International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 3 | Mar -2017 www.irjet.net p-ISSN: 2395-0072

A Novel Design For Generating Dynamic Length Message Digest To Ensure Integrity Using Block Cipher Approach

Umesh Gandhi[1], Dr.(Mrs.) Poonam Sinha[2], Ms. Rachna Kulhare[3]

[1](Department of I.I., UIT, Barkatullah University, Bhopal (M.P.), Inda)
[2](Professor, Department of E&C, UIT, Barkatullah University, Bhopal (M.P.), India)
[3](Asst. Professor, Department of I.T., UIT, Barkatullah University, Bhopal (M.P.), India)

Abstract- In today's life, security is one of the major concerns as our modern technologies changes. To ensure the integrity over the stored or transmitted data hash algorithms are designed, but as the time moves all the existing standard algorithms are either proven breakable or found not efficient to use. This paper has raised their voice on this direction and proposed their own new algorithm that can generates a hash with totally different way. It generates a dynamic size hash on the principal of cryptographic encryption/ decryption algorithm. An implementation result shows its strength and efficiency against existing hash algorithms.

Keywords: ***Computer Security, SHA, Hash, Message Digest***

1. INTRODUCTION

With the rapid change in the modern technologies there is always a requirement to develop or modernize the existing solutions that ensure the security. Integrity is one of the most important principal of security. It ensures the originality of received data packet.

Many hash algorithms have been designed to ensure the integrity over data packet. These algorithms generate a digest of fixed size i.e. finally transmitted with the data packet. At the other end, receiver receives the data packet along with its digest. The data packet again passed with the same hash algorithm to generate another digest. If the received digest is same as the generated digest it means the data packet is original otherwise it is not.

The digest generated by the hash algorithm must have the following four properties:

a. For a given message (M), it is easy to compute Message Digest MD (M).
b. For a given Message Digest MD (M), it should be practically impossible to find Message (M).
c. For a given Message (M), it should be practically impossible to fine Message (M') such that Message Digest of (M) is equal to Message Digest of (M').
d. For a given Message (M) and Message Digest (MD), little change in (M) should generate totally different Message Digest (MD).

Many algorithms have been designed up to now, to ensure the integrity, some standard algorithms are MD Family and SHA Family, but unfortunately today's date all the existing algorithms are either proven breakable or not in use because of its inefficiency in terms of execution timing.

Latest in 2009, due to security issues MD2 from MD Family get disappear in Open SSL and other network security services.

A collision attack that was published in year 2007, found a collision on MD4 from MD family in less than two hash operations.

An attack found in 2013; break the MD5 algorithm again form MD family in just 2^{18} operations.

An attack called boomerang attack make the complexity to find collision in SHA-0 is about $2^{33.6}$ operations.

Also, SHA-1 from SHA family also produces hash collision in between $2^{60.3}$ to $2^{65.3}$ operations.

After that many variants of have been designed by the researchers in order to generate a successful and secured digest but all of them get failed.

Latest, Meng-jiao Wang, Yong-zhen Li present their research with a name "Hash Function with Variable Output Length" in which they have designed an

algorithm that can generate variable size digest using MD5 algorithm. The key feature of this algorithm is variety in its digest. Before this all algorithms generate fixed sized digest but this algorithm moves one step forward and generate a variable size digest.

Although, after implementing the above solution it is found that the suggested solution is neither time efficient nor well designed compare to MD5 and SHA-1 algorithms but it opens a door in new direction of generating a variable size digest.

The aim of this paper is to design an algorithm that generates a variable size digest with efficiency and security.

2. METHODOLOGY OF PROPOSED WORK

Internal designing of the proposed algorithm is totally different from the existing SHA or MD algorithms. It is designed on a principal of cryptographic block cipher chaining mode. It has broken an earlier fashion of generating a fixed size digest; it generates a variable size digest. Also, proposed algorithm works on a pseudo – random number which gives the proposed algorithm strength and stand against various attacks.

Steps to design the hash using proposed algorithm is as follows:

Step.1 Inputs: First taking a hash length and a message as an input from the user.

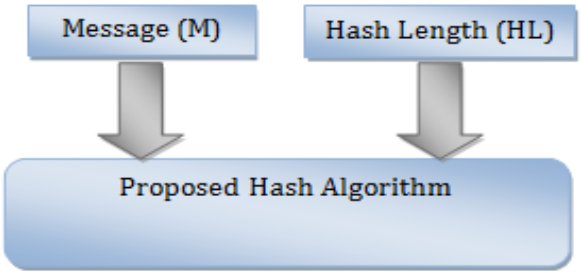


Figure 1 shows the input requirement of Proposed Hash Algorithm

Step.2 Pseudo Random Number RN1: Generating a first pseudo random number by adding ASCII value of all the characters in a message and then calculating the mod of this value by message length. Result is RN1.

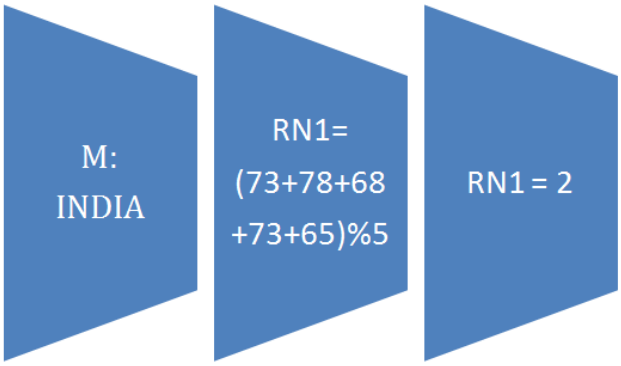


Figure 2 showing an example of generating a RN1 for a message "INDIA".

Step.3 Padding: Padding is to be performed on a message by '\0' to make a message multiple of HL. For example, if message is of 5 characters and hash length is of 8 characters.

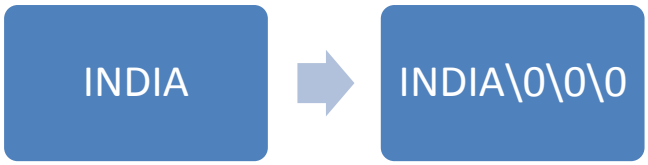


Figure 3 showing an example of padding.

Step.4 Initial Permutation: Perform the circular left rotation on M by RN1 value.

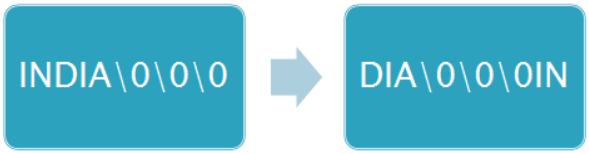


Figure 4 showing an example of left circular rotation on Message M.

Step.5 Creation of small Chunks: Divide the message M in to number of chunks equal to size HL

For example: If message (M) is "I love India" and Hash Length (HL) = 4

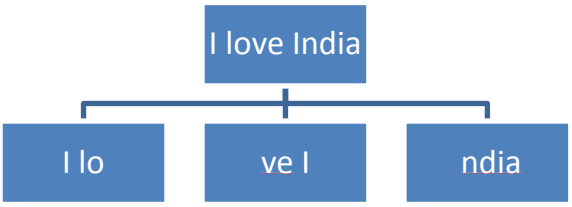


Figure 5 showing an example of chunks creation

Step.6 Repeat the following steps (a to h) for each chunks.

a. First, convert the chunk into binary format.
b. Next, calculating the pseudo random number RN2 by summing all the 1's in given chunk.

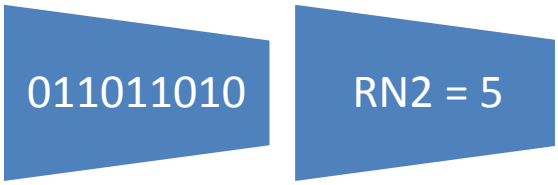


Figure 6 showing an example of generating a pseudo random number RN2.

c. Dividing the chunk again in four equal parts, C1, C2, C3 & C4.
C1 = Third Quarter of Chunk
C2 = First Quarter of Chunk
C3 = Fourth Quarter of Chunk
C4 = Second Quarter of Chunk

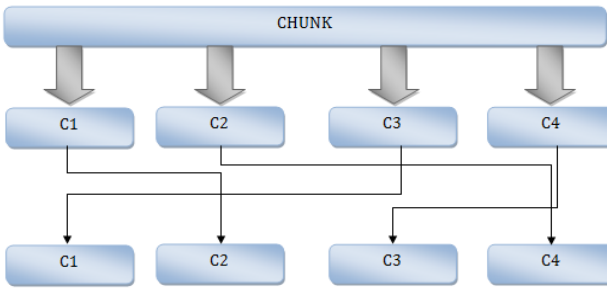


Figure 7 showing the further division of chunk in to four equal parts

d. Calculating XOR of C1 and C2 and store the result again in C2.
e. Next, perform circular left rotation on C1 by RN2 bits.
f. Calculating XOR of C3 and C4 and store the result again in C3.
g. Next, perform circular left rotation on C4 by RN2 bits.
h. Last, concatenate C1, C2, C3 & C4 and perform circular left rotation by RN1 on it & then XOR the result by next Chunk.

Step.7 **Repetition**: Result comes out from the last chunk again XOR with first chunk and repeats Step 6th once again and the final result comes out from Step 6 is the digest of a given message

Figure 8 (a) and Figure 8 (b) shows the block diagram of above discussed steps.

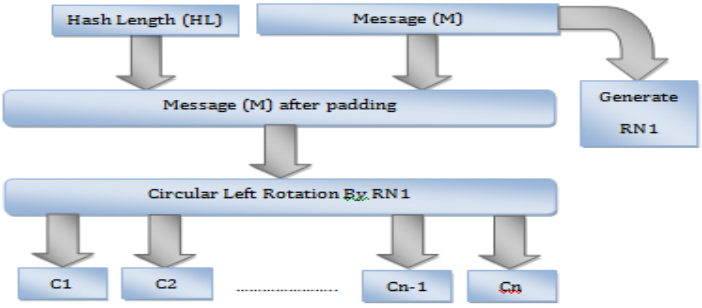


Figure 8(a) shows the block diagram of proposed algorithm

3. EXPERIMENTAL RESULTS

In this section authors have discussed the implementation results of proposed algorithms and compared it with standard SHA -1 algorithm, MD5 and modified MD-5[1] algorithm. Authors implement all four algorithms in DOT NET Framework. The testing was performed on a system having Intel Pentium Dual Core E2200 2.20 Ghz, 1 GB of RAM and Window XP Service pack 2 configurations.

Timing Analysis: It is an important for an algorithmic solution to be efficient in terms of execution time. As today modern technologies are changes very rapidly, hence demand of using Ad-Hoc network or wireless devices raises. Algorithms that are time efficient are suitable for such devices. Table 1 shows the implementation results and also compared it with the timing needed by proposed algorithm, SHA-1, MD5 and modified MD-5[1] algorithm for a same file Now, it is clearly seen from the above table

that proposed algorithm is far better solution than all the other existing solutions. Graph 1 also shows the above comparison graphically.

Strength Analysis: Comparison of only timing is not sufficient for the analysis of hash algorithm. It is required

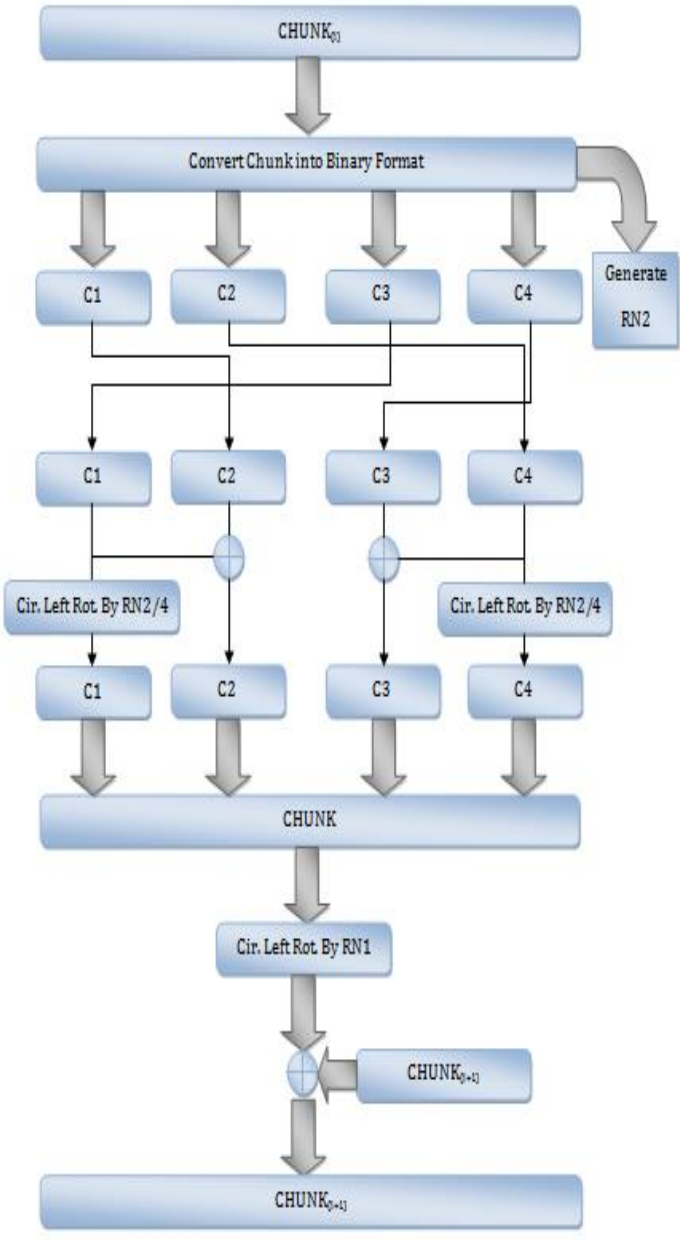
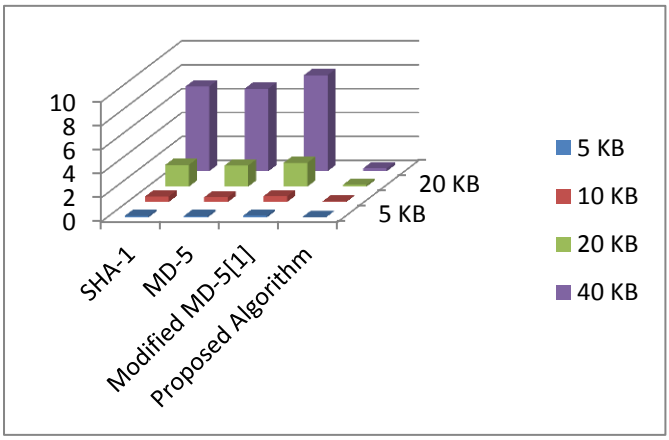


Figure 8(b) shows the block diagram of proposed algorithm

.TABLE 1 Timing Comparison between Proposed Algorithm, SHA-1, MD5 and modified MD-5[1] algorithm algorithms

File Size in KB	Algorithms (Time in Seconds)			
	SHA-1	MD-5	Modified MD-5[1]	Proposed Algorithm
5 KB	0.130	0.120	0.140	0.030
10 KB	0.460	0.400	0.480	0.060
20 KB	1.820	1.760	1.990	0.150
40 KB	7.080	6.900	8.020	0.240



Graph 1 Timing Comparison between Proposed Algorithm, SHA-1, MD5 and modified MD-5[1] algorithm

to do the analysis of internal structure of proposed algorithm to ensure that it is enough secure against any attack.

Avalanche effect is used to calculate the strength of internal structure of hash algorithm. According to the avalanche effect single bit change in message change the hash fifty percentages. It is an idle condition, an algorithm close to this condition is considered better designed compared to other which is far from this condition. Table 2 shows the comparison of avalanche effect of Proposed Algorithm, SHA-1, MD5 and modified MD-5[1] algorithm.

Now, again proposed algorithm proves its efficiency against avalanche effect. It is clearly seen from the above table that proposed algorithm is better designed and robust against various attack.

Graph 2 shows the same comparison graphically in better ways.

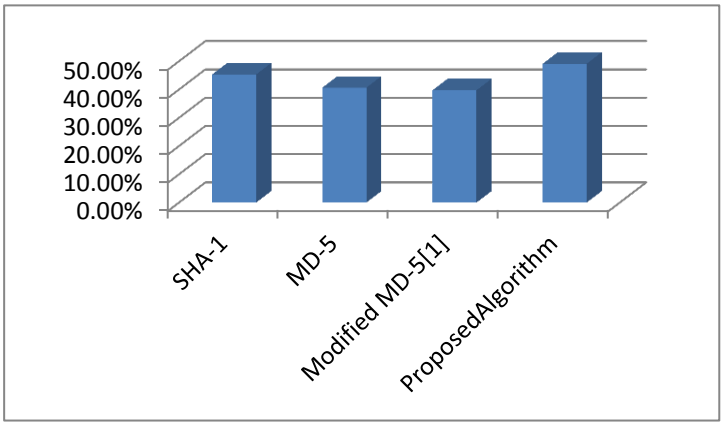
Table 2. Avalanche effect of Proposed Algorithm, SHA-1, MD5 and modified MD-5[1] algorithm

Algorithm	Avalanche Effect	
	Bit Difference (in Percentage)	Difference from Idle condition
SHA-1	45.31%	4.69 %
MD-5	40.63%	9.37%
Modified MD-5[1]	39.72%	10.28%
Proposed Algorithm	49.1%	0.9%

4. CONCLUSION

As rapid changes in modern technologies it is found that updating and modification in the existing algorithm is always required. This paper have done the same, after the detail study on various security algorithms, it is found that all the existing algorithms that are used to ensure integrity over stored or transmitted data are either proven breakable or not as efficient as it should be in order to match with the latest modern technologies and there requirements. All the existing algorithms like MD-4, MD-5, SHA-0 and SHA-1 are proven breakable recently. After this many researchers have tried and design their own algorithm to ensure integrity, so that they can fulfill the gap that was created after found collision on the mentioned standard algorithm but they were not succeed to fulfill that gap as in order to improve one or two parameters they have compromised the efficiency of other remaining parameter. This paper has taken the same goal and gets success to fulfill that gap completely. This dissertation has designed a novel algorithm which generates a variable size digest and also it works on pseudo- random number which makes it more robust against any attack. Its internal design is completely different from all existing algorithms.

To prove its efficiency and robustness against various attacks, this dissertation has implemented the proposed algorithm and presented their results and from the result it is very much clear that presented



Graph 2 Avalanche effect of Proposed Algorithm, SHA-1, MD5 and modified MD-5[1] algorithm

proposed algorithm is efficient as well as enough strong against any known attack. Also its efficiency makes it comfortable to use in Ad-Hoc network or real time communication or places where fast transmission required.

REFERENCES:

[1]. Meng-jiao WANG, Yong-zhen LI, "Hash Function with Variable Output Length", International Conference on Network and Information Systems for Computers, IEEE

[2]. Piyush Gupta, Sandeep Kumar, "A Comparative Analysis of SHA and MD5 Algorithm", International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 4492-4495

[3]. Kamlesh kumar Raghuvanshi, Purnima Khurana, Purnima Bindal, "Study and Comparative Analysis of Different Hash Algorithm", Journal of Engineering Computers & Applied Sciences(JECAS) ISSN No: 2319-5606 Volume 3, No.9, September 2014

[4]. Gupta G., Sharma, S. "Enhanced SHA-192 Algorithm with Larger Bit Difference" Published in IEEE International Conference on Communication Systems and Network Technologies (CSNT), 6-8 April 2013 Page(s):152 - 156 Print ISBN:978-1-4673-5603-9

[5]. Piyush Garg and Namita Tiwari, "Evolution of Sha-176 Algorithm", IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 2, Issue 2 (July-Aug. 2012), PP 18-22

[6]. Harshvardhan Tiwari, Dr. Krishna Asawa, "A Secure Hash Function MD-192 With Modified Message Expansion", *International Journal of Computer Science and Information Security, Vol. VII , No. II, FEB2010*

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

[8]. 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

[9]. Harshvardhan Tiwari and Dr. Krishna Asawa "A Secure Hash Function MD-192 with Modified Message Expansion" (IJCSIS) International Journal of Computer Science and Information Security, Vol. VII, No. II, FEB2010.

[10]. Debanjan Das, Megholova Mukherjee, Neha Choudhary, Asoke Nath, "An Integrated Symmetric key Cryptography Algorithm using Generalised modified Vernam Cipher method and DJSA method: DJMNA symmetric key algorithm" World Congress on Information and Communication Technologies-2011

[11]. Akhil Kaushik, AnantKumar and Manoj Bamela " Block Encryption Standard for Transfer of Data " IEEE International Conference on Networking and Information Technology 2010

[12]. Neeraj Khanna, Joyshree Nath, Joel James, Amlan Chakrabarti, Sayantan Chakraborty, Asoke Nat, "New Symmetric key Cryptographic algorithm using combined bit manipulation and MSA encryption algorithm: NJJSAA symmetric key algorithm", IEEE-2011

[13]. P.P Charles & P.L Shari, "Security in Computing: 4th edition", Prentice-Hall, lnc.,2008.

[14]. Cryptography and network Security Principles and Practices, Charles Fleeger.

[15]. SECURE HASH STANDARD Federal Information Processing Standards Publication180-2

[16]. Hash functions: Theory, attacks, and applications Ilya Mironov Microsoft Research, Silicon Valley Campus.

[17]. Cryptography And Network Security (William Stallings).

[18]. Computer Network (Andrew S. Tanenbaum).

[19]. John R. Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway,

[20]. UMAC—Message Authentication Webpage, www.cs.ucdavis.edu/~rogaway/umac/

[21]. John Black, Phillip Rogaway, and Tom Shrimpton, "Blackbox analysis of the block-cipher-based hash-functions constructions from PGV," Proc. of CRYPTO'02, Lecture Notes in Computer Science 2442, Springer, pp. 320–335, 2002. Available from www.cs.ucdavis.edu/~rogaway/papers/hash.htm