320).

REFERENCES

Bakhtiari, S., R. Safavi-Naini and J. Pieprzyk, 1995. Cryptographic hash functions: A survey. Technical Report 95-09, Department of Computer Science, University of Wollongong, http://citeseerx. ist. psu.edu/viewdoc/summary?doi=10.1.1.56.8428.

Barak, B., 2008. Lecture notes 4: UOWHF from any one-way function. http:// www.cs. princeton. edu/ courses /archive/spring08/cos598D/scribe4.pdf.

Benson, T.A., L. Gong and T.M.A. Lomas, 1994. Lomas, secure, keyed and collisionful hash functions. Technical Report SRI-CSL-94-08.

Black, J.R., 2000. Message authentication codes. Ph.D. Thesis, University of California

Chen, T.H., G. Horng and S.H. Wang, 2003. A robust wavelet-based watermarking scheme using quantization and human visual system model. Inform. Technol. J., 2: 213-230.

Hwang, M.S. and W.P. Yangz, 2002. Integrating different semantics of classification levels in heterogeneous distributed database system. J. Applied Sci., 2: 553-557.

Hwang, M.S., C.C. Lee and E.J. Lu, 2001. Cryptanalysis of the batch verifying multiple DSA-type digital signatures. Pak. J. Applied Sci., 1: 287-288.

Katz, J. and C.Y. Koo, 2008. On constructing universal one-way hash functions from arbitrary one-way functions. J. Cryptol.,

Khalique, A., K. Singh and S. Sood, 2010. Implementation of elliptic curve digital signature algorithms. Int. J. Comput. Appl., 2: 21-27.

Khan, A., X. Niu and Z. Yong, 2008. A robust framework for protecting computation results of mobile agents. Inform. Technol. J., 7: 24-31.

Meneze, A.J., P.C. van Oorschot and S.A. Vanstone, 1997. Handbook of Applied Cryptography, Discrete Mathematics and its Applications. CRC Press, USA., pp: 321-383.

Mironov, I., 2005. Hash functions: Theory, attacks and applications. http://research. microsoft.com/en-us/ people/mironov/hash_survey.pdf.

Nikooghadam, M., M.R. Bonyadi, E. Malekian and A. Zakerolhosseini, 2008. A protocol for digital signature based on the elliptic curve discrete logarithm problem. J. Applied Sci., 8: 1919-1925.

Rabah, K., 2005a. Elliptic curve elgamal encryption and signature schemes. Inform. Technol. J., 4: 299-306.

Rabah, K., 2005b. Secure implementation of message digest, authentication and digital signature. Inform. Technol. J., 4: 204-221.

Rabah, K., 2005c. Security of the cryptographic protocols based on discrete logarithm problem. J. Applied Sci., 5: 1692-1712.

Rabah, K., 2006. Implementing secure RSA cryptosystems using your own cryptographic JCE provider. J. Applied Sci., 6: 482-510.

Rivest, R.L., 2005. Abelian square-free dithering for iterated hash functions. http://people .csail. mit. edu/ rivest/ Rivest-Abelian Square Free Dithering For Iterated Hash Functions.pdf.

Shen, J.J., C.Y. Lin and H.W. Yang, 2006. Improvement of fixing problems in Lin *et al*. OSPA Protocol. J. Applied Sci., 6: 639-643.

Simon, D.R., 1998. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions. Lecture Notes Comput. Sci., 1403: 334-345.

Stallings, W., 2003. Cryptography and Network Security: Principles and Practice. 3rd Edn., Prentice Hall, New Jersey, USA.

Sun, G. and X. Sun, 2010. Software wate marking based on condensed co-change graph cluster. Inform. Technol. J., 9: 949-955.

Toorani, M. and A.A. Beheshti Shirazi, 2009. An elliptic curve-based signcryption scheme with forward secrecy. J. Applied Sci., 9: 1025-1035.