Comparison Between Various Message Authentication Code (MAC) Generation Methods

IF3058 – Kriptografi

Halida Astatin (13507049)[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*if17049@students.itb.ac.id*

***Abstract*—MAC (Message Authentication Code) Algorithm is an algorithm which accepts a private key and a message to be authenticated, and later extracts a MAC tag. The value of a MAC protects the data integrity and also the content of the message. Despites its similarity with hash functions, both algorithm has different security needs. To be considered secure, a MAC function has to be able to survive chosen-plaintext attack. This means that although an attacker knows the MAC value of a message, it has to take him a lot amount of effort to crack other messages. MAC is also different from a digital signature, because MAC is generated and verified using the same private key.**
There are several different methods that can be applied to generate MAC value, for example by using the block-cipher algorithm (OMAC, CBC-MAC, and PMAC), or using hash value (HMAC). In this paper we shall discuss the advantages and disadvantages of each method, also the possibility of creating new MAC generation method.

***Index Terms*—Algorithm, generation, MAC, methods.**

I. Introduction

The advancement of information technology has changed many aspects of our daily life. The changes made by information technology's vast advancements includes the usage of network as a means of data sharing. However, this action has brought up the issue of data integrity and security. A network is something highly accessible to unlimited amount of people around the world, and therefore in the case of data trading across network, the security of said data is relatively low.

Up until now, cryptography, especially the modern algorithms which work in bit-mode, has been considered a reliable method to secure data. Mainly the focus of cryptography is about data secrecy, but there are also other cases in which cryptography protects data integrity. One of the threats faced by data shared over the network is forgery or fraud. To ensure the integrity and authentication of data, one of the method that can be applied is Message Authentication Code (MAC).

In many occasions, people does not care about the secrecy of, for example, an email they sent using an email provider across the internet. People does not worry much about the secrecy of the email they sent, but they definitely want to be sure that the email they received was indeed the one being sent. It may seem that encryption also provides the aspect of protecting data integrity, but that is not always the case. Once an attacker finds out the algorithm used to encrypt the data, he can afterwards use the algorithm to create fake data and intercepts the real message being sent.

There are many different methods to generate message authentication code, a couple examples are using block-cipher algorithm or hash functions. In this paper we shall discuss the advantages and disadvantages of each method which can be used to generate a MAC value. Furthermore this paper will discuss the possibility of a modified MAC generation method.

II. Message Authentication Code

Message Authentication Code (MAC) is a short piece of information in the form of a code that can be used to authenticate another piece of information in form of a message. MAC is a one-way function which uses a secret key in generating its hash value. The hash value generated by using MAC is always of a fixed size for any size of message. Once generated, a MAC value is embedded into the message it corresponds to. Afterwards, MAC is used to authenticate the message without having to encrypt the message itself. However, MAC is different from a digital signature because it merely provides the function of authentication and message integrity.

A MAC Algorithm is an algorithm that accepts a private key and the message to be authenticated, and afterwards extracts a MAC tag. The MAC value protects not only the integrity of data, but also detects any changes done to the contents of the message. In mathematical notation, MAC goes:

$$MAC = C_K (M)$$

Where MAC is the hash value generated by the function, C is the hash function or MAC algorithm, and K equals secret key used in function.

Despite its similarity with hash functions, MAC and hash functions each possesses different security needs. They both corresponds to different security aspects to be handled. To be considered secure, a MAC algorithm has to survive forgery done using chosen-plaintext attack. In other words, although an attacker knows the MAC value of a message, it should take him a great amount of effort to find the MAC value of other messages.

As mentioned above, MAC algorithms differ from digital signatures. Although both methods bear the same function in its relation with data integrity and authentication, the two are different because MAC is generated and verified using a certain secret key. Therefore, before beginning communication, both sender and receiver of the message in question need to agree upon a certain key. Consequently, MAC does not serve the cryptographic function of non-repudiation. Any user who can verify a MAC can also generate MAC values for other messages. Contrary to that, a digital signature uses the private key of a key pair, or using asymmetric encryption method. This private key is accessible to its holder only, hence a digital signature proves that a document was signed by no other than the holder of that private key. This means that digital signatures provides the user with non-repudiation.

MAC can be applied in authentication of an archive used by two or more users. In exchanging data, forgery is highly likely, hence the usage of MAC. By the means of MAC, the two or more party sharing the same archive can always be convinced of the authenticity of the other parties. Other than that, MAC can also be applied in protecting the integrity and authenticity of the archive's content, for example due to a virus attack.

This is how we check an archive's integrity: compute the MAC value of said archive, then put the MAC value in a table. If the protection of the data is done by using a certain hash function (for example MD5 or SHA-1), a virus could then compute the hash value of the archive it attacks and swap the value in the table. This is not applicable, however, when a data is protected using MAC, because the virus would not know the key used to generate the message authentication code.

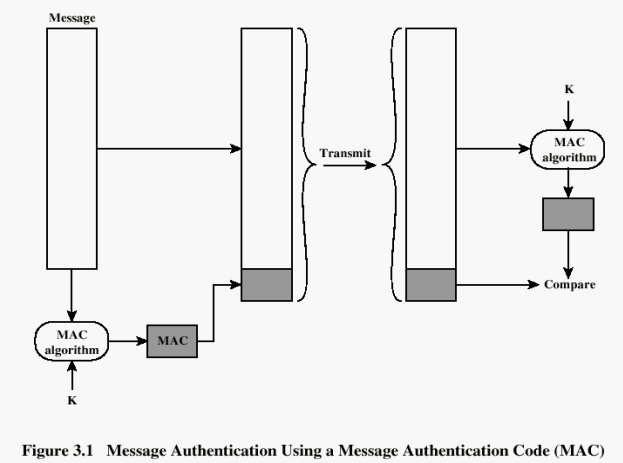


Figure 3.1 Message Authentication Using a Message Authentication Code (MAC)

Figure 1Message Authentication Using MAC

MAC is used for message authentication without having to encrypt the message itself. Using a secret key K (sender and receiver are assumed to have agreed upon a certain secret key beforehand), the sender would compute the MAC value of a message he is trying to send. The MAC value that has been computed would then be embedded into the message, and afterwards the message which has already included the MAC value will be sent to the receiver. Using the exact same key, the receiver would then compute the message's MAC value and compare it to the MAC value he received in said message. Should the two MAC values corresponds to each other, it can be concluded that the message was sent by a legit person and that the content of the message had not undergo any changes during transmission process. If the message was not originated from the real sender, the MAC value of the sender would not be the same as the MAC value generated in the receiver's end, because supposedly there are no other parties other than the sender and receiver who knows exactly what the secret key is. Same goes for the case in which the content of the message is changed during transmission, the MAC value would not be corresponding with each other.

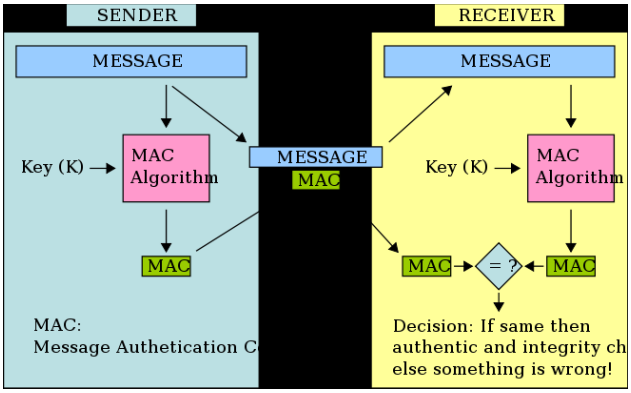


Figure 2 An example of MAC implementation in message exchange

The figure above describes the use of MAC in message exchanging. In the example, the message sender runs the message through a MAC algorithm to produce MAC tag. Afterwards, the message and the MAC data tag are sent to the receiver. The receiver then would run the same message through the same MAC algorithm using the same secret key, producing his own MAC data tag. The receiver should then compare the MAC data tag he received from the sender with the one he generated himself. If they are identical, the receiver can assume that the integrity of the message has not been compromised, and the message has not been altered during the transmission of the message.

In its application, there are several points in MAC's security aspect to be noticed, including the security of the key, the algorithm used, and forgery. The security of a MAC algorithm is highly dependent to the secrecy of the key used to generate the MAC value. Both the sender and

the receiver needs to protect the key from any threats it may face. Secondly, the MAC algorithm applied in generating a MAC also determines how secure the said MAC is. Lastly, the security level of a MAC can be determined by how it can protect a message against forgery. A MAC is considered to fail when a third party who does not possess the secret key K can figure out chunks of messages and its MAC value. Said attacker can be assumed to have compiled several examples of text and its valid MAC value by executing an act of observation to the dataflow path between the sender and the receiver.

Further information that we can use to ensure the security level of the MAC algorithm implied is by knowing the types of attack that the MAC algorithm may be up against. Several types of these attacks are known-text attack, chosen-text attack, adaptive chosen-text attack, and brute force attack. Known-text attack is a type of attack that can occur to a MAC in which the attacker may be able to determine the pattern of MAC from two or more pair of message and its MAC. Chosen-text attack is a type of attack that is likely to occur to a MAC in which the attacker may be able to determine the pattern of the MAC from a pair of message and its MAC of his own choice.

The next threat a MAC is up against is the adaptive chosen-text attack. It is a type of attack a MAC may face in which the attacker can determine the pattern of MAC from a certain pair of message and its MAC which later leads to the discovery of the secret key used in generating MAC values. Last threat against a MAC is a brute force attack. The brute force attack is one of the most common attack in cryptography. A typical brute force attack is to try every single possible combination in order to obtain a certain value. In the case of MAC, a brute force attacker would possibly try each and every single possibility of a message o correspond to a certain known hash value.

There are several methods that can be applied in generating MAC values. A couple of said method are using a certain block-cipher algorithm (this is applied in OMAC, CBC-MAC, and PMAC) or using a hash function (applied in HMAC).

III. MAC and Its Comparison Against Different Data Security Methods

A. MAC versus Hash Function

It has been briefly discussed before about the differences between MAC and hash functions. In this part the aspects that makes the two distinguishable will be further reviewed.

A hash function is a function that accepts a string input of arbitrary length then transform the string into an output string that has a fixed length, most commonly of a far smaller size than the input string.

The output string is often called a message digest. A message digest is a unique summary of a message which would help the message receiver determine whether or not the message is exactly the same as the message that was meant to be sent to him.

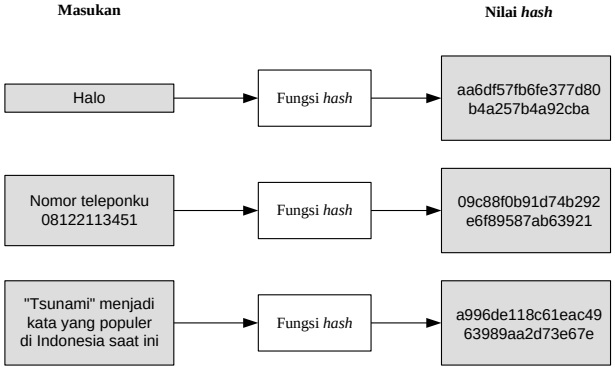


Figure 3Message examples and their message digests

Each and every message digest is unique, but it is not reversible to its original message. A message digest is calculated by the sender and sent with the message, and later the receiver should calculate the message digest of the message and compare it to the message digest value sent to him. If the two message digest values are the same, it can be concluded that the message is legit.

According to that definition, it may be inferred that MAC and hash function are the same. Both methods concludes a message into a certain unique message of a fixed length to guarantee the authenticity and integrity of a message. However, these two needs to be distinguished. Unlike MAC, a hash function only accepts one parameter, which is the document to be hashed, while MAC accepts also a certain secret key. Since the difference of the two methods only lies in the usage of secret key, MAC is also often called keyed-hash function.

The advantage MAC has against hash function is that only certain parties can generate the MAC data tag, because it requires a secret key that only authorized people has access to. Meanwhile, hash message digest can be generated by anyone who knows the function used to hash the message.

B. MAC versus Digital Signature

Several of the difference between MAC and digital signature has already been discussed in the previous part of the paper, but in this part the differences mentioned before should be discussed further.

A digital signature is a method to authenticate a digital message or document. A valid digital signature should give the receiver of a message assurance that the message was created by a valid sender and has not been altered in any ways. Digital signatures are commonly implemented in the distribution of software, financial transactions, or any other cases where forgery and tampering is a serious threat.

Digital signatures employ the concept of asymmetric cryptography. Digital signatures can be considered equivalent to the traditional handwritten signatures in the sense of its functions. Digital signatures that are properly

implemented are even more resistant to forgery than the traditional handwritten signatures. Also, different from the handwritten signature, the value of digital signature differs depending on the document it signs.

A digital signature is the cipher text generated from a document's hash value. The sender signs the document using the sender's private key, and later the receiver of the message would verify the document using the sender's public key. Therefore, unlike in the usage of MAC, the sender and receiver do not need to agree upon a certain key beforehand. The private key used to encrypt the message is accessible to the sender only, therefore a digital signature proves that a document was signed by none other than the holder of the private key. In this case, unlike MAC, digital signatures provide non-repudiation aspect of message protection.

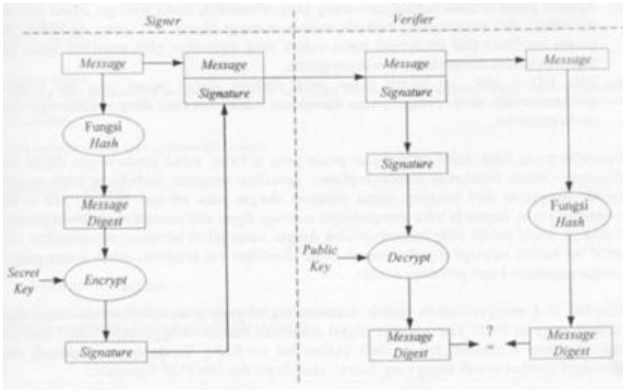


Figure 4 The process of signing and verifying using digital signature

It may seem that digital signatures are more convenient to use, but in this case MAC offers a simplicity that a digital signature does not provide. The creation of a digital signature involves a lot of complex steps, as is explained by the figure above and below.

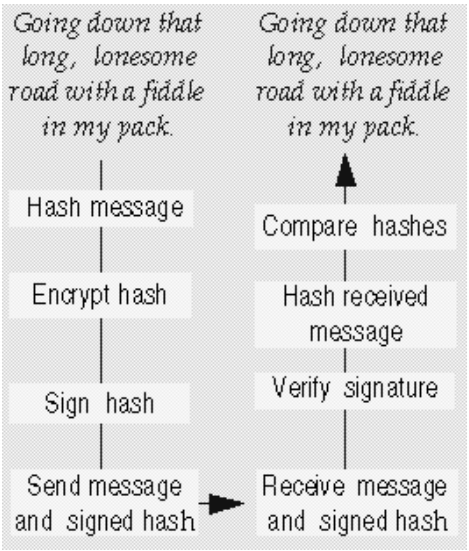


Figure 5 Digital signature signing and verification procedure

IV. MAC Generation Methods

A. Hash-based Message Authentication Code (HMAC)

HMAC (Hash-based Message Authentication Code) is a certain method to compute the value of a message's MAC that involves the use of a cryptographic hash function combined with a secret key. The value will then be used to verify data integrity and authenticity of a message. The hash function used to generate the MAC may vary; one can apply the MD5 algorithm or SHA-1, depending on their own preference. The resulting MAC algorithm is termed HMAC-MD5 or HMAC-SHA-1, depending on the hash function used to create the MAC algorithm. The strength of a MAC algorithm therefore is affected by the strength of the underlying hash function, the length of the resulting message digest and also on the size and quality of the secret key.

The size of the result of a Hash-based MAC depends on the size of the result of the hash function, although it can be truncated if desired.

The size of key used in HMAC has to be of the same size as L/2 or longer, in which L is the size of the resulting hash value. For applications that allow the secret key size of k in which k is larger than the size of the document itself, k has to be hashed before processed.

To implements a hash-based MAC algorithm, first of all for example the people involved in data sharing are Alice and Bob, and they both shared a secret key K. Alice should then connate the message M with K and compute the hash value of the result. The hash value resulted is the MAC value of said message, generated using the secret key K.

B. Block Cipher-based Message Authentication Code (CMAC)

CMAC (Cipher-based Message Authentication Code) is a certain method to compute the value of a message's MAC that is based on the use of a block cipher function. It can be utilized to provide the sender and receiver of the message with authenticity and integrity of a binary data. Unlike CBC-MAC which is only secure for a fixed-length messages, CMAC solves the security limitation of the CBC-MAC.

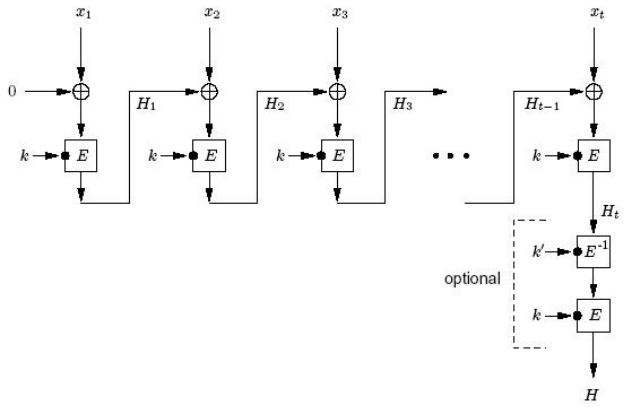


Figure 6 Cipher-based MAC Schema

The main substance of CMAC algorithm is a modification of CBC-MAC that is analyzed under the name XCBC. This algorithm is developed by Black and Rogaway; focused mainly on eliminating the security deficiencies CBC-MAC possessed, but at the time it required three different secret keys. Iwata and Kurosawa then proposed an improvement to XCBC, naming the output algorithm as one-key-CBC-MAC (OMAC). Later they submitted OMAC1, a revision to OMAC with additional security analysis. CMAC here is equivalent to OMAC1.

In block cipher-based MAC algorithm, the MAC value is generated by using a cipher block algorithm with CBC or CFB mode. The hash value, or in this case the MAC value, is the result of last block's encryption. If for example the cipher block used is DES, then the block size would be 64 bits, and the secret key to the MAC algorithm is the DES key sized 56 bits. Data Authentication Algorithm (DAA) is a DES-CBC based MAC algorithm that has been widely used.

C. Stream Cipher-based Message Authentication Code (CMAC)

Stream cipher approach is based on bit operation, which is taking chunks of bits from the document to be encrypted into a MAC. An example is the stream cipher-based MAC algorithm designed by the researcher in RSA laboratory, whose algorithm outputs MAC bits in the size of half its original document.

To create a stream cipher-based algorithm MAC value, the steps to go through are as follows. First of all, the document should be divided into two parts, and each part should undergo an LFSR operation before being processed. The document parts will then be processed to obtain the MAC value.

D. One time pad-based Message Authentication Code

There is also another approach in generating a MAC value, which uses the utilization of an ancient unbreakable cryptography technique, which is one time pad. In this method, MAC value is generated by using one time pad stored in a program. This key is known only to the person who has access to the one time pad stored in the program.

To check the integrity of the message, the receiver has to make use of the program storing the same one time pad as the one used in encrypting the document. This will become a redundancy when the program has to be sent together with the document it encrypts. The one time pad used is disposable—it has to be purged once used so that it will not be used to encrypt any other documents.

V. Analysis: Comparison Among Various MAC Generation Methods

In generating a MAC value of a message, the usage of hash function is considered a lot more complicated than using a cipher algorithm. The reason to this is because hash functions (such as MD5, SHA-1, etcetera) commonly already has a certain default function, and it has to be modified in such a way so that it can also accept and process the secret key K. Meanwhile, cipher algorithm does not need any modification since it also uses a certain secret key in generating a function result. Nevertheless, in using a cipher-based MAC, we need to determine which part of the document should be encrypted and considered as its MAC value.

In cipher block mode, when an encryption process is done only once (using any mode other than CBC), it is highly likely that several documents may have exact same MAC values. That is, if the encrypted block is identical. As a result, cipher block-based MAC requires a complicated algorithm, just like CBC, that forces the original document to be processed over and over again. On the other hand, in the usage of stream cipher-based MAC, the possibility of two different documents having an identical MAC value is relatively low. This is because MAC bits consist of message bits that is distributed everywhere inside the document, which causes the MAC value to be distinctive from others.

As for one time pad-based MAC, this method is exceptionally safe because only the program and the sender knows the key and the program could then generate MAC automatically. However, the effort required to conduct this method is not worthy of the safety, especially with the existence of other much simpler MAC methods.

It is also possible to combine a cipher-based MAC generation algorithm with hash-based MAC generation algorithm. To do so, first of all we process the document in question using a selected hash function. The result of this function will then be processed into a cipher-block encryption (preferably in CBC or CFB mode). In this case, the MAC value is the final result of this encryption process.

VI. Conclusion

There are several conclusions that can be drawn from the discussion in this paper:

a. Despite the similarity the two data security method possess, MAC is different from both digital signature and hash function. It is however said that MAC can be called a keyed-hash function because the difference lies only in the usage of a secret key in MAC.
b. There are various different method to generate a MAC value, some of them are hash-based MAC algorithm, cipher-based MAC algorithm, and one time pad-based MAC algorithm.
c. Hash-based algorithm is relatively faster to compute, but cipher-based algorithm needs less modification. One time pad, on the other hand, despite its high security level, is not preferable due to the complexity it possess.
d. As a new method to generate MAC value, the combination of hash-based algorithm with cipher based algorithm may be considered. To do so, the document to encrypt should be processed using a certain hash function, then the result will be processed using cipher-based algorithm.

VII. Acknowledgment

In the making of this paper, author thanks google.com for always being a reliable search engine; a very useful one also. Wikipedia.org for being a very resourceful website, although many had said that it is not always reliable. Author thanks her parents and sister for the understanding they had given in during the author's campus life and its never-ending stream of tasks. Also for every friends who spent the nights together working on the tasks, especially the certain friend who never mind driving halfway across Bandung to drive me home. It would have been a lot harder to finish this paper without the aid of said people. Thank you very much.

References

Munir, Rinaldi. 2004. *Bahan Kuliah IF3058 Kriptografi*. Departemen Teknik Informatika, Institut Teknologi Bandung.

Rifki, Guntur. 2009. Implementasi Algoritma Enkripsi Rijndael dan Metode Otentikasi MAC(Message Authentication Code) dalam SMS. Institut Teknologi Telkom.

Safrina, Rika. 2007. Cryptography Paper: "Pembangunan Algoritma MAC Berbasis Cipher Aliran". Institut Teknologi Bandung.

http://www.cs.princeton.edu/courses/archive/fall07/cos433/lec8.pdf

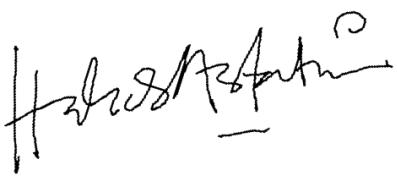
https://tao.truststc.org/Members/yuanxue/network_security/Public%20resources/lecture10_folder/lecture10_notes/download

Statement

I hereby certify that this paper is an original work; not a paraphrase or translation of someone else's paper, neither is this an act of plagiarism.

Bandung, May 17th 2010

signed



Halida Astatin
13507049