DOI: http://dx.doi.org/10.26483/ijarcs.v11i2.6508

Volume 11, No. 2, March-April 2020

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

Cryptographic Hash Functions and Attacks – A Detailed Study

Rituparna Kundu
Department of Computer Science,
T H K Jain College
Kolkata, India

Ambar Dutta
Amity Institute of Information Technology,
Amity University
Kolkata, India

Abstract: The term hash function has been used in computer science from quite some time and it refers to a function that compresses a string of arbitrary input to a string of fixed length. Cryptographic hash functions are one of the most important tools in the field of cryptography and are used to achieve a number of security goals like authenticity, digital signatures, pseudo number generation, digital steganography, digital time stamping etc. For the past few decades cryptographic hash function become the centre of attention in the cryptographic community. The security of hash function became an important topic as almost every day the world of hash function is facing a new attack. The present paper provides an extensive study on cryptographic hash functions with their applications, properties and detailed classification and also presents a detailed description of cryptographic hash algorithms. It also discusses a general classification of all kinds of possible attacks on hash function analyses some attacks on specific hash functions.

Keywords: Hash function; Classification, Hash Algorithms, Compression function; Comparison, Attacks

I. INTRODUCTION

The term cryptology consists of two concepts – one is cryptography that is the technique of information security and other is cryptanalysis that is the technique of information disclosure. Though cryptology mainly concerns with protecting confidentiality of information but by protecting the privacy of the information, other security parameters such as authenticity of the information could be achieved automatically. The concept of converting intelligible data into unintelligible format before transmitting is popular from ancient ages. This technique of hiding information is called cryptography in other words cryptography is the technique to create secure communication protocol and this is done by Cryptosystems. Mainly cryptography can be classified in three categories: (i) Symmetric key cryptography, where the same key is used in both encryption and decryption process (Forouzan, B, A. and Mukhopadhyay, D. 2010). The key must be secret, as the same key is used. (ii) Asymmetric key cryptography, where instead of one a pair of key is used, one public key (K_p) and one secret key (k_s) (Forouzan, B, A. and Mukhopadhyay, D. 2010). The public key is known to all and the secret key is known to authorized user or users. (iii) Hashing, where we get a fixed-length message digest out of a variable-length message. Compared to the message the digest is normally much smaller. The main purpose of hashing is related with message security like protecting message integrity, authenticity etc. That means hashing can be referred as a technique to achieve the purpose of cryptography or considered as new category of cryptography.

II. CRYPTOGRAPHIC HASH FUNCTION

A hash function can represent a much longer message with a small unique message i.e. maps a variable length message into a fixed length output called a hash value or message digest. There are different uses of hash function in computer science. Such as uniform distribution of storage, sorting, searching (as hash table in data structure), in checksum algorithm for error detection and in cryptology. As hash function used in security related application sometimes they specifically called Cryptographic hash function. The main purpose of cryptographic hash function related to computer security. The most important role it plays in checking whether the receiver receives the converted message or not. So basically we can achieve the purpose of cryptography through hash functions.

There are various applications of cryptographic hash functions, some of which are mentioned below:

Protecting Authenticity – Hash function takes the message to be authenticated and the secret key as input and gives a message authentication code. The MAC can be recomputed at the user end where any changes to the message can be detected. This provides both authentication and integrity.

Digital Signature – Instead of applying digital signature generation on the entire message it can be used with the hash value of the message which is more convenient.

Password Protection – If we can protect the password from unauthorized user, the hash value of the message can be stored. At the time of user authentication the hash value of the password presented by the user is compared with the stored hash value.

Basically there are three fundamental properties of the hash function but there are other derived properties also.

Pre-image resistance – A hash function "g" is said to be pre-image resistance if from a known output (h) of the function if is quite impossible to find the input (x) i.e. g(x)=h. This property is also known as one-wayness [5].

2nd pre-image resistance – A hash function g is said to be second pre-image resistance if for any known input (x), it is hard to find another input (y) such that both gives the same output. i.e. g(x) = g(y).

Collision resistance – A hash function g is collision resistance if it is hard to find any two different input such that for both the input the function generates same output.

The following figure (Figure 1) is the general classification of hash functions based on three criteria.

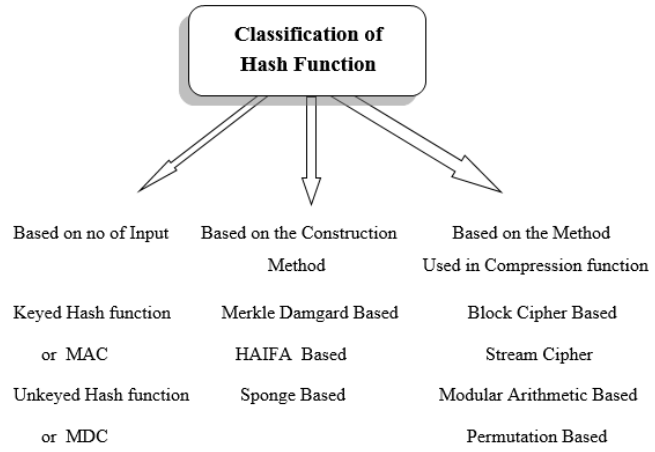

Figure 1. Classification of Hash function.

A. Classification based on number of input

A hash function can be classified depending on the number of input the hash function is taking - (i) unkeyed hash function and (ii) keyed hash function. Hash function that takes only one input (the variable length message) is called Unkeyed hash function and sometimes referred as Message detection code (MDC). On the other hand, keyed hash function takes a pair of input (the variable length message and the fixed length secret key). Usually it known as Message Authentication Code. The MDC can be further classified as (a) One Way Hash function, and (b) Collision Resistant Hash function. A hash function is called one-way hash function (OWHF) when it satisfies pre-image and second pre-image properties. And, a hash function that satisfies all three properties is called collision-resistance hash function (CRHF).

B. Classification based on Hash function construction

Though iterated hash functions are the most successful method but there are many approaches to do the iteration and based on that different hash functions are there.

1) Merkle-Damgard Construction:

Among the methods used for hash function construction Merkle-Damgard is the most popular one. It was designed in 1989 by R. Merkle (Merkle, R, C.1989) and I. Damgard (Damgård, I. August 1989). It can be described in three steps.

First, the message is padded to make it multiple of message block length. Hash functions based on this method can process messages of maximum 2^{64} -1 length.

In the **Second** step the padded message is divided into m bit blocks $(m_0, m_1, \ldots, m_t)$

Third, a fixed known initialization value (IV) is applied to a chaining process, and each time the output is treated as the next chaining value along with a message block.

The hash function can be described as –

$$h_0 = IV$$
$$h_i = f(x_i, h_i) \qquad i = 1,2, \ldots \ldots, t$$
$$h(x) = g(h_t)$$

where f is the compression function of the algorithm and sometimes an optional transformation g(x) applied to the h_t (described in Figure 2).

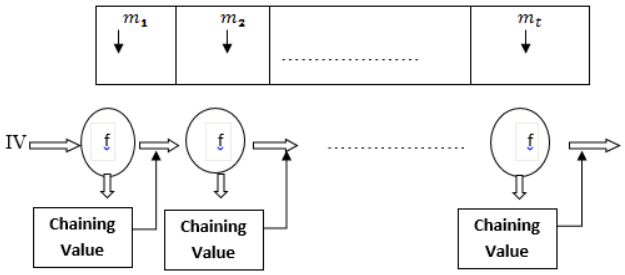

Figure 2. Process of Merkle Damgard Construction.

Due to the iterative structure of MD construction arbitrary length message can easily be processed by the hash function. Also the main strength (Daum, M. May 2005) of the hash function is, padding the message length and using a non-zero IV and increase difficulty for an attacker. An important property of this construction is that if the compression function is collision resistance the hash function preserve this property. Some Example of Hash function based on MD construction are-MD4, MD5, SHA-0, SHA-1, SHA-2.

Though Merkle-Damgard construction is the most popular structure, with passing years several weaknesses found in this construction method so there are some modified version was proposed in recent years.

Wide pipe and Double wide pipe Construction -

Stefen Lucks introduced amore secured version of Merkle Damgard construction in 2004. The main difference with the MD construction is in the internal state size (Figure 3). It uses large size chaining value that could increase the complexity of the attack depending on the chaining values (Kocak, O. 2009). Lucks introduce another construction double wide pipe (Lucks, S. 2005) construction (Figure 4), where the length of the internal chaining value is twice than the length of the message digest. The input IV is divided into two halves where the first half is directly inputed to the compression function and the second half XOR-ed with output of that compression function (Kocak, O. 2009). It is a faster process than the previous one.

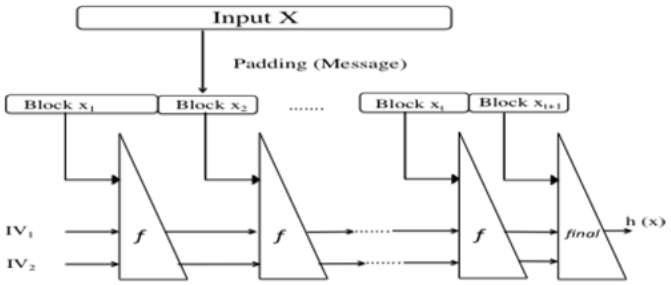

Figure 3. Wide Pipe Construction

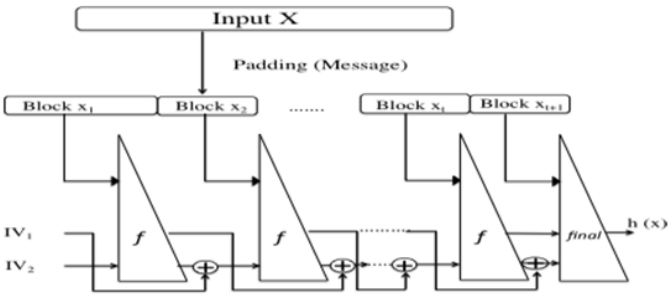

Figure 4. Double wide pipe construction

Fast wide pipe Construction –
(Nandi, M. and Paul, S. 2010) proposed the fast wide pipe construction. It consists of two parallel iteration with two different initial values (*IV* and $\overline{IV}$) and in the final iteration the outputs are mixed before obtaining the result.

2) HAIFA Construction.
(Biham, E. Dunkelman, O. 2006) designed HAsh Iterative FrAmework in. The compression function f takes only message block x_i and chaining value h_i in the MD construction. In HAIFA construction there are two extra input – the number of bits hashed so far (b) and a salt value (s). The number of bits hashed so far is included as an input to prevent the fixed point attack and salt is a precaution against attacks which has pre-computation phase. Once a fixed point is found such that h= C_{MD} (h,x) the attacker can use it as many times as want but now even if finds a fixed point of the form h= C_{HAIFA} (h,x,b,s), it cannot be concatenated many times as the number of bits hashed so far will be change. Sometimes the attacker pre builds some structure (either messages or chaining values) and after knowing (X, H(X)) pair produce a collision or second preimage, but now the attacker should know the salt value which is a random generation.

3) Sponge Construction method:
It was designed by Guido Bertoni, Joan Diemen, Micheal Peeter and Gilles Van Assche to replace Merkle - Damgard construction in 2007. It was built on a function which can be expressed as a random function or random permutation. The function operates on fixed size bits b=r+c. Unlike the compression function in MD and HAIFA construction it maps l bit input to l bit output (Matusiewicz, K. August 2007). (variable length input to variable length output). It consists of two phases – (i) in the absorbing phase data is input to the sponge iteratively. The message block is XOR-ed with the first r bits of the state of the function and input to the next function. (ii) the output of this phase is the first r bits of the iteration (f) and the iteration is applied until the desired hash size is achieved. The other part of the state denoted as 'c' is the capacity of the function. The security of Sponge construction depends on c, hash size n and function. Higher value of r increase speed and higher value of c increase security. Figure 5 describes the two phases.

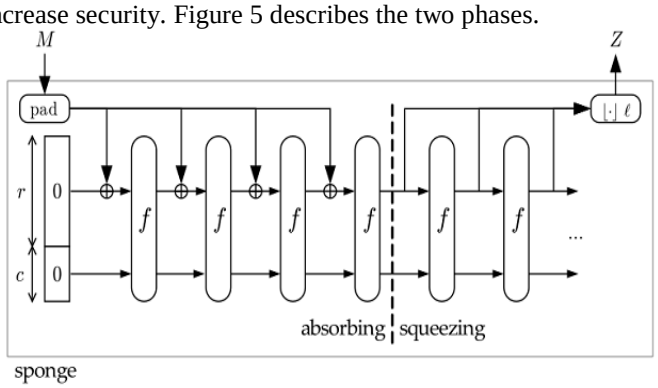

Figure 5. Sponge construction

C. *Classification based on the Methods used in the compression function*

Usually hash functions are built on two components – the domain extender and the compression method. Compression functions are take fixed length inputs and the domain extenders are algorithms that use compression function with arbitrary length inputs. As discussed in the previous section there are basically three construction methods and construction methods iterates compression function to map arbitrary length input so there must be a compression function with the construction method. Compression functions should also maintain the security properties of hash functions.

1) Hash function based on block cipher

Here some secure symmetric key block cipher (Thomsan, S, S. 2008) (i.e AES, DES) can be used as the compression function rather than using a newly created compression function specifically for this purpose. This approach reduce implementation cost as well as minimized designing effort. As the compression function is replaced by any encrypting cypher, the message block is used as the key. The message is divided into blocks according to the block cipher size. Each message block is used for some number of encryption. The number of block processed in each encryption is the rate of this construction.

The simplest type of construction is single block length hash function and little more complex type is double block length. In case of block length construction the size of the output i.e the hash value is equal to the cipher text size of each block. Davies–Meyer, Matyas–Meyer-Oseas, Miyaguchi–Preneel are example of single block length construction. The first block cipher based hash function is based on DES where the size of the digest is only 64 bits. Although this construction is not so interesting because the output is not large enough to give a sufficient security level in terms of generic attacks. The output of the Double block length hash function will be almost 2n if n is the size of the input of the block cipher. MDC – 2, MDC- 4 are example of double block length.

Davies–Meyer:

The message block (m_i) serve as a key and the previous hash value (h_i) as the plaintext and h_{i+1} is feed forward with the next output. Figure 6 is a block diagram of this construction.

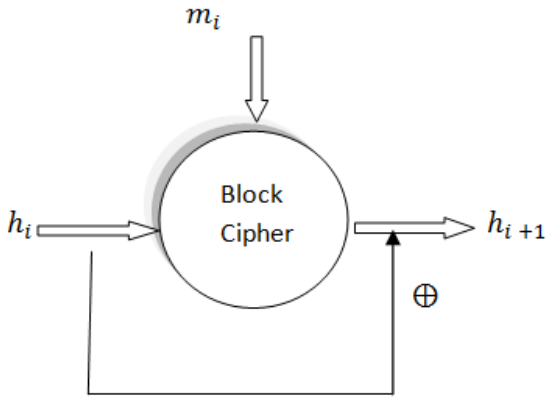

Figure 6. Block diagram of Davies Meyer

Matyas–Meyer–Oseas:

Here the role of h_i and m_i are changed. The message block m_i serve as a plaintext and the previous hash value serve as a key. As the bit length of the previous hash value may not be same as the bit length required for the key to block cipher a function (g) is used that maps the variable length h_i to required length for key (as described in Figure 7).

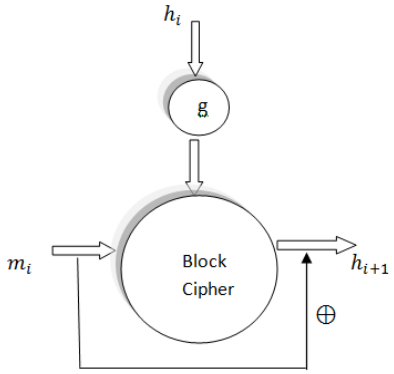

Figure 7. Block diagram of Matyas–Meyer–Oseas

Miyaguchi–Preneel

In Miyaguchi–Preneel, the only extension is that the previous hash value is also XOR–ed with the next step output. Figure 8 is a block diagram of Miyaguchi–Preneel.

MDC – 2

Manipulation detection Code algorithm implement two parallel Matyas–Meyer–Oseus scheme. Figure 9 describes the process.

MDC – 4: It extended MDC-2 by taking four separate iteration of Matyas–Oseas scheme.

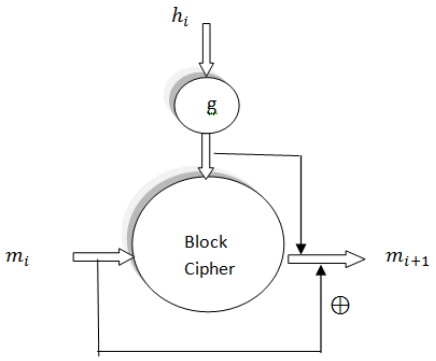

Figure 8. Block diagram of MIyaguchi Preenel

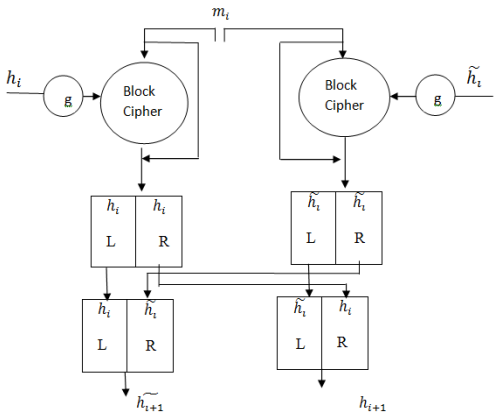

Figure 9. Block diagram of MDC-2

2) Hash function based on Stream Cipher

Though the restriction of output size in case of block cipher based hash function can be overcome by using double block length but the speed efficiency could not be achieved. Another approach is stream cipher (Kocak, O. 2009) based hash function which can reduce total cost of implementation and can overcome the disadvantages of block cipher. But in terms of security this approach is not so reliable. Stream Cipher based hash functions are very weak in case of known input attack. The general construction of stream cipher based hash function was first introduced by Golic. The main core part of a stream cipher based hash function is a key stream generator which takes message in blocks generated by another function and mixes it to generate the key stream. The first stream cipher based hash function was Panama proposed in 1998.

3) Moduler Arithmetic Based Hash Function

As a compression function, a hash function can use modular arithmetic also. This type of design is based on hard mathematical problems such as number theory problem, factorization problem, and discrete logarithm problem. Modular arithmetic makes it easy to scale the security as we can choose modulus of appropriate length. It has a disadvantage that it is slow when compared with block cipher or stream cipher. The first modular arithmetic based hash function was using RSA algorithm and some chaining process with it. Another example is MASH-1.

4) Hash function based on Stream Cipher

Permutation is also a popular method used as compression methods. Here few permutation rounds on the block cipher have been performed. To implement this approach it is very important to apply sufficient length of modular arithmetic. SHA-3 is a big and successful example of permutation based hash function.

III. CRYPTOGRAPHIC HASH ALGORITHMS

There are multiple cryptographic hash algorithms based on different construction methods. Each of them uses different steps rounds and initial value.

A. *MD2*

MD2 (Mullar, F. 2004) was designed in 1989 by Ronald Rivest. It does not follow Merkle Damgard construction rather it is based on simple permutation based iteration. It uses 8- bit instruction that is useful for old architectures and it is no longer considered to be secure.

Input : 16- byte message block

Output: 16- byte Hash value

As the first step, the message is first padded so that its length become multiple of sixteen bytes. Then a sixteen byte checksum is appended. This step uses a 256- byte table permutation. The initial value of the buffer (IV) is initialized with zero and a 48- byte auxiliary block is used to compute the chaining value. The algorithm consist of a loop where it permits each 16- byte input message eighteen times in the auxiliary block and at the end again 256- byte permutation used. The message digest produced as the first partial output of the auxiliary block.

B. *MD4*

MD4- a cryptographic hash function, developed by (Rivest, R, L. 1990). The main motivation behind this was the Merkle Damgard Construction.

Input : 512 bit message (Sixteen 32 bit Words)

Output: 128 bit Hash value

C. *MD5*

After the first weakness found in MD4 Ron Rivest came with a stronger version of MD4 in 1992 as MD5.

Input : 512 bit message (Sixteen 32 bit Words)

Output: 128 bit Hash value

Sainger, N and Agarwal, A, P. proposed several improvements of MD5 over MD4.

D. SHA-0

The first cryptographic hash function published by NIST is SHA – 0 in (NIST. May 1993). It was also based on the Merkle Damgard construction .
Input : 512 bit input message.
Output : 160 bit hash value .

E. SHA-1

NIST published the first revised version of SHA-0 as SHA – 1 (NIST. April 1995) where the procedure for extended message word has changed and one extra rotation added to the procedure.

F. SHA-2

With the popularity of cryptographic algorithm AES (Advanced Encryption Standard), which support larger and variable key size there was a need of hash functions to match wit the larger output sizes.
According to the demand NIST came up with a new series of hash function. The first algorithm of SHA-2 family was published in 2001. In all of the algorithm the first step is padding a message word to make it multiple of 52 bit or 1024 bit and use eight registers with diffrent initial value.

G. SHA-3

The design principal of SHA-3 (NIST. August 2002) is totaly different. It is based on the sponge construction and the main property of it is it supports variable length input and variable length output.
This construction is based on two parameters, b(bitrate) and c(capacity) and the sum of b and c determine the width of the permutation. Based on the bit width of the permutation there are seven members in SHA-3 family.
The core function contains 24 rounds and each has five sub steps. As SHA-3 is based on sponge construction it has three phases- Initialization, absorbing, squeezing. In the initialisation phase the state matrix is initialized with zero. The 24 rounds are absorbing phase and in the squeezing phase we get desired length hash value by trancating the state matrix.

H. PANAMA

Basically it is a stream cipher and can be used as a hash function. PANAMA was published in 1998 and desighned by (Daemen, J and Clapp, C. 1998). It maps a message of arbitrary length to a hash result of 256 bits. Like all other hash function here also a padding algorithm needed. The input string is padded to make it multiple of 256 by appending a single 1 and rest of 0-bit. PANAMA algorithm consist of a state(544 bit) and a buffer (8192 bit). The algorithm can be described in three parts –
Initialization – the state is denoted by seventeen 32-bit word and the buffer buff is initialized to zero.
Push mode – the eight word input is applied. The buffer is responsible to inject the input bits over a number of iteration.
Pull mode – in this stage the hash value is retrieved. In each iteration a eight word output can be recieved.

I. MASH-1

MASH-1 is a hash function based on modular arirhmetic. It was presented in Part-4 of ISO/IEC 10118 standard. It is based on the RSA cryptographic algorithm. Let n is the RSA modulus used by the algorithm. The length(l) of the chaining variable is equal to the largest multiple of sixteen and strictly smaller than the length of the RSA modulus. The input string is padded in the right most to make it multiple of l/2 bit length. Then it is devided into t number of half blocks. Like other hash functions the hashing process consist of a compression function that is applie iteratively.
In the following table (Table-II) a comparative analysis of different hash function algorithms is presented.

Table I. Comparison of Hash function algorithms

Hash Functions	Year of Publication	Designer	Internal Word Size(bit)	Meesage block length (bit)	Internal state size	Output (bit)	Rounds	Construction method
PANAMA	1998	J. Daemen & C. Clapp	32	Stream based	544 17×32	256	--	Buffer based iteration, combination of two modes
MASH-1	1998		Depend on the RSA Modulus used by the algorithm	--	--	Not fixed	Moduler Arithmetic Based.	based on factorization problem of an RSA modulus along with compression functions.
MD2	1989	Ronald Rivest	8	128	384 48× 8	128	18	Basic iteration method using checksum and permutation.
MD4	1990		32	512	128 4× 32	128	3 (48 steps)	Merkle Damgard Construction
MD5	1992		32	512	128 4× 32	128	4 (64 steps)	

SHA-0	1993	NIST	32	512	160 5× 32	160	4 (80 steps)	
SHA-1	1995		32	512	160 5× 32	160	4 (80 steps)	
SHA-2 ⇧ **SHA-224** **SHA-256** **SHA-384** **SHA-512**	2002		32/64	512/1024	256/ 512 5× 32 8 × 64	224/ 256/ 384/ 512	64/80 steps	
SHA-3	2012	G. Bertoni J. Daemen M. Peeters G. V Assche	64	1600		224/256/38 4/512	24	Sponge

IV. ATTACKS ON HASH FUNCTION

If any of the properties of the hash function can be break in any way then that will be considered as attack on Hash function. Minimum Security Requirement for a Hash Function given in Table I.

Table II. Minimum security requirements of hash function

Attack	***Security Boundary***
Preimage	2^n
Second-Preimage	2^n
Collision	$2^{n/2}$

i.e the preimage. Second preimage and collision of a hash function should not be found with probability less than $2^n, 2^n$ and $2^{n/2}$. Attacks on Hash function can be classified in three categories.

Figure 10 describes a general classification of all kinds of possible attacks on hash function.

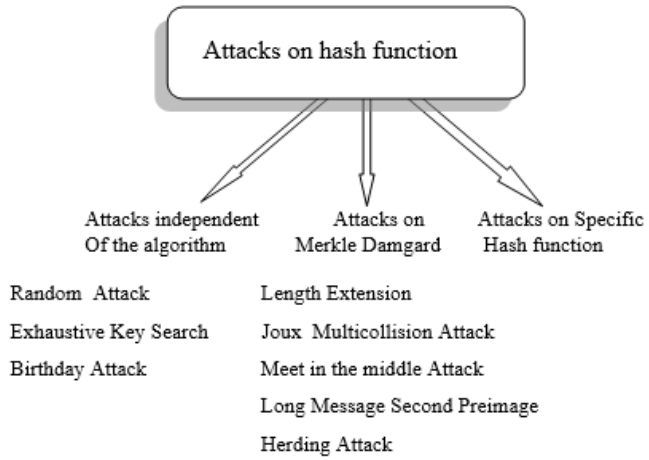

Figure 10. Classification of attacks on Hash function

A. *Attacks independent of the algorithm*

This class of attacks are some general methods that do not depend on the knowledge of the algorithm. This can be applied to any hash function that maps set of messages uniformly to set of independent random variables.

Random Attack

The attacker selects a message and or part of message randomly and hopes the hash value will be unchanged. In a random attack the probability of success will be $1/2^m$ if m is the no. of bits for the hash value (Kelsey and Kohno, 2006).

Exhaustive Key Search

It is a cryptanalytic attack and can be applied on any keyed hash function where attacker knows the plaintext – MAC pai and precompute the MAC to guess the key. The attacker systemetically searches for all possible combination untill the correct combination found. This is usefull when finding any weakness is not possible(Kelsey, Jand Kohno, T. 2006).

Birthday Attack

Birthday Attack is used as a basic method to find collision. The attack is based on the famous birthday problem. The birthday problem is associated with the probability and can be explained from four perspective (Forouzan, B, A. and Mukhopadhyay, D. 2010).

Problem 1: What is the minimum number of instances, k, such that it is likely that at least one instance from a set is same with a predefined value.

Problem 2: What is the minimum number of instances , k, such that at least one value equal with the other selected one within a set.

Problem 3: What is the minimum number of instances , k, such that at least any two instances are equal within a set.

Problem 4: What is the minimum number of instances , k, such that at least one instance from the first set is equal to another instance in the second set.

The first two problems are related to the preimage and second preimage attack and the third and fourth problem are related to collision attack.

Say, The attacker is looking for any collision for birthday among N people then the probability will be

$$P_{collision}(N) = \left(\frac{365}{365}\right)\left(\frac{364}{365}\right)\left(\frac{363}{365}\right)\cdots\cdots\left(\frac{365-N+1}{365}\right)$$

Here N=23 people are enough to have a match in birthdays with probability greater than ½

If attacker is looking for a specific collision , then the probability will be $P_{collision} = 1 - (\frac{364}{365})^N$

B. *Attacks independent of the algorithm*

Soon after Merkle- Damgard construction proposed some attacks and weaknesses have been identified.

Length Extension attack :

This can be applied to keyed hash functions that accepts a message and secret key pair and generate the hash value (Kocak, O. 2009). The attacker who knows the part of the message and the final hash value but without knowing the secret key extend the length of the message and compute the hash value and sends it.

Joux- Multicollision attack :

Multicollision is finding multiple different messages that can be maps to the some hash value.Joux shows that multicollision can be easily found with Merkle Damgard construction. He assumes that there ia a collision finding algorithm, that can use birthday paradox or some other method. The algorithm (Danda, M, K, R. 2007) can find collision for the compression function that takes $2^{t/2}$ computation of the compression function.

Long Message second preimage Attack :

At Eurocrypt 2005, kelsey and schneir presented a second preimage attack (Kocak, O. 2009) on all hash functions based on Merkle Damgard Construction .

Attacker tries to find out a second preimage M' for M where $M \neq M'$ and $H(M) = H(M')$ with an effort less than 2^t computation of H. In other words the attacker tries to find out a preimage for long message with a linking message M_{Link} where the digest of f_{IV} (the initial value) of the linking message block is one of the intermediate chainig value h_i

Meet in the Middle Attack :

It can break the preimage resistance property of hash function but more specifically it give second preimage. The attacker can construct a message with prespecified hash value (Boer, B and Baseliners, A. 1992). The attacker starts forward from the initial value and backward from the hash value and follow some iteration step with multiple variation of thm in aim of meeting at a predefined chaining value.

Herding Attack -

It is an attack on Merkle Damgard construction and based on the property chosen target forced prefix. In this attack the attacker chosen a target Hash value (H) through some pre computation, and then the challenges came with some prefix (P) and the attacker has to produce some string (S) so that hash(P||S) = H.

The attacker constructs possible chaining values from which the attacker knows how to reach target hash value. The attacker build a diamond structure (In data structure terms it can be called as a tree structure) with multiple single block messages (m_j) and value of f(h_i ,m_j) for all i and j. The attacker will find out the collision in intermediate state of the diamond structure and construct string S with the path in the diamond (where the root is the target hash).

C. Evolution of Hash functions and Analysis of Attacks on specific hash functions

During 1968 uses of Hash function started to protect password but at that time hash function did not have the capability of compression. Then in 1976 after the publication of public key cryptography use of cryptographic hash function started where hash function is based on block ciphers.

The first hash function was MD2 designed by Ronald Rivest in 1988. This is the first member of MD-family where the family contains a series of hash functions. Here MD stands for message Digest. The first attack against the full MD2 was published by Mullar in 2004 and it breaks preimage property. Then in 1989 Merkle Damgard independently describes construction methods for hash functions. In 1990 a new member called MD4 added to the MD-family. Within just one year after the publication of MD4, an attack on the last two round of the MD4 was proposed by (Boer, B and Bosselaers, A. 1993). There was a series of attacks against MD4. Against the first two rounds of MD4 Vaudenay described another attack (Vaudenay, S. 1995). In 1996 a collision attack was found by Dobbertin with probability 2^{22} on full round of MD2 (Diffie, W and Hellman, M, E. Nov 1976). Wang et al also found a collision attack on MD4 and Sasaki et al presented an improved version of this after publication of this. Due to all these attack MD4 was no more secure hash function. But R Rivest realized the weaknesses in the design of MD4 soon after its publication and proposed a more improved version - MD5 based on the same construction method.

The first attack on MD5 was presented by Boer and Bosselaer in 1993 (Boer, B and Bosselaers, A. 1993). It shows collision of two messages but two different initial values (IV). In 1996 Dobertain came with another collision attack but with a chosen initial value. Finally, in 2004 a term led by Wang et al announced collision for MD5 with real initial value as well as an attack on a couple of hash function like MD4, HAVAL-128, RIPEMD.

NIST published their first hash function as SHA-0 after two years of the publication of MD5 i.e. in 1993. Some weakness of SHA-0 was found internally and after two years NIST published improved version as SHA-1 in 1995 and SHA-2 in 2002. In 2005 Rijmen and Oswal published an attack on a reduced version of SHA-1. First attack full 80 step of SHA-1 was presented by X. Wang and her team with a complexity2^{69}.

In the time of 2002-2005 multiple attacks on different hash functions were published and till then the hash functions were mostly based on the same construction method. This motivates NIST to move to a new construction method. In the year 2007, NIST announce a competition for a new set of hash function and receive 64 entries around the world within 31st October 2008. In October 2012 NIST announced Keccak as the winner of the competition and accept the algorithm as SHA-3 which is based on a new construction, sponge construction.

In addition to MD and SHA family in the literature, there exist a large number of hash functions. Y L Zheng presented HAVAL-128 in 1992. RIPEMD ((Race Integrity Primitives Evaluation Message Digest) also appeared as a family as there exist different versions of this algorithm for 128, 160, 256, 320-bit hash digest.

Preimage Attack on MD2

The attack shows that MD2 does not reach the ideal security level of 2^{128} (Mullar, F. 2004). The complexity of this attack is almost 2^{73} evaluation of the compression function. The attack can be described in two parts. The first part gives multiple preimages by using symmetry between matrices that used to compute the chaining value and the second part tries to find the preimages that compiles with the checksum function.

Attack On the last two round of MD4

Among the three round of MD4 only the last two rounds considered in (Boer, B and Bosselaers, A. 1992). It shows that for a given input two different message block hashed to the same output.

According to the architecture of the MD4 algorithm the first and last four elementary operations of the second and third round used same message word and the 8 middle elementary operations are also used the same message word. The idea is to construct two different messages that has same value in the

position which words are used by the first and last four operations. The message word only differs in the remaining eight message word that uses by the middle eight operations. Also the two message word should give two alternatives after 8^{th} and 24^{th} elementary operation but should give same value for the 12^{th} and 28^{th} elementary operation. Hence the input will be one with two different messages and hashed to the same output. In this problem multiple equations has to be solved for the unknown values of the message words for every dual possibility of the middle 8 elementary operation.

V. CONCLUSIONS

Cryptographic hash functions have broad applications in the domain of computer security, and programs built on top of cryptographic hash functions have the ability to help a system administrator detect changes of valuable data on his or her network. These concepts are particularly relevant in the growing online world, where every message sent across the wire can be worth money, and every file on a server is a valuable resource. Without safeguards such as those afforded by hash functions, data would be extremely vulnerable to attack. The security of hash function became an important topic as almost every day the world of hash function is facing a new attack. In this paper, an extensive study on cryptographic hash functions with their applications, properties and detailed classification is presented. It also describes a general classification of all kinds of possible attacks on hash function analyses some attacks on specific hash functions.

VI. REFERENCES

[1] AlAhmad , M, A. and Alshaikhli, F. (July 2013) 'Broad View of Cryptographic Hash Functions', Vol. 10, Issue 4, 239 – 246.

[2] Biham, E. Dunkelman, O. (2006) 'A Framework for Iterative Hash Functions: HAIFA', *In Proceedings of Second Cryptographic Hash Workshop*, Krakow.

[3] Boer, B and Bosselaers, A. (1992) 'An Attack on the last two rounds of MD4', CRYPTO 1991, *in Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, 194-203.

[4] Boer, B and Bosselaers, A.(1993) 'Collision for the compression function of MD5', In Eurocrypt, *Lecture Notes in Computer Science, Springer, Vol. 765*, 293 – 304.

[5] Daemen, J and Clapp, C. (1998) 'Fast Hashing and Stream Encryption with PANAMA', Springer-Verlag, Berlin Heidelberg.

[6] Damgård, I.(August 1989) 'A Design Principle for Hash Functions' *CRYPTO 1989, Springer LNCS*, Volume 435, 416-427.

[7] Danda, M, K, R. (2007) 'Design and Analysis of Hash functions', *Master Thesis*, Victoria University.

[8] Daum, M. (May 2005) 'Cryptanalysis of the MD4 family', Bochum.

[9] Diffie, W and Hellman, M, E. (Nov 1976) 'New directions in cryptography', *IEEE Transactions on Information Theory*, Volume 22 Issue 6, 644-654.

[10] Forouzan, B, A. and Mukhopadhyay, D. (2010) 'Cryptography and Network Security',McGraw Hill Education (India) Private Limited, New Delhi.

[11] Kelsey, Jand Kohno, T. (2006) 'Herding Hash Functions and the Nostradamus Attack', *Advances in Cryptology - EUROCRYPT 2006, 25th International Conference on Theory and Applications of Cryptographic Techniques*, St. Petersburg, Russia, May 28 - June 1, page 183-200.

[12] Katz, J. and Lindell, Y. (2011) 'Introduction to Modern Cryptography', Chapman & Hall, CRC.

[13] Kocak, O. (2009) 'Design and Analysis of Hash Functions', Master Thesis, Middle East Technical University.

[14] Lucks, S.(2005) 'A Failure-Friendly Design Principle for Hash Functions', University of Mannheim, In ASIACRYPT Germany. Springer, Pages 474-494.

[15] Matusiewicz, K.(August 2007) 'Analysis of Modern Dedicated Cryptographic Hash Functions', *Ph**D**Thesis*, Centre for Advanced Computing, Algorithms and Cryptography Department of Computing Division of Information and Communication Sciences Macquarie University.

[16] Merkle, R, C.(1989) 'One Way Hash Functions and DES', *Crypto '89: Proceedings on Advances in cryptology*, 428–446.

[17] Mullar, F. (2004) 'The MD2 Hash Function is not one-way', *Advances in Cryptology - ASIACRYPT 2004, Volume 3329, Lecture Notes in Computer Science*, 214-229.

[18] Nandi, M. and Paul, S. (2010) 'Speeding Up The Wide-pipe: Secure and Fast Hashing', National Institute of Standards and Technology Security Technology Group.

[19] NIST. (May 1993) 'Secure hash standard',*Federal Information Processing Standard*, FIPS-180.

[20] NIST. (April 1995) 'Secure hash standard', *Federal Information Processing Standard*, FIPS-180-1.

[21] NIST. (August 2002) 'Secure hash standard',*Federal Information Processing Standard*, FIPS 180-2.

[22] NIST. (August 2015) 'SHA - 3 Standard: Permutation-Based Hash and Extendable -Output Functions',*Federal Information Processing Standard*, FIPS 202.

[23] Rivest, R, L. (1990) 'The MD4 message digest algorithm', *Advances in Cryptology-CRYPT0' 90, Volume 537, Lecture Notes in Computer Science*, Santa Barbara, 303–311.

[24] Sainger, N and Agarwal, A, P. (July 2014) 'Modification in Hash Function from MD4 to SHA-3', *International Journal of Emerging Research in Management &Technology*, Volume 3, Issue7, 53 – 60.

[25] Thomsan, S, S. (2008) 'Cryptographic Hash Function', *PhD Thesis*, Technical University of Denmark.

[26] Vaudenay, S. (1995) 'On the Need for Multi-permutations: Cryptoanalysis of MD4 and SAFER', In Bart Preenel, edition, Proceedings of Fast Software Encryption (2), LNCS 1008, Springer-Verlag, 286 – 297.