A New One-Way Cryptography Hash Function For WSN

Vivekananda Journal of Research
July- Dec 2019, Vol. 8, Issue 2, 149-157
ISSN 2319-8702(Print)
ISSN 2456-7574(Online)

http://www.vips.edu/vjr.php
योग: कर्मसु कौशलम्
IN PURSUIT OF PERFECTION

Pooja*
R.K.Chauhan**

Abstract

Hash functions are used to protect the integrity and authentication of information. The most popular cryptography hashing techniques are MD5, SHA-1 and HMAC. Due to the recent attacks on these standard techniques, there is a huge demand for new hashing functions to secure the information. This paper discusses a novel one-way cryptographic hash algorithm which takes input of any length and produces an output of fixed length, called digest. The proposed technique satisfies the basic properties of hash function like pre-image resistance, second pre-image resistant and collision resistance.

Keywords: *Hashing, SHA1, MD5, Security*

Introduction

Cryptography hash functions are the fundamental building blocks for information security and have plenty of security applications to protect the data integrity and authentication like digital signature schemes, construction of Message Authentication Codes and random number generators. Hash functions should follow the basic properties like one-way (pre-image resistance), second pre-image resistance, collision resistant and avalanche effect; and these are expected to be preserved. The recent study shows that MD and SHA family cryptography hash functions are vulnerable to security flaws which leads to the designing of more secure hashing function. This paper focuses on developing a novel one-way cryptography hash function, that accepts input of arbitrary length and generates a fixed size

* Ph.D Scholar, Department of Computer Science and Applications, Kurukshetra University, Kurukshetra. E-mail: poojasingh59@ymail.com

** Professor, Department of Computer Science and Applications, Kurukshetra University, Kurukshetra

output satisying all the basic properties of cryptography hash.

Attacks on Cryptography Hash Functions

Several attacks have been launched by an attacker on the hash functions to break the integrity of a message. The attacks are classified into two major categories and these are:

1. Brute force attack: It works on all hash functions irrespective of their internal construction and any other functioning. The Brute force relates to trying all the possible combinations of keys to launch an attack. *Birthday attack?* [Bellare and Kohno 2004] is the most common example of this attack.

2. Cryptanalytical Attacks: It focuses on the structure of hashing function. It is further divided into two categories [Gauravaram 2003]: Generic attacks and Specific attacks.

Generic Attacks work on the general hash function construction and major examples of this type of attacks are: Length extension attacks, Jouxmulti collision attacks, Herding Attacks and Meet in the Middle Attacks [Gauravaram 2003].

Specific attacks work on the compression function's algorithm and examples of this attack are: differential cryptanalysis, linear cryptanalysis, rotational cryptanalysis and attacks on the underlying encryption algorithms [Gauravaram 2003].

Rest of the paper is organized as follows: in section 3, literature survey of proposing new cryptography hash technique is illustrated. In section 4, proposed hashing technique is explained with implementation and results of algorithm. Lastly in section 5, conclusion and future directions are discussed.

Literature Survey

Authors of paper [Chowdhury et al.2014] developed a light-weight and one-way cryptography hash algorithm, which they named LOCHA. The algorithm was primarily designed in such a way that it is useful for energy starved wireless networks by consuming low energy in transmission. Their algorithm produced a digest of 96 bits for any length of input string and satisfied all the main properties of hashing, that is, one-way; second pre-image resistance and collision resistance.

Authors of paper [Arya et al.2013] proposed a novel hashing scheme that incorporated the usage of key so that intruder cannot break the hash code without the help of the key. Their algorithm produced a digest of 128 bits for an arbitrary message length. They have divided it in two phases: pre-processing and hash calculation.

Authors of paper [Abutaha and Hamamreh2013] designed a one way hash algorithm that used two steps, wherein, they have arranged the input data in a non-invertible matrix by using conversions and generated the initial hash initially. Secondly, they used an output of previous step and added a salt value to it, which they have sent to the receiver. They have compared their proposed work with MD5, SHA-1 and SHA-512.

Authors of paper [Raouf et al.2013] given a Pizer hash technique that was based on two methods, elliptic curve and expander graph. After computing the hash with the help of Pizer function, they signed a message by ECDSA. They used MATLAB to simulate their results and proved that their work was collision resistant.

Authors of paper [Mirvaziri et al.2007] presented a hybrid cryptography hash function that was the combination of SHA-1 and MD5. The compression function used in the proposed hash function was made from the four rounds of encryption function; and each and every round consisted of 20 transformation steps.

Authors of paper [Moe and Win2017] developed a new honeyword generation technique that stored the users' password as honeywords to decrease the typo safety problem, storage overhead and some other drawbacks which were existing in older honeyword generation methods. They have used special hashing technique to store the passwords and honeywords in database which reduced the time complexity of an algorithm.

Authors of paper [Rubayya and Resmi2014] designed a new technique that was based on HMAC/SHA-2. The results were analysed and compared using various design goals and strategies like Balanced and Area Reduction. They proved that their proposed work utilized less area and consumed less power.

Authors of paper [Tiwari and Asawa2012] developed a dedicated cryptography hash technique called MNF-256 that was based on the concept of NewFork-256. Their design used three branch parallel structure and each branch consisted of eight operations. The results and rigorous analysis claimed that proposed work was robust against cryptanalytic attacks and faster than NewFORK-256.

Authors of paper [Mondal and Mitra 2016] discussed the Timestamp defined hash function called TDHA for the secure transmission among vehicles. In this technique, sender vehicle transmitted deformed message and incomplete message digest and receiver vehicle produced digest from the intermediate digest and distorted form of message. They have simulated their algorithm and compared with MD5, SHA-1 and LOCHA using comparison metrics like overhead (communication and computation) and storage overhead. Their design outperform the other techniques both qualitatively and quantitatively.

Authors of paper [Chen and Wang 2008] developed an enhanced algorithm that followed the fundamentals of Store-Hash and Rehash for context triggered piecewise hashing technique (CTPH) also known as FKsum. They have compared it with spamsum and results proved that the performance and ability of the proposed method are better. The new design was valuable for forensics practice.

Authors of paper [Rasjid et.al. 2017] surveyed several type of attacks on hash algorithms. They reviewed existing and present methods which are used in digital forensic tools to create an attacks. They have compared the common features of MD series and SHA series in terms of output size, number of round, collision found and performance (MiB/s).

Proposed Work

Hash function consists of two components: first one is compression function, a mapping function, used to transform a large input string into a small output; and second component is construction, a method by which the compression function is being repeatedly called to process a variable-length message.

Design of Proposed Hashing:

Step 1: *Padding*- Perform padding in such a way that the length of message after padding is congruent to 448 modulo 512. In padding first bit is 1 and rest of the bits are 0.

Step 2: *Appending*- After padding, append the length of original message (represented in binary form) to the output of the step 1. After doing this step, the length of message is now in multiples of 512.

Step 3: *Message into Blocks*- The message obtained from the previous step is divided into n Blocks of size 512 bits. That is:

B0, B1, B2, B3, B4, B5, B6, B7,………… Bn

Step 4: *Blocks into Sub Blocks*- Each Block which is obtained in previous block is of bigger size. In this step, we have divided the blocks into sub blocks of size 128 bits. That is:

(B01, B02, B03, B04), (B11, B12, B13, B14), (B21, B22, B23, B24),……………. (Bn1, Bn2, Bn3, Bn4)

Step 5: *Apply XOR operation*- Next step is to perform XOR among each part of the block and override the block's value with its output. As the size of sub block is 128 bits, after performing XOR the overridden block's size is 128 bits. That is:

B0 = B01 XOR B02 XOR B03 XOR B04

B1 = B11 XOR B12 XOR B13 XOR B14

B2 = B21 XOR B22 XOR B23 XOR B24

………… and so on

Step 6: *Overall XOR*- The size of blocks which are produced from the previous step is 128 bits. In this final step, apply XOR among all the blocks. This will result into message digest of size 128 bits.

output = B0 XOR B1 XOR B2 XOR B3

Algorithm

Following is the Algorithm of our proposed hashing technique.

Input: Message of arbitrary length.

Output: Digest of fixed size, i.e., 128 bits.

```
Begin
1. Apply padding of $10^*$ bits.
2. Append length of message (binary form) to the previous step output, so
   that length of final message is in the multiples of 512.
3. $i=0, j=-1$
4. while ($i<n$) {
    $j=j+1$
    $Bj=M(i,i+511)$
    $i=i+512$ }
$i=0$
while ($i \le j$) {
    $Bi1=Bi(0,127)$
    $Bi2=Bi(128,255)$
    $Bi3=Bi(256, 383)$
    $Bi4=Bi(384,511)$
    $i= i+1$}
$i=-1$
while ($i < j$ ) {
    $i = i+1$
    Bi=Bi1 XOR Bi2 XOR Bi3 XOR Bi4}
digest = B0 XOR B1 XOR B2 XOR ………. Bj
End
```

Results and Comparisons

The proposed hashing technique is compared with MD5 and SHA-1 in terms of execution time and size of digest produced, as shown in table 1. The size of message digest in MD5, SHA-1 and Proposed is 128 bits, 160 bits and 128 bits respectively. From the table 1, it is analysed that execution time of proposed hashing is less as compared to that of others.

Table 1: Comparison of proposed hashing with standard hashing

Parameters	Execution Time (ms)	Message Digest (Bits)
MD5	310	128
SHA-1	398	160
Proposed	297	128

In figure 1, comparison of standard hashing algorithms and proposed algorithm in terms of their execution time is shown. On x-axis, hashing techniques are displayed and on y-axis, execution time is given. Figure 1 shows that proposed hashing has less execution time as compared to others.

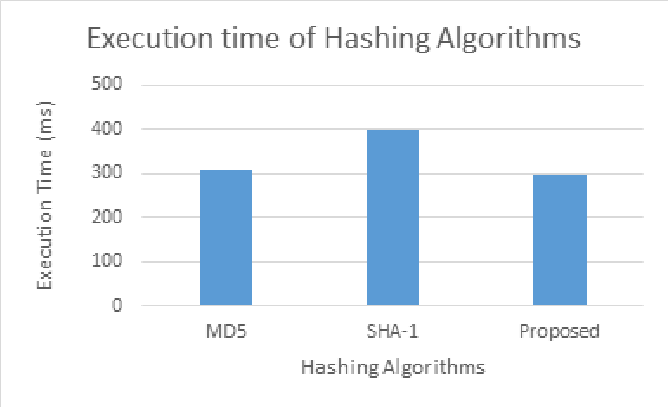

Figure 1: Comparison of Execution time of Hashing Algorithms

We have also simulated proposed technique with different file sizes (1000, 5000, 10,000, 20,000, 30,000, 40,000 and 50,000 bytes) to analyse the time if we increase the file size and it is shown in Table 2.

Table 2: Comparison of Execution time of proposed technique using different file sizes

File Size	Execution time (ms)
1000	297
5000	410
10,000	580
20,000	830
30,000	990
40,000	1210
50,000	1480

Conclusion and Future Scope

The present algorithms for calculating message digests are susceptible to brute force attacks. In this paper, a new hashing technique has been proposed which uses blocks subgrouping and XOR operation in a unique way, so it is impossible for the intruder to break the algorithm. Therefore, proposed hashing technique is more secure than the existing ones. Secondly, it uses very simple mathematical operation which increases the speed of computation of message digest. Therefore, proposed hashing technique is simpler than the existing one.

Though the proposed algorithm satisfies the basic properties of hashing function, we are trying to modify the proposed algorithm in future so that it will be resistant to other known security attacks.

References

Abutaha, M., & Hamamreh, R. (2013, January). "New One Way Hash Algorithm Using Non-Invertible Matrix". Paper presented at International Conference on Computer Medical Applications (ICCMA), Sousse, Tunisia.

Arya, R. P., Mishra, U., & Bansal, A. (2013)." Design and Analysis of a New Hash Algorithm with key Integration ". *International Journal of Computer Applications*, 81(1), 33–38.

Bellare, M., & Kohno, T. (2004). *Hash Function Balance and Its Impact on Birthday Attacks*. Advances in Cryptology - EUROCRYPT, 3027, 401-418.

Chen, L., & Wang, G. (2008, January). "An Efficient Piecewise Hashing Method for Computer Forensics". Paper presented at First International Workshop on Knowledge Discovery and Data Mining, Adelaide, SA, Australia.

Chowdhury, A. R., Chatterjee, T., & Dasbit, S. (2014, December). LOCHA: "A Light-Weight One-Way Cryptographic Hash Algorithm For Wireless Sensor Network". Paper presented at 5th International Conference on Ambient Systems, Networks and Technologies, Procedia Computer Science, India.

Gauravram, P. (2003). *Cryptographic Hash Functions: Cryptanalysis, design and Applications*. Ph.D. thesis, Faculty of Information Technology, Queensland University of Technology, Brisbane, Australia.

Mirvaziri, H., Jumari, K., Ismail, M., & Hanapi, Z. M. (2007, December)." A New Hash Function Based On Combination Of Existing Digest Algorithms". Paper presented at 5th Student Conference on Research and Development, Malaysia.

Moe, K. S. M., & Win, T. (2017, October)." Improved Hashing and Honey-Based Stronger Password Prevention Against Brute Force Attack ". Paper presented at International Symposium on Electronics and Smart Devices (ISESD), Yogyakarta, Indonesia.

Monadal, A., & Mitra, S. (2016, December)." TDHA: A Timestamp Defined Hash Algorithm for Secure Data Dissemination in VANET ". Paper presented at International Conference on Computational Modelling and Security, Procedia Computer Science, India.

Manel, D., Raouf, O., Ramzi, H., & Mtibaa, A. (2013, December)." Hash Function and Digital Signature Based on Elliptic Curve. Paper presented at International Conference on Sciences and Techniques of Automatic Control Computer Engineering, Sousse, Tunisia.

Rasjid, Z. E., Soewito, B., Witjaksono, G., & Abdurachman, E. (2017, December). A review of collisions in cryptographic hash function used in digital forensic tools. 2nd International Conference on Computer Science and Computational Intelligence, Jakarta, Indonesia.

Rubayya, R. S., & Resmi, R. (2014, December). Memory optimization of HMAC/ SHA-2 encryption. In 2014 First International Conference on Computational Systems and Communications (ICCSC), Trivandrum, India.

Tiwari, H., & Asawa, K. (2012)." A Secure and efficient Cryptographic Hash Function Based On NewFORK-256". *Egyptian Informatics Journal*, 13 (3), 199–208.