# A Survey of Cryptographic Hash Algorithms and Issues

Sandhya Verma* M. Tech Scholar, G. S. Prajapati# Head of Department,
Dept of CSE, VNS Group of Institutions Bhopal, Madhya Pradesh, India
sandhya.verma44@gmail.com*, gsprajapati1234@gmail.com#

***Abstract*—Now a day's data is transmitted between users through internet in the form of multimedia. The data transmission requires less time complexity as well as security of data, security is the big issue during transmission of data. Data integrity is one of the major parameter to ensure the security. There are many algorithms or methods have been developed for maintain the integrity of data during transmission, i.e. if the message has been changed after transmission from sender and before it may be received by the corresponding receiver, can be traced by the receiver, and thus, such a modified message can be discarded. There are many Hash algorithms proposed to provide the integrity but almost all the algorithms have proven breakable or less secure. As per the analysis of existing Hash algorithms we implemented these algorithms and compare the results on the basis of time complexity and bit difference effect.**

***Index Terms* — Message Digest, Secure Hash Algorithm, Hash Function, Message Integrity, Time Complexity, Cryptography**

## I. INTRODUCTION

Cryptographic Hash Functions are important building blocks in computer security. They provide message Integrity that is a surety that the receiver is receiving the same message that was sent by sender, and has not been modified by any attacker during the transmission of message. Hash functions are the mathematical functions that take arbitrary length input and produce a small output of fixed size [2]. This output is known as message digest or hash code or hash result or simply hash. Hash function generates a fixed size message digest of a given message; this message digest is treated as a signature of that message. Hash function has the following properties [9].

1. It is a one way function means it is easy to calculate MD from M but it is impossible to calculate M from MD.
2. It should be difficult to find to such messages M1 and M2 which generates same message digest i.e. $H_F(M1) \neq H_F(M2)$.

A cryptographic hash function is used to ensure the integrity of the transmitted data or stored data. Sometimes it is also called digest of a message [3].

There are many algorithms designed to implement the hash function. MD-2, MD-4, MD-5, SHA-0, SHA-1 and SHA-2, RIPMID, HAVAL etc, are the best known algorithms for message digest. After this many researchers have also proposed their own algorithms for the same such as SHA-192. In the next section authors have discussed all these algorithms in detail.

## II. HASH FUNCTIONS FAMILIES

In this section we discuss the detail description of existing cryptographic hash families.

### A. *Message Digest 2*

It was developed by Ronald Rivest in 1989. The algorithm is optimized for 8-bit computers. Although MD2 is no longer considered secure, even as of 2014, it remains in use in public key infrastructures. MD2 takes 8 bit as an input and produced 128 bits digest. It takes 18 rounds of its compression function to generate a 128 bit digest.

In 2004, MD2 was shown to be vulnerable to a preimage attack with time complexity equivalent to 2104 applications of the compression function (Muller, 2004). The author of MD2 concludes, "MD2 can no longer be considered a secure one-way hash function".

In 2008, MD2 has further improvements on a preimage attack with time complexity of 273 compression function evaluations. In 2009, MD2 was shown to be vulnerable to a collision attack with time complexity of 263.3 compression function evaluations [8].

### B. *Message Digest 4*

The MD4 Message-Digest Algorithm is a cryptographic hash function developed by Ronald Rivest in 1990. The digest length is 128 bits.

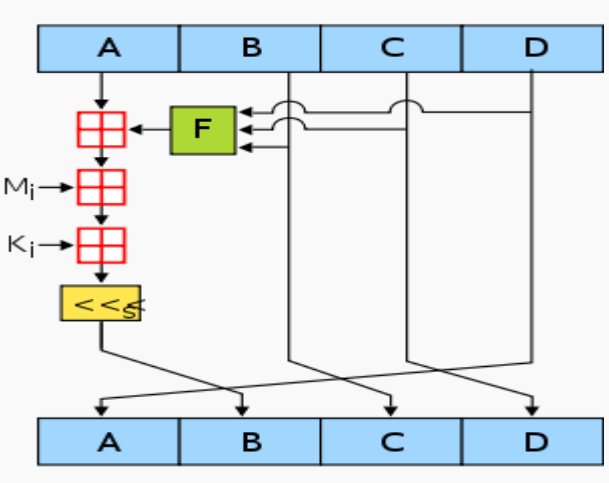


Figure 1: Operations of MD 4

MD4 consists of 48 of these operations, grouped in three rounds of 16 operations' is a nonlinear function; one function is used in each round. $M_i$ denotes a 32-bit

 

block of the message input, and $K_i$ denotes a 32-bit constant, different for each operation.

The security of MD4 has been severely compromised. The first full collision attack against MD4 was published in 1995 and several newer attacks have been published since then. As of 2007, an attack can generate collisions in less than 2 MD4 hash operations. A theoretical preimage attack also exists [4].

### C. *Message Digest 5*

MD5 was designed by Ronald Rivest in 1991 to replace an earlier hash function, MD4. The message digest algorithm is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number. MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity.

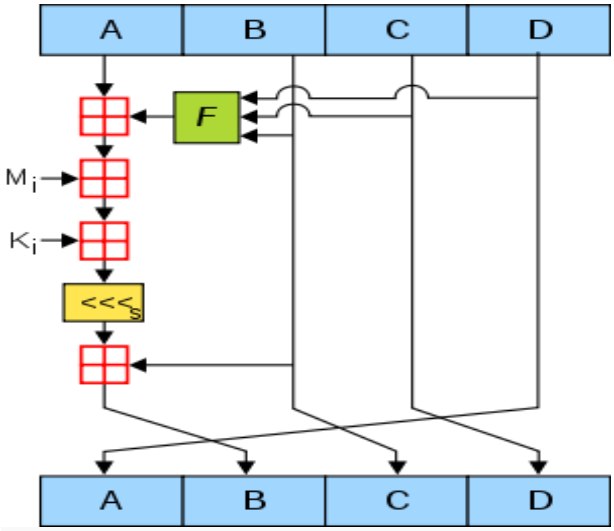


Figure 2: Operations of MD 5

In 2004 it was shown that MD5 is not collision resistant. As such, MD5 is not suitable for applications like SSL certificates or digital signatures that rely on this property for digital security. Also in 2004 more serious flaws were discovered in MD5, making further use of the algorithm for security purposes questionable, specifically, a group of researchers described how to create a pair of files that share the same MD5 checksum. Further advances were made in breaking MD5 in 2005, 2006, and 2007.

### D. *Secure Hash Algorithm 0*

The Secure Hash Algorithm is cryptographic hash functions published by the National Institute of Standards and Technology (NIST) as a U.S. Federal Information Processing Standard (FIPS), SHA-0 belongs from SHA family; it is another cryptographic hash algorithm generates a message digest of fixed 160 bits. It takes 80 rounds. In 2004, cryptanalysts attack has been found by Bihamet. Al breaks SHA-0 collision resistance at $2^{41}$.

### E. *Secure Hash Function 1*

In cryptography, SHA-1 is a cryptographic hash function designed by the United States National Security Agency and is a U.S. Federal Information Processing Standard published by the United States NIST. SHA-1 produces a 160-bit (20-byte) hash value known as a message digest. A SHA-1 hash value is typically rendered as a hexadecimal number, 40 digits long.

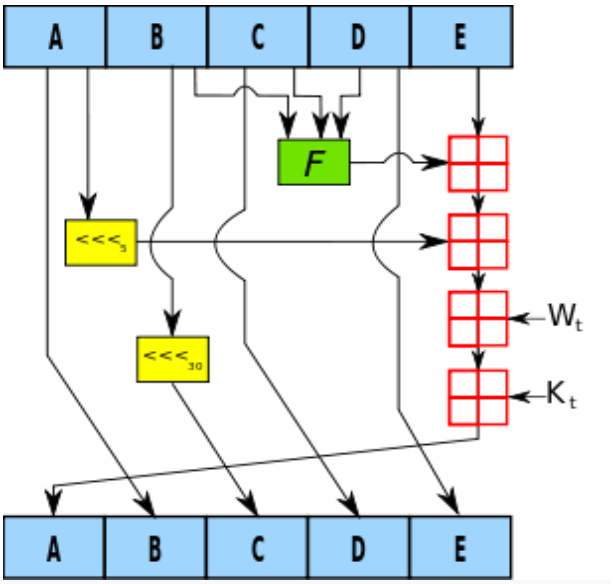


Figure 3: Operations of SHA 1

SHA-1 generates a message digest of 160 bits. It takes 80 rounds and was published in 1995. It is the most widely used algorithm for integrity. Reason for its popularity among existing algorithms is its time efficiency and its robustness.

In 2005, cryptanalysts found attacks on SHA-1 suggesting that the algorithm might not be secure enough for ongoing use. Later on, a 2011 attack by Marc Stevens can produce hash collisions with a complexity of $2^{61}$ operations.

### F. *Secure Hash Algorithm 2*

SHA 2 is a set of cryptographic hash functions designed by the NSA (U.S. National Security Agency). SHA stands for Secure Hash Algorithm. Cryptographic hash functions are mathematical operations run on digital data; by comparing the computed "hash" (the output from execution of the algorithm) to a known and expected hash value, a person can determine the data's integrity [6,7].

For example, computing the hash of a downloaded file and comparing the result to a previously published hash result can show whether the download has been modified or tampered with. A key aspect of cryptographic hash functions is their collision resistance: nobody should be able to find two different input values that result in the same hash output.

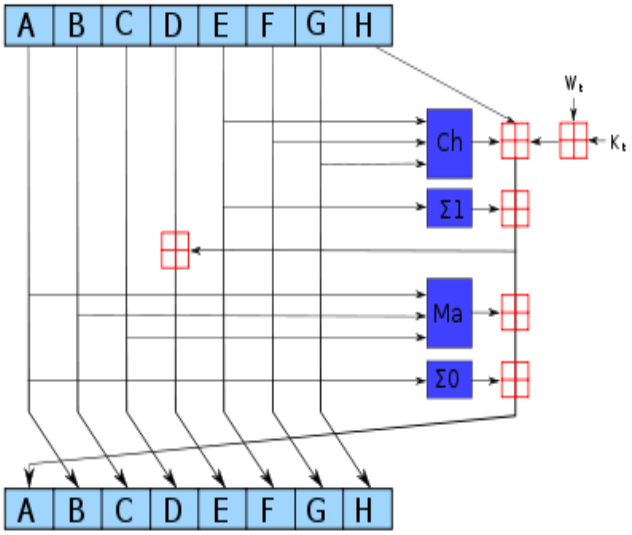


Figure 3: Operations of SHA 2

SHA-2 is a collection of different hash functions i.e. SHA-224, SHA-256, SHA-384 and SHA-512 [7]. None of them have proven completely breakable but still these algorithms are not preferred to ensure the integrity because they are not time efficient as SHA-1.
It is found that, none of the hash algorithm is secure to ensure the integrity except SHA-2 but it is found that it is not time efficient. Many researchers have found these problems and proposed their own algorithms as a solution.

### *G. Secure Hash Function 192*

SHA 192 is another hash algorithm proposed in 2013. In this authors have proposed a new compression function to generate a message digest of 192 bits [5]. Authors have combined the compression function of MD-5 and SHA-192 and take 64 rounds of compression function for each 512 bits message block [1]. Compression function of SHA-192 is shown in Figure.

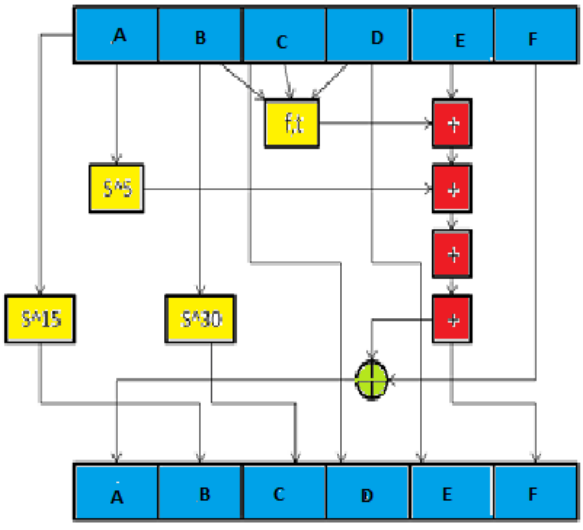


Figure 3: Operations of SHA 192

Here, A, B, C, D, E, F is the chaining variable. Each chaining variable holds 32 bits information. Initially all the chaining variable initialized with some value and during processing it changes its value and hold processing results and at last generates a result of 192 bits message digest.

## III. COMPARATIVE STUDY

In this section we compare the implemented results; we take output size, rounds and collision as a parameter of comparison in following table.

Table 1: Comparison Hash Algorithms

| Algorithm Name | Output Size | Rounds | Collision Found |
|---|---|---|---|
| MD-2 | 128 | 18 | YES |
| MD-4 | 128 | 48 | YES |
| MD-5 | 128 | 64 | YES |
| SHA-0 | 160 | 80 | YES |
| SHA-1 | 160 | 80 | YES |
| SHA-2 | 256/384/512 | 64/80 | No |
| SHA-192 | 192 | 64 | No |

Also some authors have developed their own SHA algorithms by doing some modification on compression function named SHA-192. To check the efficiency and strength of these algorithms, authors have developed these algorithms and compare it with SHA-1.

### *A. Time Analysis*

An experimental result of time taken by each algorithm for generating message digest is shown in Table 2.

Table 2: Timing Comparisons

| File Size | SHA-1 | SHA-192 |
|---|---|---|
| 1KB | 0.016 | 0.065 |
| 15KB | 0.10 | 0.53 |
| 20KB | 0.62 | 1.3 |

It is clearly seen that time taken by SHA-1 is very less compared to other algorithms hence it can be said that SHA 1 and SHA 2 are not time efficient algorithms.

## IV. CONCLUSION

This work concluded the overall view about the exiting hash function based algorithms. It is found that almost all the integrity algorithms have proven breakable except SHA-2 but it is not time efficient. SHA-1 hashing algorithm in terms of the number of brute force attacks needed to break it and moreover it is fast when compared to the other secure hash algorithms.

Many researchers have proposed their own algorithms but none of them are time efficient as SHA-1 and also there are chances of improving the internal strength of these algorithms. In near future we can develop a algorithm that is more secure, less time consuming, and better bit difference as compared to existing algorithms.

## REFERENCES

[1]. Garbita Gupta and Sanjay Sharma, *"Enhanced SHA-192 Algorithm with Larger Bit Difference"* IEEE International Conference on Communication Systems and Network Technologies, DOI 10.1109/CSNT.2013.42.

[2]. Richa Purohit, Upendra Mishra and Abhay Bansal, "A Survey on Recent Cryptographic hash Function Designs" International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), ISSN 2278-6856, Volume 2, Issue 1 January - February 2013.

[3]. Saurabh Jain and Saket Gupta "Automated Process of Server and Client Environment with attack alert based on DES",Fourth International IEEE Conference, on Communication Systems and Network Technologies (CSNT) DOI: 10.1109/ CSNT.2014.155 Page (s): 739 - 742 ,December 2014.

[4]. Rajeev Sobti, G.Geetha "Cryptographic Hash Functions: A Review" IJCSI International Journal of Computer Science Issues, ISSN: 1694-0814. Vol. 9, Issue 2, No 2, March 2012.

[5]. Harshvardhan Tiwari "A Secure Hash Function MD-192 with Modified Message Expansion" International Journal of Computer Science and Information Security. ISSN 1947-5500, Vol. 7 No. 2 February 2010.

[6]. L. Thulasimani and M. Madheswaran "Security and Robustness Enhancement of Existing Hash Algorithm" IEEE International Conference on Signal Processing Systems 2009.

[7]. Ricardo Chaves, Georgi Kuzmanov, Leonel Sousa and Stamatis Vassiliadis "Cost-Efficient SHA Hardware Accelerators" IEEE transactions on very large scale integration (VLSI)Systems, VOL. 16, NO. 8, AUGUST 2008.

[8]. Saurabh Jain, D S Tomar and Divya Rishi Sahu "Detection of JavaScript vulnerabilities at Client Agent" in International Journal of Scientific & Technology Research (IJSTR) ISSN: 2277-8616 Volume 1, Issue 7, and PP.36-41 in August 2012.

[9]. William Stallings, "Cryptography and Network Security: Principles and Practice", Third edition, Prentice Hall.2003.

[10]. Florent Chabaud and Antoine Joux, "Differential collisions in SHA-0", Advances in Cryptology-CRYPTO'98, LNCS 1462, Springer-Verlag, 1998.

 