Message Authentication Codes

Sachin Tripathi

IIT(ISM), Dhanbad

Outline

- Principle behind MACs
- The security properties that can be achieved with MACs
- How MACs can be realized with hash functions and with block ciphers

Introduction

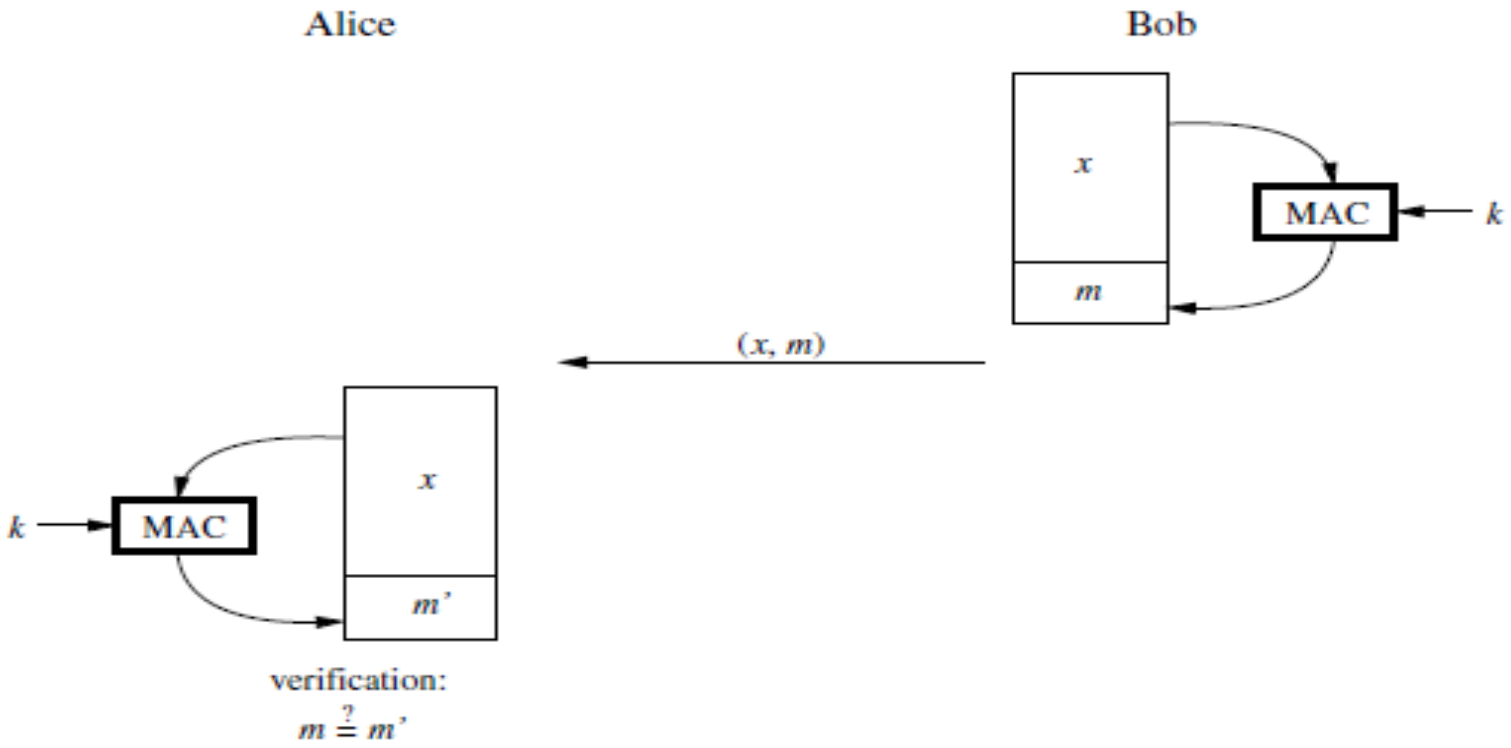
- A Message Authentication Code (MAC), also known as a cryptographic checksum or a keyed hash function, is widely used in practice.
- In terms of security functionality, MACs share some properties with digital signatures, since they also provide message integrity and message authentication.
- Unlike digital signatures, MACs are symmetric-key schemes and they do not provide nonrepudiation.
- One advantage of MACs is that they are much faster than digital signatures since they are based on either block ciphers or hash functions.

Principles of MAC

- Similar to digital signatures, MACs append an authentication tag to a message.
- The crucial difference between MACs and digital signatures is that MACs use a symmetric key k for both generating the authentication tag and verifying it.
- A MAC is a function of the symmetric key k and the message x.

$$m = mac_k(x)$$

Working Procedure



Principle of message authentication codes (MACs)

Properties of MAC

Cryptographic checksum: A MAC generates a cryptographically secure authentication tag for a given message.

Symmetric: MACs are based on secret symmetric keys. The signing and verifying parties must share a secret key.

Arbitrary message size: MACs accept messages of arbitrary length.

Fixed output length: MACs generate fixed-size authentication tags.

Message integrity: MACs provide message integrity.

Message authentication: The receiving party is assured of the origin of the message.
No nonrepudiation: Since MACs are based on symmetric principles, they do not provide nonrepudiation.

Formation of MAC

In practice, message authentication codes are constructed in essentially two different ways from **block ciphers** or from **hash functions**. We will introduce both options for realizing MACs.

MACs from Hash Functions: HMAC

- HMAC is used in both the Transport Layer Security (TLS) protocol and in the IPSec protocol
- The basic idea behind all hash-based message authentication codes is that the key is hashed together with the message
- Two obvious constructions are possible

  Secret prefix MAC: $m = MAC_k(x) = h(k||x)$

  Secret suffix MAC: $m = MAC_k(x) = h(x||k)$

Attack Against Secret Prefix MACs

Alice	Oscar	Bob
		$x = (x_1, \ldots, x_n)$ $m = h(k\|\|x_1, \ldots, x_n)$
	$\lightning$ intercept $\xleftarrow{(x,m)}$	
	$x_O = (x_1, \ldots, x_n, x_{n+1})$ $m_O = h(m\|\|x_{n+1})$	
	$\xleftarrow{(x_O, m_O)}$	
$m' = h(k\|\|x_1, \ldots, x_n, x_{n+1})$		

since $m' = m_O$

$\Rightarrow$ valid signature!

Attacks Against Secret Suffix MACs

Assume Oscar is capable of constructing a collision in the hash function, i.e., he can find x and x_O such that:

$$h(x) = h(x_O).$$

The two messages x and x_O can be, for instance, two versions of a contract which are different in some crucial aspect, e.g., the agreed upon payment. If Bob signs x with a message authentication code

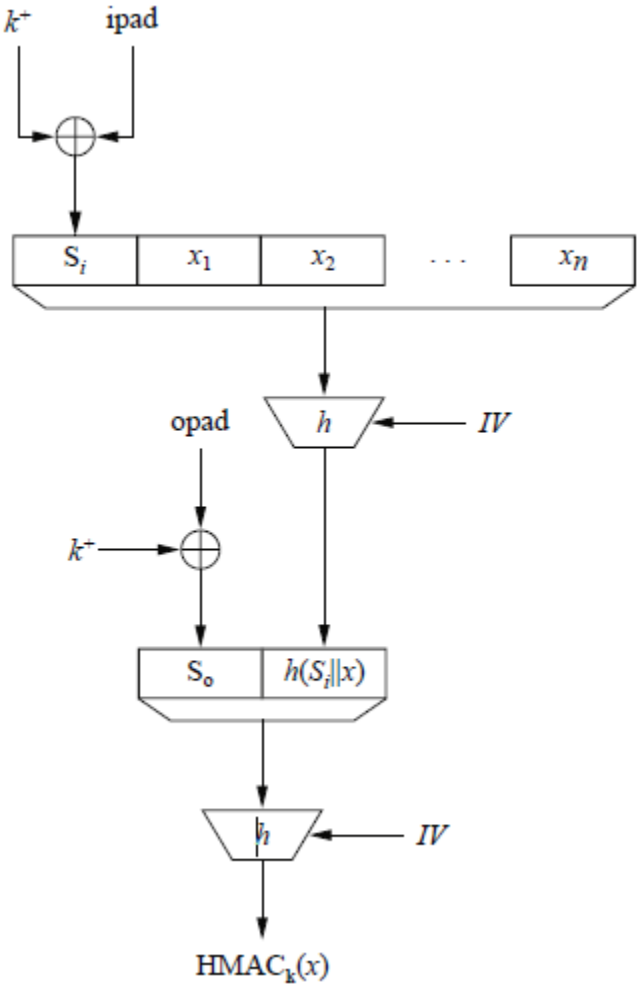
$$m = h(x||k)$$

m is also a valid checksum for x_O, i.e.,

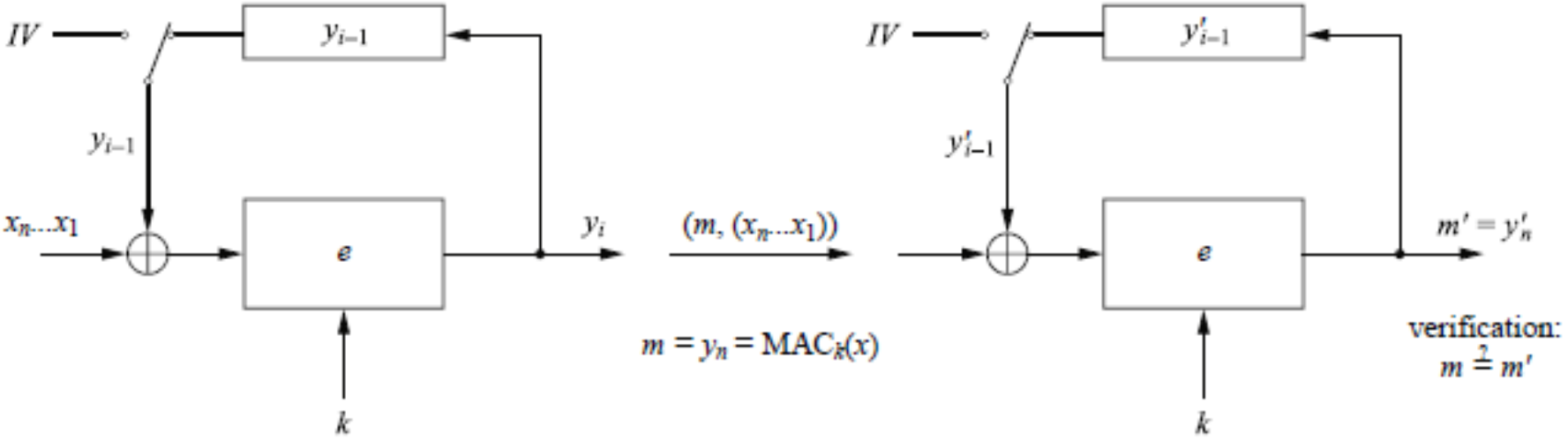
$$m = h(x||k) = h(x_O||k)$$

The reason for this is again given by the iterative nature of the MAC computation.

HMAC



MACs from Block Ciphers: CBC-MAC



Conclusion

- MACs provide two security services, *message integrity and message authentication,* using symmetric techniques.
- Both of these services are also provided by digital signatures, but MACs are much faster.
- MACs do not provide nonrepudiation.
- In practice, MACs are either based on block ciphers or on hash functions.

HMAC is a popular MAC used in many practical protocols such as TLS.

Thank You