International Journal of Engineering Research & Management Technology
ISSN: 2348-4039
November-2018 Volume 5, Issue-6
Email: editor@ijermt.org www.ijermt.org

AN OVERVIEW-COMPARATIVE STUDY OF HASH FUNCTIONS

Vaishali Sharma, Nilufar Yasmin
Department of Electronics & Communication Engineering,
AKGEC, Ghaziabad
APJ Abdul Kalam Technical University, Lucknow (INDIA)

ABSTRACT
The tool adopted in the digital time sampling, integrity of messages and digital structure are Hash functions. For the authentication in cryptography with public key, message digest algorithm is adopted. Some hash function computation is made using Digest algorithm which are for 32-bit words message and digest values based on the operation of simple primitive set. To accomplish the number of security purposes Hash functions are used. In this paper, we carry out the significance of hash functions, its numerous structures, design practices, attacks and the progressive modern improvement in this field.

KEY WORDS: Computer Security, Hash, Message digest, Cryptography, Compression function.

INTRODUCTION
Cryptography is the skill to develop secret writing. The basic service delivered by cryptography is the facility to send information between applicants in such a way that stops others from understanding it. The four fundamental objectives of cryptography practice are Confidentiality, Data Integrity, Authentication, and Nonrepudiation. There are three variations of cryptographic functions: Hash functions, Public key functions and Secret key functions. Diffie and Hellman bring together the concept of public-key cryptography in order to solve the main management problem. In their perception, each person acquires a pair of keys, public key and private key. Each person's public key is in print out while the private key is preserved secret. Secret key cryptography comprises the practice of one key. All keys in a secret-key cryptosystem essential continue secret, secret-key cryptography frequently has trouble on condition that safe key management, particularly in open systems with a huge number of operators. Hash functions comprise the use of zero keys. Try to imagine what that could probably mean, and what use it could maybe have – an algorithm everybody identifies with no secret key, and yet it has practices in security. Today there exist many hash functions, but MD4 family of hash functions (MD4, MD5, SHA, RIPEMD, etc.) are commonly adapt because of the belief in their safety and the implementation speed they promise. A symmetric algorithm through a random session key is aimed to encrypt the message, and a public-key algorithm is recycled to encrypt the random session key. With a worthy cryptographic arrangement it is perfect OK to take everyone together with the bad dudes (and the cryptanalysts) identified the algorithm since information of the algorithm short of the key does not help unmingled the information.

In this paper, we emphasis on the cryptographic hash functions. We effort on the comparison study of hash functions. The widely adopted approaches of cryptography are:

1. Symmetric-Key Cryptosystem - In which the identical public key is practiced by both the sender to send and the receiver to recover the message, that is, the identical key is taken by both for encryption and decryption. A number of symmetric-key cryptosystems are: DES (Modes: ECB, CBC, CFB, OFB, CM), 3DES, AES, IDEA, Blowfish, RC4, RC5, CAST, SAFER, Two fish.

2. Asymmetric-Key Cryptosystem –In this, Keys are used for encryption and decryption is not identical. The public key has been used to encrypt the message by sender while receiver practices the private key to decrypt the message. The numerous asymmetric-key cryptosystems are: Diffie-Hellman, RSA, El Gamal, and Elliptic Curve Cryptography (ECC).

3. Hybrid Cryptosystem – It incorporate the features of both approaches (symmetric and asymmetric-key cryptosystems). Asymmetric allocates symmetric key, also recognized as a session key. Symmetric offers bulk encryption. The example of a hybrid cryptosystem is SSL.

The limitations of a symmetric-key algorithm:-Key-exchange becomes a problematic situation.
(a) Confidence problems between the proposed parties arise.
(b) More harm is produced, when someone involves on a symmetric key since they can decrypt the whole thing encrypted with that key.
(c) As the number of secret key participant increases, the danger of harm and the costs of this loss rises.

The weaknesses of an asymmetric-key algorithm:-
a) Asymmetric-keys is several times longer and additionally computationally expensive than the secret-key in symmetric-key algorithm.
b) They are vulnerable to attacks in less than brute-force time.
c) It is as well susceptible to man in the central attack.
d) To confirm the dependability of public keys, uses a third party in various public key systems
e) The message-digest or one-way hashing functions were then suggested as an substitute to achieve all the features of information security because of the following advantages:
   i) It is computationally convenient to analyze the hash of any known message.
   ii) Using the identical hash, it can`t have two messages associated with it.
   iii) Message cannot be altered without any changes in the hash value. It is illogical to produce the message with the assumed hash value.

2. HASH FUNCTION

A cryptographic hash function is used to confirm the integrity of the communicated data or stored data. Occasionally it is also termed as digest of a message. Hash function produces a message digest of a specified message for fixed size; this message digest is preserved as initials of that message. Hash function can be well-defined mathematically as MD = HF (M), where HF is a hash function having the following features-
(a) It is a one way function means it is convenient to estimate MD from M but vice-versa is not true at all.
(b) To find out two such messages M1 and M2 which produces same message digest i.e. HF (M1) ≠ HF (M2) is very tedious.

There are numerous algorithms considered to implement the hash function. MD-2, MD-4, MD-5, SHA-0, SHA-1 and SHA-2 are the greatest recognized algorithms for message digest.

There are mainly two types of cryptographic Hash function,
Keyed Hash functions and Un-keyed Hash Function. The only difference of these two techniques is that first one uses a secret key and second one does not. The keyed Hash functions are known as Message Authentication code. Generally, the term hash functions known to be unkeyed hash functions. In this paper, we will pay attention on Un-keyed Hash functions only. Depending upon the additional properties it follows, unkeyed or simply Hash functions (sometime known as MDC – Manipulation Detection Code) can further divided into OWHF (One Way Hash Functions), CRHF (Collision Resistant Hash Functions) and UOWHF (Universal One way Hash Functions).
One Way Hash Functions (OWHF) - OWHF is defined as hash function H by Merkle which fulfill the following conditions:
(a) H can be utilized to block data of any length. (In practice, 'any length' may be actually be bounded by some huge constant, larger than any message we ever would want to hash.)
(b) Output produced by H has a fixed-length.
(c) It is easier to compute message digest H(x) for a given input H and x.
(d) For given H and H(x), it is inappropriate computationally to know x. Given H and H(x), it is improbable to find x and x' such that H(x) = H (x') computationally.

For practical applications to message authentication and digital signatures of a hash function, first three conditions are must. Pre-image resistance or one way property, the fourth requirement states that it is easy to generate a message code given a message but hard (virtually impossible) to generate a message given a code. The fifth requirement, the second pre-image resistance condition ensures that the same code for the given code can`t be found by using alternative message hashing.

Collision Resistant Hash Functions (CRHF) - Merkle given the early definitions of Collision Resistant Hash functions. CRHF can be defined as a Hash function H which fulfills the requirement of OWHF and in addition satisfy the following collision resistance property:

Given H, it is computationally infeasible to find a pair (x, y) such that H(x) = H(y).

Universal One Way Hash Functions (UOWHF) –
It presented the idea of Universal One Way Hash functions imparted a digital signature scheme which is not on trapdoor functions based. To construct UOWHF on the basis of one way function which leads to implement Digital Signature scheme? The property of UOHWF Security is analyzed as follows:

Let U has a hash functions of finite number and each of them have the same probability of being used. Let a probabilistic polynomial time algorithm A (A is collision adversary) works in two modes. Initially, a hash function H is selected from the family U for receive input k and outputs a value x called as initial value. A then receives H in such a fashion that output y must be H(x) = H(y). In other words, once we have a hash function it tries to find a collision with the initial value. Now U will be labelled as a family of Universal One Way Hash Functions if for all polynomial-time A the probability that A succeeds is negligible.

3. SECURITY SERVICES OF CRYPTOGRAPHIC HASH FUNCTIONS

3.1 Achieving Integrity & Authentication -The prime essentials in computer systems and networks is to ensure the integrity and authenticity of information. In particular, two users communicating over an insecure and unsafe channel require a way by which information delivered by one user can be validated as authentic (or unmodified) by the other. Message Authentication and integrity of message may be achieved in different ways. Mechanisms based on schematic Encryption may be used but they have their own disadvantages. Speed, optimization for data sizes, cost factor etc. are the drawbacks which is pointed by Tsudik Confidentiality and Authentication functions are combined by such methods. Although, encrypting full message (confidentiality) is not required in some scheme. For such applications it is important concern to keep authenticity rather than message secret. For example, in SNMP (Simple Network Management Protocol), hiding the SNMP traffic is not required but for managed system to authenticate incoming SNMP commands (like changing the parameters at the managed system) is important. The alternative techniques MAC or hash functions are required to implement message authentication and integrity.

3.2 Implementing Efficient Digital Signatures- The security goal of a cryptosystem is digital signature which leads to achieve security service, authenticity and or property of non-repudiation. The security goal of Digital Signatures can`t be carried out by using MAC and Hash Functions only. Many algorithms have been used to show the digital signature. To optimize the digital signature schemes, Hash functions are used. The size of the signature in the message will remain the same without the use of Hash. The sender only signs the digest of the message adopting a signature generation algorithm rather than applying fundamental concept of generating the signature for the whole message which need to be authenticated. The sender then transmits the message and the signature to the contracted receiver. The receiver certify the signature of the sender by examining the digest of the message adopting the same hash function as the sender and correlating it with the output of the signature verification algorithm. It is noticeable that in the absence of hash functions, this method saves a lot of calculation associating in signing and verifying the messages.

3.3 Authenticate Users of Computer Systems - For user authentication at the time of login, Hash functions may be accounted. To avoiding the access of the same even to Database Administrators (because of Pre-Image resistance of Hash digest) passwords are stored in the form of message digest. The message digest of the entered

password is compared and computed with the digest saved in the database, when user tries to login and enter the password. If it matches, then login is successful, otherwise user is not authenticated.

3.4 Digital Time Stamping – For changing digital documents, number of simple tool and techniques are available to change text, audio and video documents in digital format. Some approach is required to confirm when document was designed or last modified. To solve this purpose, digital timestamp contribute to temporal authentication. There are several techniques such as simple scheme based on trusted third party, scheme based on timestamps into temporal chain and planted of Merkle Tree. Digital time stamp adds in assuring intellectual property rights, providing firm auditing procedures and achieving true non-repudiation services. The users can find out the accurate knowledge how to use digital signatures, one way hash functions for carrying out the digital time stamping.

3.5 Hash functions as PRNG - Hash functions as one way functions can be adopted to achieve PRNG (Pseudo random number generator). An uncomplicated way to start from an initial value(s) commonly known as seed and computer H(s) and then H(s+1), H(s+2) and so on. Some other methods of designing Pseudo random strings from Hash functions is given in it.

3.6 Session Key Derivations - For constructing sequence of session keys used for the protection of successive communication sessions can be achieved by implementing Hash functions as one way functions. Beginning with a master key K0, K1 = H (K0) and K2 = H (K1) can be first and second session key respectively and so on. Based on control vectors, key management scheme is being described which accounts the benefit of hash functions and Encryption functions for session keys generation.

3.7 Constructions of Block Ciphers - For developing a cryptographic hash function, Block ciphers can be used and vice-versa. The compression function of cryptographic hash function SHA-1 is expected to use by user in encryption mode. The SHACAL was the name of cipher. SHACAL-1 (originally named SHACAL) and SHACAL-2 based on SHA-1 and SHA-256 respectively are block ciphers. SHACAL-1 (originally named SHACAL), SHACAL-2 is 160-bit clock cipher and 256 bit block cipher respectively. Both were preferred for the second phase of NESSIE project. Due to concerns about its key schedule, SHACAL-1 was not selected for NESSIE portfolio, while SHACAL-2 was selected as one of the 17 NESSIE finalists finally. Adopting the state input as the data block and availing the data input as the key input, the compression function of SHA-1 was adopted by SHACAL-1 and turned it into a block cipher. In other words, SHACAL-1 considered the SHA-1 compression function as an 80-round, 160-bit block cipher with a 512-bit key. Keys which are shorter than 512 bits are promoted by padding them with zero up to 512. Keys shorter than 128-bit were not preferred to use SHACAL-1.

3.8 Other Applications -
We can also use Hash Functions to index data in hash tables, for fingerprinting, to identify duplicate data or uniquely identify files, random numbers generation and as checksums to detect accidental data corruption.
For wide range of applications, it does not seem to be quiet imperative that Hash Functions is the part of one of the particular cryptographic sub branch. These cryptographic tools justify a isolated status for them. Almost in all places, these can be used in cryptology where we need the efficient processing of information.

One of the unsolved key challenges is to develop UOWHF of higher orders efficiently in cryptography.

4. METHODS OF ATTACK ON HASH FUNCTIONS

Breaking one of the security properties like basic, extended or certification property of hash functions known as attacking a hash function means. For example, breaking pre-image resistance simply means adversary is capable of cracking the pre-image property means construct a message that hashes to a specific hash. We will analyze different types of attacks on hash functions. Attacks on Hash functions can be divided into - Brute Force Attacks and Crypt analytical Attacks.

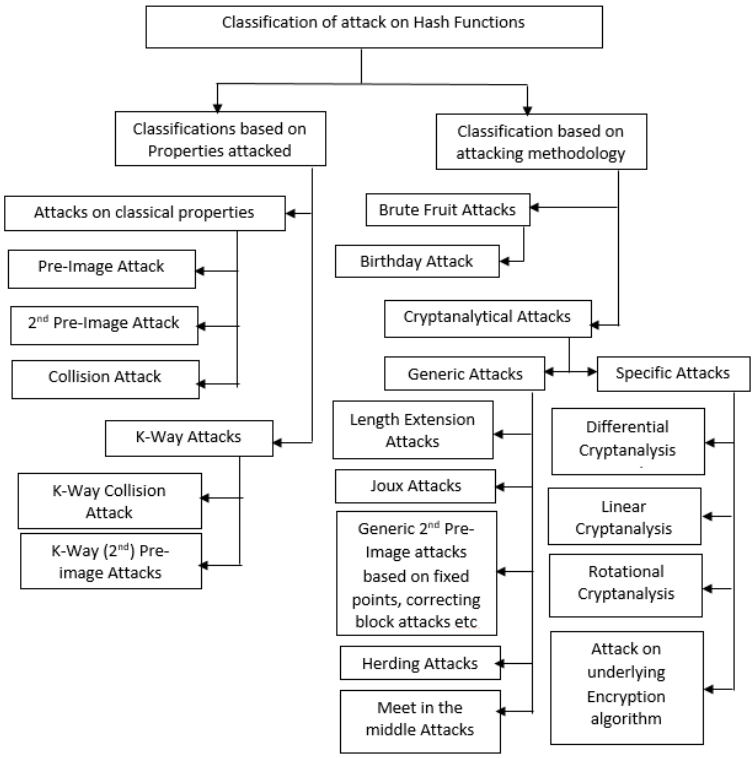

Fig. 1 Classification of attacks on Hash Functions

5. HASHING ALGORITHMS

In this section authors have provided brief overview about all standing hash Algorithms-

A. Message Digest 2 (MD2) – In 1982 MD2, a cryptographic hashing algorithm came in existence to generate message by consuming a compression function of 18 rounds for digests of 128 bits. Post 2004, MD2 is recognized to be subject to pre image attacks of time complexity equal to 2104 applications of the compression function (Muller, 2004). Hence, in sense of the MD2 author, "MD 2 can no longer be assumed a secure one-way hash function". Post 2008, MD2 was shown to be more in danger and utilizable than it was originally believed to be effective pre image attacks being achieved with a time complexity of 273 compression function estimates making it more reasonable to exploit and more risky to practice. MD2 is also evidenced to be vulnerable to collision attacks, time complexity of 263.3 compression function evaluations in 2009.

B. Message Digest 4 (MD4) - Another cryptographic hashing algorithm published in 1990 is MD4 adopted to produce message digests of 128 bits and a word size of 32 bits by means of a compression function of 48 rounds. It succeeds the small endian notation as other hashing algorithms carried. The length of message in bits using MD4 extended up to the total length is corresponding to 448 mod 512. MD4 algorithm adds a '1' bit for padding at the end of the message and ads '0' bits until the padding settings are satisfied. The length before padding (64-bits) is attached in the end. In 2007, MD-4 was verified to be very unproductive hashing algorithm when it was known that hash collisions can be construct in fewer than 2 hash procedures in a collision attack that was afterward issued in the same year. It has also been demonstrated that it is feeble and inefficient to pre-image attacks.

C. Message Digest 5 (MD5) - Ronald Rivest in 1991 developed a MD-5, a cryptographic hashing algorithm to produces hash value for a 128 bits fixed length. MD5 is the successor of the faulty MD4 and was considered about the structures and performance of 32-bit processors in mind but is in fact sluggish than MD4 but accepts weighty similarity to MD4. An importance on 32-bit processors is completed because, the four word buffers (A, B, C, D) that are adopt to work out the message digest are having a 32-bit register each. The central process of MD5 is very

similar to MD4 and holds the same phases of attaching padding bits monitored but joining the length of the original message monitored the initialization of the MD buffers tracked by processing the message in 16-word blocks which contains of 64 operations, assembled in four sequences of 16 operations surveyed by the concluding output. In 2005 Xiaoyun Wang and Hongbo Yu confirmed that MD5 was promising to achieve a modular differential attack and break the collision resistance.

D. Secure Hash Algorithm (SHA-1)-SHA-1, a cryptographic hashing algorithm was established by the NIST [4] in 1993 to for 160-bit message digest to produce. SHA-1 stands a outstanding resemblance to the MD5 cryptographic hashing algorithm. At one point of time, it was the best favored hashing algorithms for integrity examination due to its time effectiveness and flexibility.

E. Secure Hash Algorithm (SHA-2) - SHA-2 settled by NSA is a cryptographic hashing algorithm. SHA-256 and SHA-512 are two namely variations of it [13]. The main variations between these two modifications depends upon the size of words used. SHA-256 and SHA512 uses 32-bit words and 64-bit words respectively. While, neither SHA-256 nor SHA-512 have been evidenced to be inconsistent, still they are not favored for integrity authentication as they are not as effective and well organized as SHA-1 in relations of time complexities. Also, as SHA-2 is resultant from SHA-1 which in turn is built on the Merkle Damgård structure exploited to break the SHA-1 cryptographic hashing algorithm. Thus, in theory SHA-2 can also be destroyed.

F. Secure Hash Algorithm 3 (SHA-3) - In 2012, SHA-3, a cryptographic hashing algorithm selected by the NSA after a unrestricted competition among non-NSA originators. The former name of the SHA-3 hashing algorithm previous to the consequences of the struggle was keccak. When keccak developed as the winner of the SHA-3 competition, it was give name again to SHA-3. While SHA-3 cares the same hash lengths as SHA-2, the inside construction is very different and is untouchable to attacks like length extension which both the MD5 and SHA-1 were verified to be vulnerable. The key purpose for the formation of the SHA-3 algorithm is due to the hypothetical attacks that are likely contrary to SHA-2. While there no concrete evidence has been give in to revealing the defects of SHA-2, one cannot reject that it is really possible.

G. BLAKE2 - A cryptographic hashing algorithm named BLAKE, established by Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and Raphael C.-W. Phan was one of the contributors to the SHA-3 competition. Some developments of BLAKE2 over the original BLAKE take account of higher performance due to features similar to shortening the number of rounds of compression from 16 to 12 for BLAKE2b and 14 to 10 for BLAKE2s and falling the number of initialization words from 24 to 8. Due to the lesser number of rounds, the necessity of random-access memory of the BLAKE2 algorithm is meaningfully lower than the original BLAKE by 33% approximately. BLAKE2 equipment with tree hashing for additional update or confirmation of large files. BLAKE2 apparatuses nominal padding for messages and is overall, computationally quicker and humbler than BLAKE to implement.

H. SHA-256(Secure Hash Algorithm) - There are number of restrictions and security concerns of SHA-1. The limitations of SHA-1 are eliminated by SHA-2 algorithm. SHA-2 has number of hashing algorithms like SHA-224, SHA256, SHA-384 and SHA-512. The difference between these algorithms is based on only they have different message digest. On SHA-1 the collision attacks are available. But on SHA-2 there is no any collision attack yet been formed. The SHA-256 has message digest length of 256. The SHA256 is more protected and faster than SHA-1 Algorithm. It takes less time to produce hash value as compared to SHA-1. The SHA-256 has 64 number of rounds. On SHA-2 no attack yet has been produced. The algorithm of SHA-2 is similar as SHA-1 algorithm.

I. SHA-512(Secure Hash Algorithm) - In SHA-2 the SHA-512 is strongest among other SHA-2 algorithms. The SHA-512 produces message digest three times greater than SHA-1. That's why SHA-512 is more protected. The algorithm of SHA-512 adopts more complex operations to SHA-1, making the algorithm by itself robust and durable. In SHA-2 the SHA-512 is more protected and faster than other SHA-2 algorithms. The SHA-2 has 80

number of rounds. The SHA-512 has block size of 1024 bits. On SHA-512 no attack yet has been suggested. It is safe because length of message digest.

J. SHA-160 Hash Function: The SHA-1 produces a single output of 160-bit message digest (the output hash value) from an input message. The input message is confined of multiple blocks. The input block, of 512 bits, is divided into 80 of 32-bit words, denoted as, one 32-bit word for each computational round of the existing SHA-1 algorithm. All round includes several operation like additions and logical operations, and bitwise logical operations and bitwise rotations to the left. Total calculation of the algorithm subject to on the round being performed, as well as the value of the constant. Four groups of 20 iteration of each splits from SHA-1 80 iteration for different values and the applied logical functions.

K. SHA-192 Hash Function: In SHA-192 algorithm chaining variable is increased by one more variable is the extension of the SHA-160 algorithm. Due to this modification message digest produced is of 192 bits. The extended sixteen 32 bit into eighty 32 bit words are given as input to the round function and some variations has been made in shifting of bits in chaining variables, computation configuration of this algorithm. SHA-192 Hash Function: SHA-160 is the extension of the SHA-192 algorithm. This algorithm consists of chaining variable by increasing one more variable in the existing algorithm.

MD5 vs MD4

A fourth round has been added. Each step has a unique and add constant. The function g in round 2 is changed from (XY v XZ v YZ) to (XZ v Y not (Z)). Each step adds in the result of the previous step. The order in which input text words are fetched in rounds 2 and 3 are altered. The shift amounts in each round have been optimized. The shifts in different rounds are dissimilar.

SHA vs MD5

In one platform, SHA1 and MD5 appear to be very same. Their diagrams contain bundles of bits, bit rotation, xor and extraordinary functional operation. Generally, their carrying out are of the same length, but many of identifies widely known that MD5 is fall down, but currently SHA1 is functioning. Some of key designable changes like - SHA-1 have a huge state: 160 bits message vs 128 bits message. SHA-1 has more step rounds: 80 vs 64. SHA-1 rounds have an extra bit rotation and the clubbing of state words is very less poles apart. Bitwise clubbing functions and round constants are not same. Bit rotation counts in SHA-1 are the similar for all rounds, while in MD5 each round has its own rotation count. The message bit words are pre-scheduled in SHA-0 and SHA-1, In MD5 each round uses one of the 16 message words as it is.

6. COMPARITIVE ANALYSIS

The comparison between the different hashing algorithms has been offered in this section. With some analysis, some of the algorithms are proved to be weak and breakable. To ensure higher security, succeeding advancement have been completed to the newer ones. Most of the collective features of a hashing algorithm are positioned into emphasis to help and understand the benefits and shortcomings of a hashing algorithm in comparison to another hashing algorithm

Hashing Algorithms	Properties of Algorithm		
	Digest Length (in bits)	Number of Rounds	Collision Status
MD2	128	18	YES
MD4	128	3	YES
MD5	128	60	YES
MD6	<=512	Max(80,40+[d/4])	NO
SHA-1	160	80	YES

SHA-2	256/512	60/80	THEORI-TICAL
SHA-3	256/512	24	NO
BLAKE-2	256/512	10/12	NO
SHA-256/224	256/224	64	YET
SHA-512/384	512/384	80	NONE/ YET
SHA-160	160	80	YES
SHA-192	192	80	NO

Table 1: Comparison of Multiple Hashing Algorithms

From Table 1, we understand that both SHA-3 and BLAKE 2 have not been evidenced to be liable to hash collision and are thus contenders for ideal hashing algorithm for message singing. Further, there are no identified security issues for SHA-3 and BLAKE2. A detailed comparison between the two most popular hashing algorithms in active use (MD-5 and SHA-1) and a strong contender for our hashing purposes. As shown in Table 2, BLAKE2 support to further build an understanding to these hashing algorithms and their features.

FEATURES	Hashing Algorithms		
	MD-5	SHA-1	BLAKE-2
Security	Less secure than SHA-1	More secure	Secure as SHA-3
Length of message digest	128 bits	160 bits	256/512 bits
No. of attacks needed to find original message	2^123.4 bit operations required	2^151.1 bit operations required	2^256 or 2^512 (Exhaustive search)
Attacks to try and find two message producing the same MD	2^49.8 bit operations required	Between 2^60.3 and 2^65.3 bit operations	2^256 or 2^256 (Exhaustive search)
Speed	60 iterations, Faster	80 iterations, Slower	Faster than SHA and MD
Successful attacks reported	YES	YES	NO

Table 2: Feature Comparison of Hashing Algorithms

Although all cryptographic hashing algorithms strive to obtain and prefect the core principles of cryptographic the process they follow are very different from each other. Thus, while all cryptographic hashing algorithms are using to generate a hash, each algorithm generates a different hash, compared to each other, for the same input.

TIMING ANALYSIS - Timing is one of the important factors in evaluation of performance of any algorithm. An algorithm that take more time to generate the message digest will considered less preferable than other which generate fast message digest. Authors have implemented both the algorithms and evaluated the time taken by these algorithms to generate the message digest and after testing on more than 50 files of each size the average time of the experimental results is made known in Table 3.

File Size in KB	Algorithms (Time in Seconds)		
	SHA-1	**MD-5**	**Modified MD-5**
5 KB	0.174	0.128	0.140
10 KB	0.525	0.423	0.492
15 KB	1.156	1.054	1.121
20 KB	1.982	1.921	1.935

Table 3: Comparison of Timing between SHA-1, MD-5 and modified MD-5 algorithm

SECURITY ANALYSIS - Another important factor in designing an algorithm is security. Whether the algorithm is secure or not is always a question. It is always a point of discussion that how to measure a security of any algorithm. As such no cryptanalysis attack has been found on Modified MD-5 [1], but because of only this reason nobody can say that Modified MD-5 is secure. So to check the security Modified MD-5 algorithm avalanche effect of all three algorithms is calculated. Avalanche effect is one parameter which can be used to check the internal strength of any cryptographic algorithm. According to avalanche effect, change in a single bit closer to avalanche value is considered more preferable. After testing on more than 50 different files authors have concluded the result shown in Table 4.

Algorithm	Avalanche Effect	
	Bits Changed	Percentage
MD-5	58/128	45.31%
Modified MD-5	52/128	40.63%
SHA-1	73/160	45.63%

SPACE ANALYSIS - Another constraint to assess the performance of all the algorithms is space. As discussed SHA -1 uses five chaining variable of 32 bit which actually store the hash value, but on the other end MD-5 uses four chaining variable of 32 bit while modified MD-5 of n bit digest uses n/32 chaining variable. If the value of n is more chaining variable is more. Therefore, SHA needs more space than MD-5 whereas Modified MD-5generates variable size digests hence if value of n is more than it required more space and if value of n is low than it required low space. Further we evaluated the all space analysis according to the algorithms.

ANALYSIS OF HASH CODE - If we practice of n bits to characterize the hash code, there are only 2n distinct hash code values. If there place no limitations whatsoever on the messages and if there can be a random number of different possible messages, then definitely there will exist several messages giving rise to the same hash code. But then allowing for messages with no constraints whatsoever does not characterize reality because messages are not noise they must retain considerable arrangement in order to be understandable to humans. Collision resistance refers to the probability that two different messages holding certain simple configuration so as to be meaningful will result in the same hash code. Given a block of k messages, the question "What is the chance that there exists at least one message in the pool whose hash code is the same to a specific value?" the answer is that select a block of k messages, each of which has a hash code value from N possible such values, the probability that the block will comprise at least one pair of messages by means of the same hash code is given by

$$1 - N! / (N - k)! N^k.$$

So if there use an n-bit hash code, then it has N = 2n. In this case, a message pool of 2n/2 arbitrarily produced messages will hold at least one with a quantified value for the hash code with a probability of 0.5.

ACKNOWLEDGMENT

Ajay Kumar Garg Engineering College (AKGEC), Ghaziabad is affiliated to Dr. A.P.J. Abdul Kalam Technical University and is approved by the All India Council for Technical Education. The college is accredited by NAAC. The college was established in 1998.

The authors would like to thank the Ajay Kumar Garg Engineering College, Ghaziabad (UP), India, for his guidance and support throughout the completion of this work & help me to overcome the hurdles.

CONCLUSION AND FUTURE SCOPE

This paper contributes the summary about completely integrity algorithms. All efforts have been made to provide a whole representation of cryptographic hashes, its arrangement procedures and weaknesses. It is found that almost all the integrity algorithms have demonstrated fragile but SHA but it is not time efficient. Many researchers have proposed their own algorithms but none of them are time effective as SHA and also there are risks of enlightening the internal strong point of these algorithms. Upcoming effort can be prepared on this to decrease the time delay and also certain labor can be done to advance the inner strength of this algorithm**.** We can also work more on these algorithms so as to improve the space and time complexity further. We can design tools like the modified version of present day salts to improve these.

REFERENCES

1. Text book William Stallings, Data and Compute Communications, 6eWilliam 6e 2005.
2. FIPS 180, Secure Hash Standard (SHS), National Institute of Standards and Technology, US Department of Commerce, Washington D. C., 1993.
3. P. Rogaway, and T. Shrimpton, “Cryptographic Hash Function Basics: Definitions, implications and separations for preimage resistance, second preimage resistance, and collision resistance”, inFSE, 2004, pp.371-388.
4. P. Gauravram, “Cryptographic Hash Functions: Cryptanalysis, design and Applications”, Ph.D. thesis, Faculty of Information Technology, Queensland University of Technology, Brisbane, Australia, 2003
5. Garbita Gupta and Sanjay Sharma, “Enhanced SHA192 Algorithm with Larger Bit Difference” IEEE International Conference on Communication Systems and Network Technologies, 2013.
6. W. Diffie and M. E. Hellman, (1976) “New Directions in Cryptography”, *IEEE Transactions on Information Theory*, Vol. 22, No. 6.
7. Harshvardhan Tiwari. A Secure Hash Function MD192 with Modified Message Expansion” Vol. 7 No. 2 February 2010 International Journal of Computer Science and Information Security.
8. X. Wang, H. Yu,(2005),How to Break MD5 and Other Hash Functions, Advances in Cryptology, proceedings of EUROCRYPT 2005, Lecture Notes in Computer Science 3494, pp. 19–35.
9. L.Thulasimani and M.Madheswaran “Security and Robustness Enhancement of Existing Hash Algorithm” IEEE International Conference on Signal Processing Systems 2009.
10. Florent Chabaud, Antoine Joux, “Differential collisions in SHA-0,” Advances in CryptologyCRYPTO’98, LNCS 1462, Springer-Verlag, 1998.

AUTHOR BIOGRAPHY

Vaishali Sharma pursuing M.TECH (Electronics and Communication Dept.) from Ajay Kumar Garg Engineering College, Ghaziabad. She had completed her graduation from Lord Krishna College of Engineering in 2016. Her Working field is Cryptography and is currently working on message digest applications