

## Chapter 12

### Message Authentication Codes (MACs)

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. However, 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.

In this chapter you will learn:

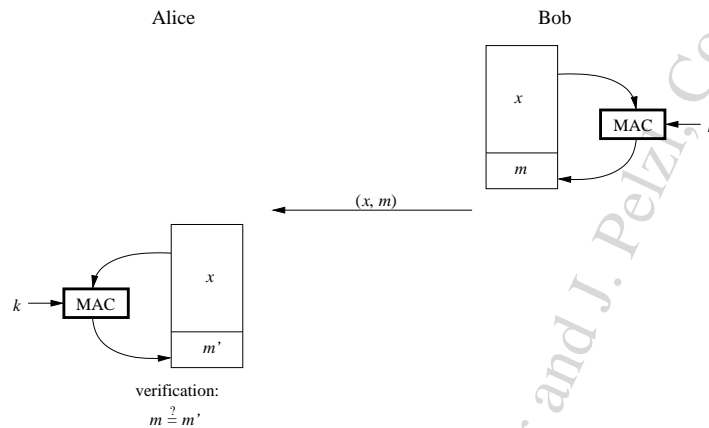
- The principle behind MACs
- The security properties that can be achieved with MACs
- How MACs can be realized with hash functions and with block ciphers

## 12.1 Principles of Message Authentication Codes

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  $x$ . We will use the notation

$$m = \text{MAC}_k(x)$$

for this in the following. The principle of the MAC calculation and verification is shown in Figure 12.1.



**Fig. 12.1** Principle of message authentication codes (MACs)

The motivation for using MACs is typically that Alice and Bob want to be assured that any manipulations of a message  $x$  in transit are detected. For this, Bob computes the MAC as a function of the message and the shared secret key  $k$ . He sends both the message and the authentication tag  $m$  to Alice. Upon receiving the message and  $m$ , Alice verifies both. Since this is a symmetric set-up, she simply repeats the steps that Bob conducted when sending the message: She merely recomputes the authentication tag with the received message and the symmetric key.

The underlying assumption of this system is that the MAC computation will yield an incorrect result if the message  $x$  was altered in transit. Hence, *message integrity* is provided as a security service. Furthermore, Alice is now assured that Bob was the originator of the message since only the two parties with the same secret key  $k$  have the possibility to compute the MAC. If an adversary, Oscar, changes the message during transmission, he cannot simply compute a valid MAC since he lacks the secret key. Any malicious or accidental (e.g., due to transmission errors) forgery of the message will be detected by the receiver due to a failed verification of the MAC.

That means, from Alice's perspective, Bob must have generated the MAC. In terms of security services, message authentication is provided.

In practice, a messages  $x$  is often much larger than the corresponding MAC. Hence, similar to hash functions, the output of a MAC computation is a fixed-length authentication tag which is independent of the length of the input.

Together with earlier discussed characteristics of MACs, we can summarize all their important properties:

#### Properties of Message Authentication Codes

1. **Cryptographic checksum** A MAC generates a cryptographically secure authentication tag for a given message.
2. **Symmetric** MACs are based on secret symmetric keys. The signing and verifying parties must share a secret key.
3. **Arbitrary message size** MACs accept messages of arbitrary length.
4. **Fixed output length** MACs generate fixed-size authentication tags.
5. **Message integrity** MACs provide message integrity: Any manipulations of a message during transit will be detected by the receiver.
6. **Message authentication** The receiving party is assured of the origin of the message.
7. **No nonrepudiation** Since MACs are based on symmetric principles, they do not provide nonrepudiation.

The last point is important to keep in mind: MACs do not provide nonrepudiation. Since the two communicating parties share the same key, there is no possibility to prove towards a neutral third party, e.g., a judge, whether a message and its MAC originated from Alice or Bob. Thus, MACs offer no protection in scenarios where either Alice or Bob is dishonest, like the car-buying example we described in Section 10.1.1. A symmetric secret key is not tied to a certain person but rather to two parties, and hence a judge cannot distinguish between Alice and Bob in case of a dispute.

In practice, message authentication codes are constructed in essentially two different ways from block ciphers or from hash functions. In the subsequent sections of this chapter we will introduce both options for realizing MACs.

## 12.2 MACs from Hash Functions: HMAC

An option for realizing MACs is to use cryptographic hash functions such as SHA-1 as a building block. One possible construction, named HMAC, has become very popular in practice over the last decade. For instance, it is used in both the Transport Layer Security (TLS) protocol (indicated by the little lock symbol in your Web browser) as well as in the IPsec protocol suite. One reason for the widespread use of