International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor (2018): 7.426

Comparison of MAC and Digital Signature

G. Pranitha

Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, India

Abstract: ***A Message Authentication Code is what we get from symmetric cryptography A MAC is used to prevent Intruder from creating a new message and inserting it in to original message. Signatures are generally used for verifying all the financial documents such as stamp papers for registration, payment receipts and for personal identification like Identity cards, marks reports. As all of these will have transactions of money and identity verification, it should be genuine in nature. Digital signature helps to identify the identity as well as to protect the integrity of message. In this paper, we compare the importance of digital signature and message authentication code for certification.***

Keywords: Message authentication code, digital signature, key, Hash

1. Introduction

Security has become very important criteria these days. One way to provide security is to protect the data from Intruder. Nowadays it became a necessity to authenticate the sender and to prevent the data to be modified while sending the data. Information security will be used in cryptography on several levels. The information cannot be known to anyone without a key which can decrypt it. The information maintains its integrity while it is transmitted and being stored. Cryptography helps in providing nonrepudiation. This means that the sender and the delivery of a message can be easily verified.

2. Message authentication Code

2.1 Introduction

A MAC, Message Authentication Code, preserves data integrity, i.e., it ensures that creation and any changes of the message have been made by authorised entities. Only the authorised entities can check a MAC, and all who can check can also change the data .In most legally interesting cases, you want to be able to verify that one single individual wrote something .Also, in many situations it is good if everyone is able to check the signature.

MAC is used as a proof in symmetric key cryptography, which then is added to the end of the crypto message. At the receiver point, the receiver decrypts the message, generate a MAC from it and compare with the received MAC one. The integrity of the message can be guaranteed practically in the case of the same received and computed MACs.

2.2 Discussion

MAC Description:
Based on the name, the MAC is used for authentication. Moreover, it is used to check the integrity and to become sure regarding non-repudiation of the message.

MAC has a lower length in comparison with the plaintext. Thus, it is not unique like hash function. In other words, two different plaintexts may have the same MAC values. However, the likelihood of this occurrence is very low and thus it can be used for authentication and integrity.

In comparison with checksum, a private key is used to generate MAC. So it cannot be regenerated by an imaginary intruder, who has an oracle who decrypts the crypt massage.

2.3 Types of MAC

A. One-time MAC
Similar to one-time encryption, one can define a one-time MAC algorithm, which provides security against potential attacks and it is generally faster than other message authentication algorithms based on PFR functions.

Definition One-time MAC is a pair of algorithms (S, V):
$S(m, k_1, k_2) := P(m, k_1) + k_2 \pmod q$: returns an authentication code t
V(m, k_1, k_2, t): returns a value true or false depending on the correctness of the examined authentication code t
where:
q is a large prime (about 2^{128}),
m is a message that contains L blocks of size of 128 bits,
$\mathbf{k_1}$, $\mathbf{k_2}$ are two secret keys; each of them has value from the interval [1, q],
$\mathbf{P(m, x) = m[L]\cdot x^L + ... + m[1]\cdot x}$ is a polynomial of degree L

It can be proved that two messages secured by using the same keys are indistinguishable for potential observers.

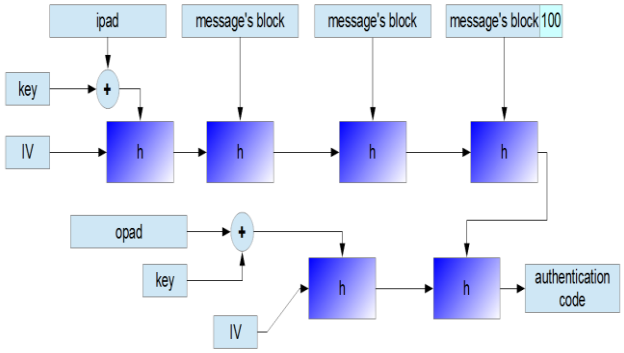
B. Carter-Wegman MAC
The construction of Carter-Wegman MAC is based on the idea of one-time MAC. It is extended by a pseudorandom function to allow using one secret key many times for subsequent messages.
Definition Having a secure one-time MAC (S, V) defined over sets (M, K_J, T) and a secure pseudorandom function $F: K_F \times \{0,1\}^n \rightarrow \{0,1\}^n$, one can define a pair of algorithms Carter-Wegman MAC:
$S_{C\text{-}W}(m, k_F, k_J) := (r, F(k_F, r) \text{ XOR } S(m, k_J))$: returns an authentication code that is a pair $(r, t_{C\text{-}W})$
$V_{C\text{-}W}(m, k_J, F(k_F, r) \text{ XOR } t_{C\text{-}W})$:
returns true or false depending on the correctness of the examined authentication code $(r, t_{C\text{-}W})$

where:

k_J, k_F are two secret keys of sets **K_J** and **K_F**,
T is a set of all possible values of authentication codes of the one-time MAC algorithm (of length of n bits),
r is a random number of length of n bits,
(r, t_{C-W}) is a value of an authentication code of the Carter-Wegman algorithm (of length of 2n bits)

C. HMAC

HMAC is a popular system of checking message integrity, which is quite similar to NMAC. The HMAC algorithm uses one-way hash functions to produce unique mac values.



The input parameters ipad and opad are used to modify the secret key. They may have various values assigned. It is recommended to choose the values that would make both inputs to the hash functions look as dissimilar as possible (that is, that modify the secret key in two different ways).

Using a secure hash function (that means the function which doesn't produce the same outputs for different input data) guarantees the security of the HMAC algorithm.

Nowadays, the HMAC algorithm is used in many systems, including some popular Internet protocols (SSL, IPsec, SSH)

3. Digital Signature

In asymmetric ciphers, one single individual holds the private key, while everyone can get the public key .So if you encrypt with the private key, and send both cryptogram and message, anyone can check that “decryption” with the public key does indeed create the message .We know that some public key systems do not allow “encryption” with the private key .Most systems can be modified to generate and verify signatures.

A digital signature should not only be tied to the signing user, but also to the message • The example of encrypting with the private key does this: only Alice can create it, and it is valid only if the decryption and the plaintext coincide • No attempt is made to hide the information (unless it is encrypted using another method).

3.1 RSA signatures

Alice sets up RSA as usual .In order to sign a message m, Alice uses her private key d (and not Bob’s public key) to create the signature s = m d mod n .

Alice now gives both m and s to Bob .He uses Alice’s public key to verify the signature by comparing m and s e mod n

3.2 ElGamal signatures

Choose a large prime p, and a primitive root α mod p. Also, take a random integer a and calculate β = α a mod p • The public key is the values of p, α, and β, while the secret key is the value a • Signing uses a random integer k with gcd(k, p − 1) = 1, and the signature is the pair (r,s) where (r = α k mod p s = k −1 (m − ar) mod (p − 1) (encryption: (α k , β km)) • Verification is done comparing β r r s and α m mod p, since β r r s = α arα k(m−ar)/k = α m mod p

3.3 DSA signatures

Choose a 160-bit prime q, a large prime p so that q is a factor in p − 1, and a primitive root g mod p. Set α = g (p−1)/q mod p so that α q = 1 mod p, take a random integer a < q − 1 and calculate β = α a mod p • The public key is the values of p, q, α, and β, while the secret key is the value a • Signing uses a random integer k < q − 1, and the signature is the pair (r,s) where (r = (α k mod p) mod q s = k −1 (m − ar) mod q • To verify, compute t = s −1m mod q and u = s −1 r mod q. Verification is done comparing (α tβ u mod p) and r mod q, since (α tβ u mod p) = (α km/(m+ar)α ar/(m+ar) mod p) = (α k mod p) = r mod q

There are advantages when using α = g (p−1)/q mod p (so that α q = 1 mod p) instead of the primitive root directly • One advantage is that the signatures can be generated mod q instead of mod p, which reduces calculation and increases speed • Security is still based on the difficulty of discrete log mod p (and discrete log algorithms have problems with large factors in p − 1) • Finally, the verification uses two exponentiations rather than ElGamal’s three, this speeds things up the execution.

4. Comparison of Digital Signature and MAC

When a file is downloaded from a website, the file can be checked against the hash (for example an MD5 or SHA-1 hash) that is often listed next to the file download link. If the original hash and the downloaded file's hash are the same, you can be fairly sure the data hasn't been tampered with. If they are not the same, the downloaded file should be discarded as it has probably been compromised. Cryptographic hash functions can be used in conjunction with digital signatures and MACs as well.

1) **Digital Signatures:** These are mathematical processes for proving the authenticity of a document or message. A proper digital signature implies that the message or document was created by the actual sender and not someone else. They are commonly used to prove that an electronic signature (a person's actual signature performed on the computer) is authentic. They use asymmetric cryptography.
2) **MAC (Message authentication code):** This is a small piece of information used to authenticate a message. The MAC algorithm takes a message and secret key and outputs a MAC value or "tag". MACs only use secret

keys, and rely on symmetric encryption. However, to function as intended the MAC must be able to resist plaintext attacks even if a hacker knows the secret key. Although the hacker can create their own MACs from the key, the MAC algorithm must be strong enough to make it impossible for the hacker to calculate the MAC for other messages. MACs can be built from hash functions; these are known as keyed hash functions.

Let's discuss some of the differences between the three:

A MAC can be a cryptographic hash function (as in keyed hash functions), but a cryptographic hash function is not always a MAC.

Digital signatures can be used in conjunction with cryptographic hash functions (as in SHA-1 and the Digital Signature Algorithm), but a cryptographic hash function is not always a digital signature.

Digital signatures utilize asymmetric cryptography, whereas MACs use symmetric cryptography.

Digital signatures provide for non-repudiation, whereas MACs do not, and cryptographic hash functions usually do not

MACs require additional security requirements than cryptographic hash functions, for example, the MAC must be resistant to plaintext attacks also.

A good comparison between cryptography hash functions and MAC:

Cryptographic security goal	HASH	MAC	Digital Signature
Integrity	Yes	Yes	Yes
Authentication	No	Yes	Yes
Non-Repudiation	No	No	Yes
Kind of keys	None	Symmetric keys	Asymmetric keys

Comparison of hash, MAC and digital signature.

A **message authentication code** (MAC) (sometimes also known as **keyed hash**) protects against message forgery by anyone who doesn't know the secret key (shared by sender and receiver).

This means that the receiver can forge any message – thus we have both integrity and authentication(as long as the receiver doesn't have a split personality), but not non-repudiation.

An attacker could replay earlier messages authenticated with the same key, so a protocol should take measures against this (e.g. by including message numbers or timestamps). (Also, in case of a two-sided conversation, make sure that either both sides have different keys, or by another way make sure that messages from one side can't sent back by an attacker to this side.)

MACs can be created from unkeyed hashes (e.g. with the HMAC construction), or created directly as MAC algorithms.

A (digital) **signature** is created with a private key, and verified with the corresponding public key of an asymmetric key-pair. Only the holder of the private key can create this signature, and normally anyone knowing the public key can verify it. Digital signatures don't prevent the replay attack mentioned previously.

5. Conclusion

A digital signature and MAC are used for authentication only. But based on the requirement of the application we will be using either of them. In order to increase the authentication we can use both of them also.

References

[1] Christian Badertscher,Ueli Maurer," On composible security for digital signature', Springer
[2] D.Johnson, "The elliptic curve digital signature algorithm"A.Certicom Whitepaper
[3] H Rezaeighaleh, R Laurens," Secure Smart Card Signing with Time-based Digital Signature, Conference on Computing 2018 ieeexplore.ieee.org
[4] WJ Long, W Lin ,"An authentication protocol for wearable medical devices-- ieeexplore.ieee.org

Author Profile

G. Pranitha is working as Assistant Professor in Anil Neerukonda Institute of Technology for computer science department. Area of interest is information security and cryptography.