A Review on Cryptographic Hashing Algorithms for Message Authentication

Nishant Sahni
Computer Engineering
Department, Mukesh Patel
School of Technology
Management & Engineering,
NMIMS University, Mumbai,
India

ABSTRACT
The main purpose of Message Authentication is to prevent manipulation of the message which is sent. MAC stands for Message Authentication Code which is also known as "Integrity Check Value" or "Cryptographic Checksum".

The basic objectives of a hash function are to:

• Prevent finding a message from a given hash value (Inversion)

• Prevent finding two messages with the same hash value (Collision)

On the other hand, Message Authentication Codes are mainly to prevent forgery. Thus, using hash functions for Message Authentication may get a bit complex as hash functions do not have the in-built functionality of a key.

In this paper, we discuss a few popular cryptographic hashing algorithms and compare their performance with respect to each other.

Keywords
Hashing Algorithms, Authentication Code, MD5

1. INTRODUCTION
Let A and B be two parties transmitting information between each other. When A sends a message to B, an "authentication tag" is generated by the MAC algorithm and is appended to the message. This authentication tag is a function of the information to be transmitted and a "shared secret key". When B receives the message, it uses the same algorithm and key to re-compute the authentication tag and checks if its value matches the tag attached to the message sent by A. If it matches, we can be sure that the information was not altered on its way from A to B. This is a basic construct of a cryptographic hashing algorithm. We will now discuss in detail, the working of some popular algorithms.

2. BRIEF DESCRIPTION

2.1 Nested Message Authentication Code (NMAC)
This works on the concept of nesting a function. Let k = (k1,k2) where k1 and k2 are keys to the function F, i.e., random strings of length 'l' each). The MAC function NMAC(x) works on inputs x of arbitrary length as-

NMACk(x) = Fk1(Fk2(x)) [1]

The construction is stated to be simple and efficient. The cost of the internal function is the same as that of hashing with a keyless hash function. The main cost is the outer function which is involved in only a single iteration.

2.2 Hashed Message Authentication Code (HMAC)
The HMAC scheme improvises in a few shortcomings of the NMAC scheme. The HMAC scheme involves a single 'l' bit long key k, whereas in NMAC two keys are present. This makes key management a lot simpler. Also, the NMAC scheme directly accessed the code for the compression function to key the initial variable (IV).

The function of HMAC is as follows:

Let F be the hash function initialized with its usual fixed IV. The function HMAC works on inputs x of arbitrary length and uses a single random string k of length 'l' as its key:

$$\mathrm{HMACk(x)} = \mathrm{F}(\overline{k} \oplus \mathrm{opad}, \mathrm{F}(\overline{k} \oplus \mathrm{ipad}, \mathrm{x}))$$ [1]

where $\overline{k}$ is the completion by adding 0's of k to a full b-bit block-size of the iterated hash function, opad and ipad are two fixed b-bits constants (the "i" and "o" are mnemonics for inner and outer), and $\oplus$ is the bitwise Exclusive Or operator [1].

2.2.1 Advantages and Disadvantages
The authors state that it is possible that some attacks may work against HMAC but fail against NMAC. Also, it is stated that having a single l-bit long key instead of two randomly chosen key does not compromise on the security. Overall, NMAC is a faster scheme to implement. On the other hand, HMAC requires only one l-bit long key, as opposed to two keys as in NMAC and hence simpler computations.

Some disadvantages of using NMAC scheme are as follows:

• The underlying hash must be modified to key the Initial Variable (IV) which is not too difficult in software [1].

Some disadvantages of using HMAC scheme are as follows:

• The HMAC function is slower than the NMAC function as it requires two more computation of the compression function.

• If the length of key is less than l-bits, the strength of the keyed IV is reduced.

• A periodic refreshment of keys is required.

• It is inconsistent as there are some attacks that work against HMAC but fail against NMAC.

2.3 Geometric Hashing

Geometric Hashing is a type of computer vision tool used to detect an object model in a visual scene. A basic model is a set of 2D points which also forms a set of 2D coordinates.

M={Pi=(xi,yi)} (i=1….n) [2]

The main objective is to find the object model in a collection of 2D points called the measurement points.

2.3.1 Preprocessing:

1. Two points of the model are chosen P0 and P1. These are denoted the base points.

2. A 2D transform is found that maps P0 to the origin and P1 to the coordinates (0,1).

3. All model points are transformed using the 2D trans-formation found in Step 2.

2.3.2 Recognition:

After a set of measurement points are obtained from a visual scene, two measurement points are chosen at random and the normalization procedure is applied for the model points as shown above.

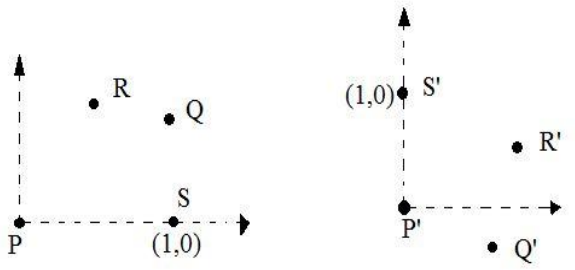

Fig 1: Preprocessing

Some issues one might face during the preprocessing and recognition stage are as follows:

• Noise may get introduced and affect the accuracy of the measurement points.

• In case of noisy measurement points, taking just 2 as the basis for normalization may prove faulty and will lead to incorrect recognition.

To solve these issues, in reality, the recognition process must be carried out repeatedly and with different measurement points as the base points each time.

2.3.3 Geometric Hashing and Watermarking

For geometric hashing to work for watermarking, one basic requirement is that the code embedding method must affect the discrete image values [2]. These can be further associated with 2D coordinates. Also, while decoding, the set of marked values must be detectable. DCT Watermarking is one of the watermarking techniques which fulfils these requirements.

2.3.4 Geometric Hashing and DCT Watermarking

DCT stands for Discrete Cosine Transform. Here, the image blocks for encoding are not chosen in a random fashion. The pattern for choosing the blocks is predefined and has undergone a random transformation. This predefined pattern of image blocks forms a model. It is defined by a set of 2D coordinates which represent the position of the image blocks. Image block coordinates are assigned to any transformed version of the model as well.

In contrast to the original watermarking scheme which uses Random Generator Seed (RNGS), Geometric Hashing involves the extracting of all image blocks which maybe possibly marked for decoding purposes [2]. These blocks, when extracted, form the measurement set. The decoder has to traverse each and every image block to check if the specific model is present.

To sum it up, geometric hashing checks whether the model was found within the measurement sets. If it is found, the image is watermarked.

2.3.5 Advantages and Disadvantages

2.3.5.1 Advantages:

• The main advantage of using geometric hashing for watermarking is that randomization of the watermark can be done without having to maintain large amounts of information such as Random Generator Seed (RNGS) for decoding purposes [2]. If an appropriate watermarking technique is chosen, geometric hashing can perform well against attacks.

• It helps prove and preserve the authenticity of an image file.

2.3.5.2 Disadvantages:

• Noise might seep in and the measurement set of possibly marked blocks may consist of many noise blocks. This may prevent smooth performance. Nullifying low frequency DCT coefficients was found to reduce noise in the measurement set.

• Image block coordinates are discrete. Thus the rounding off of the transformed model coordinates leads to the introduction of digitization noise. numbers.

2.4 Message Digest 5 (MD5)

2.4.1 Prefix Approach

Here a key is simply concatenated with the message and passed through the MD5 hash function where the key comes first and then the message to be hashed. It can be represented as follows-

MD5 (k . m) [3]

2.4.2 Damgard/Merkle Iterative Structure

The Damgard/Merkle structure is one where the compression function is repeatedly applied to each of the successive message blocks. We take a 128bit chaining value and a 512bit message block as the input to the hash function. When the compression function is applied, the output produces another 128bit chaining value which is used as input along with the subsequent 512bit message block. The compression function is iteratively applied to all the message blocks. Before we begin this process, the message is padded to obtain a multiple of 512bits so that the message can be broken down into equal blocks of 512bits each. After processing the last message block, the final chaining value which is obtained as output is the hash of the message. This entire process can be illustrated through the following image:

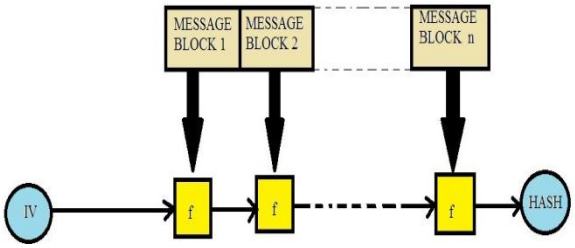

Fig 2: Damgard/Merkle Iterative Structure

2.4.3 A Few Other Approaches

The authors recommend three other methods which can be employed for MD5, mainly for IP Security (IPSEC). They are as follows:

1. MD5 (k1 . MD5 (k2 . m)) [3], where k1 and k2 are independent 128bit keys.

2. MD5 (k . p . m . k) [3], where k is a 128-bit key and p is 384bits of padding.

3. MD5 (k . MD5 (k . m)) [3]$, where k is a 128-bit key.

In the first approach, two independent keys k1 and k2 of 128bit each are taken and the MD5 hash function is called recursively to obtain the hashed value. Here k1 and k2 are obtained as follows:

k1 = MD5 (k . α) and

k2 = MD5 (k . β) [3],

where α and β are distinct constants.

The third approach is very similar to the first one. Here a single key k of 128bits is taken and the MD5 hash function is called recursively to obtain the hashed value. This approach is more vulnerable to attacks than the first one as only one key is used.

In the second approach we pad the message with p of 384bits and a single key k of 128bits is used. Very short messages are more vulnerable to attacks, thus the padding is essential to ensure the appropriate length of the message.

2.4.4 Advantages and Disadvantages

2.4.4.1 Advantages:

• MD5 is much easier to implement as compared to other hash functions.

• MD5 is easily available.

• It provides good resistance against attacks.

2.4.4.2 Disadvantages:

• Not all approaches to obtain MAC through MD5 are attack resistant.

2.5 Secure Hashing Algorithm 1 (SHA1)

2.5.1 Message Padding

The message can be of variable length. SHA-1 digests the message in the form of message blocks each of 512bits. To be able to break the message into multiple blocks of equal length, we must pad the message. SHA-1 sequentially processes these blocks of 512bits each. The message can be padded by putting a "1" on the right or 'n' number of "0"s [4].

2.5.2 Computing the Message Digest

There are two main methods which SHA-1 adopts to obtain the message digest.

Method 1:

• First the message is padded before digesting as described before.

• It involves two buffers, each of which have five 32bit words.

• It also involves a sequence of eighty 32bit words.

• The words of the first 5-word buffer are named A,B,C,D and E.

• The words of the second 5-word buffer are named H0,H1,H2,H3 and H4.

• The words of the eighty word sequence are named W(0),W(1)…….W(79).

• To obtain the message digest, the individual message blocks of 512bits each processed in order.

• After processing, the message digest is the 160bit string given by the 5 words H0 H1 H2 H3 H4 [4].

• It has a much lesser execution time than Method 2 as the address computations are comparatively simpler.

• Uses more storage than Method 2.

Method 2:

• In this method, instead of using 80 32bit words we use only W(0),….,W(15).

• Here the 16 32bit words form a circular queue.

• The message digest is given by the 5 words H0 H1 H2 H3 H4 [4].

• Thus, using the second method saves 64 32bit words of storage.

• But the execution time is much more than Method 1 due to the complexity of address computations.

2.5.3 Advantages and Disadvantages

2.5.3.1 Advantages:

• SHA-1 is easy to implement as compared to a few other hashing algorithms.

• SHA-1 is easily available.

• It provides good resistance against attacks.

• It is more secure than MD5.

• It is less likely to have collisions in case of SHA-1.

2.5.3.2 Disadvantages:

• SHA-1 is more complex to implement than MD5.

3. INFERENCE

HMAC and NMAC use already existing hashing algorithms along with a key to provide the hashed output. These pre-existing hash functions could be MD5 or SHA-1. This is because these algorithms can be easily obtained and are widely implemented. The objective is to maintain the integrity of the pre-existing hash functions and also simultaneously providing the added security of a key. Also, the underlying hash function can be replaced with minimum effort if a function more secure than MD5 or SHA-1 is found.

HMAC is an improvisation of NMAC and is more secure. But on the other hand, HMAC is much slower than NMAC. Also, NMAC involves the use of two keys whereas HMAC involves the use of a single key.

MD5 and SHA-1 are the most commonly used hashing algorithms. This is due to their ease of implementation and high availability. HMAC and NMAC use these hashing algorithms along with a key to produce a hash value. MD5 produces a hash value of 128bits whereas SHA-1 produces a hash value of 160bits. Since the greater the size of the hash value, greater the security, SHA-1 is more secure than MD5. But on the other hand, MD5 is easier to implement as compared to SHA-1.

In all the algorithms mentioned above the workload is a text of a certain length. But when our message is an image file, we use geometric hashing to authenticate he image. Here 2D coordinates are passed through the hash function. These undergo a 2D transformation known as normalization. The two coordinates to be transformed are chosen randomly. Here the output image can be of a certain number of bytes to a few kilobytes as compared to a text of l-bits.

4. CONCLUSION

In this review paper, we have reviewed the various cryptographic hashing algorithms and their application in message authentication. The benefits and drawbacks of each of these algorithms were evaluated as well. Also, each of their performance and level of security was compared. The use of keys along with hashing algorithms was also observed and its advantages were highlighted.

There is still scope for improvement as none of these algorithms are perfect. They all have their drawbacks and in the future, work can be conducted to overcome these shortcomings.

5. REFERENCES

[1] Mihir Bellare , Ran Canetti and Hugo Krawczyk "Keying hash functions for message authentication (1996)" in Advances in Cryptology – Crypto 96 Proceedings, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed., Springer-Verlag, 1996. Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[2] H.Z. Hel-Or and Y. Yitzhaki "Geometric Hashing Techniques for Watermarking". Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[3] Burt Kaliski and Matt Robshaw "Message Authentication with MD5" in CryptoBytes volume 1, Number 1, spring 1995.

[4] D. Eastlake and P. Jones "US Secure Hash Algorithm 1 (SHA1)" September 2001. Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.

[5] H. Krawczyk, M. Bellare and R. Canetti "HMAC: Keyed-Hashing for Message Authentication" February 1997. Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullender