

# Lecture 14: Message Authentication Code

Yuan Xue

In this lecture, we will study message authentication. This lecture is organized as follows.

- First we will review the cryptographic algorithms that we have learned so far, and discuss their limitations in protecting data integrity and supporting message authentication.
- Then we will examine the concept and the design of Message Authentication Code (MAC).

## I. REVIEW

In the previous lectures, we have learned symmetric encryption algorithms and asymmetric encryption algorithms. Now let's review what security goals these algorithms can achieve. Here we are interested in the following security properties: confidentiality, message authentication, and non-repudiation. In particular, *message authentication* involves two aspects:

- *Source authentication*, which verifies the identity of the source, prevents the acceptance of messages from a fraudulent source.
- *Data integrity*, which protects the data from modification.

Let's start with symmetric encryption. As shown in Fig. 1 (a), A sends B a message  $M$  encrypted by their shared secret key  $K$ . Because a third party is unable to recover the plaintext of the message without the knowledge of  $K$ , confidentiality is provided. Now let's examine how encryption mechanism can provide message authentication. Generally, B is assured that the message is from A, because A is the only person (other than B) who is able to generate the ciphertext that can be decrypted using  $K$ . Further, if  $M$  is fully recovered, B knows none of the bits of  $M$  have been altered.

However, to achieve this goal B needs to be able to identify the "correct plaintext" from the ones that is decrypted from an altered ciphertext, or the ciphertext generated with a different key. And there are several scenarios:

- If  $M$  is in ordinary English, then B can recognize the message by reading off it. But this "plaintext" is difficult to be recognized automatically.
- If  $M$  is in binary code, and can be any arbitrary bit pattern, then there is no way to determine automatically, whether the recovered message is legitimate or not.

Lacking of an automatic way to verify the recovered message limits the usage of symmetric encryption as a mechanism for message authentication. Moreover, if a block cipher (such as DES, AES) is used,

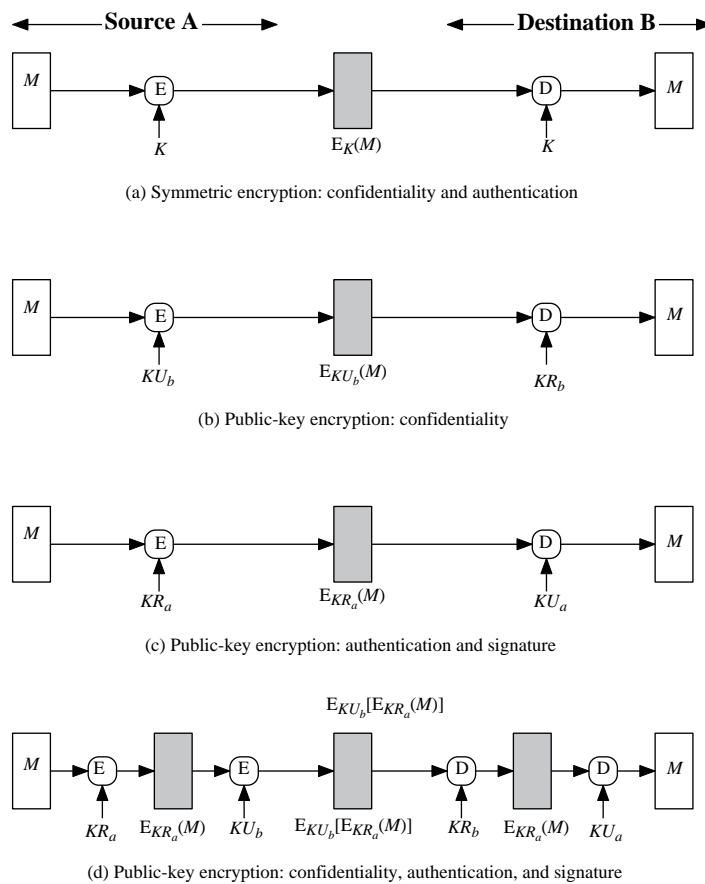


Fig. 1. Basic Usages of Message Encryption.

then modes of operations need to be applied for messages whose length is larger than a block. As we have discussed in the previous lectures, no data integrity protection is provided by any mode of operation. This means that an attacker is able to alter the message (such as re-arranging the blocks) without being detected by the receiver.

Now let's proceed with public key encryption. As shown in Fig. 1 (b), if A sends a message  $M$  to B, which is encrypted with B's public key  $K_{U_b}$ , then message confidentiality can be preserved. This is because the ciphertext of  $M$  can only be decrypted by B's private key. However no authentication is provided, as any one may have access to B's public key and generate such a ciphertext of  $M$ . If  $M$  is encrypted by A's private key as shown in Fig. 1 (c), the encryption provides both authentication and source non-repudiation in limited scenarios. This is because only A could have prepared the ciphertext of  $M$ . Since any other person with the knowledge of A's public key is able to decrypt the message, confidentiality is not provided. Due to the same reasoning as symmetric encryption, the authentication is only provided in limited scenarios such as (1) receiver is able to distinguish the well-formed plaintext and random bits, and (2) modes of operation are not used in encryption procedure. A double use of the public

key and private key encryption is needed to achieve confidentiality, authentication, and non-repudiation simultaneously.

From the above discussions, we also observe that to provide message authentication (even in limited cases), the whole message needs to be encrypted, which may involve high computational overhead.

At this point, we understand that the authentication support from the encryption mechanism (including symmetric and asymmetric encryption) is (1) limited; (2) inefficient. So additional message authentication mechanism needs to be developed. In what follows, we will study message authentication code, which is one of such mechanisms.

## II. MESSAGE AUTHENTICATION CODE

### A. Intuition

One of the reasons that encryption mechanism does not provide a good solution for message authentication is that it is difficult for the receiver to identify the legitimate plaintext.

To address this problem, we can apply an error detection code to the message so that only legitimate plaintext can pass the error detection. Such error detection codes are used in the network communication to provide data integrity verification against bit errors introduced by communication channel noise. But it can not provide data integrity protection against malicious attackers. The reason is that the attackers can manipulate the message in a way which can not be detected by error detection code. Although encrypting the message and its error detection code as a whole seems to be a valid approach, yet existing work shows that it still suffers from some attacks<sup>1</sup>. Also it can not solve the efficiency issue of the encryption mechanism.

In light of error detection code, we can design a code that uses a secret key. Without the key, modifying the message in a way that it matches the code is impossible. This idea leads to the design of message authentication code (MAC).

### B. Concept and Model

Essentially, the message authentication code ( $MAC$ ) is a small fixed-size block of data that is generated based on a message  $M$  of variable length using secret key  $K$  as follows. It is also called cryptographic checksum.

$$MAC = C(K, M) \quad (1)$$

<sup>1</sup>Jueneman, R. R., Matyas, S. M., and Meyer, C. H., "Message Authentication", IEEE Communication, Vol 23, No. 9, 1985, pp 29-40.

If A wishes to send B a message  $M$ , and protects it via a MAC, they first need to share a secret key  $K$ . Then A calculates code  $MAC$  as a function of  $M$  and  $K$ . Then the message  $M$  plus the code  $MAC$  are transmitted to B. B performs the same calculation on  $M$ , using  $K$  to generate a new code  $MAC'$ . The received code  $MAC$  is compared to the calculated code  $MAC'$  to verify the data integrity. As only A is able to generate  $MAC$ , source authentication is also achieved. This procedure is shown in Fig. 2.

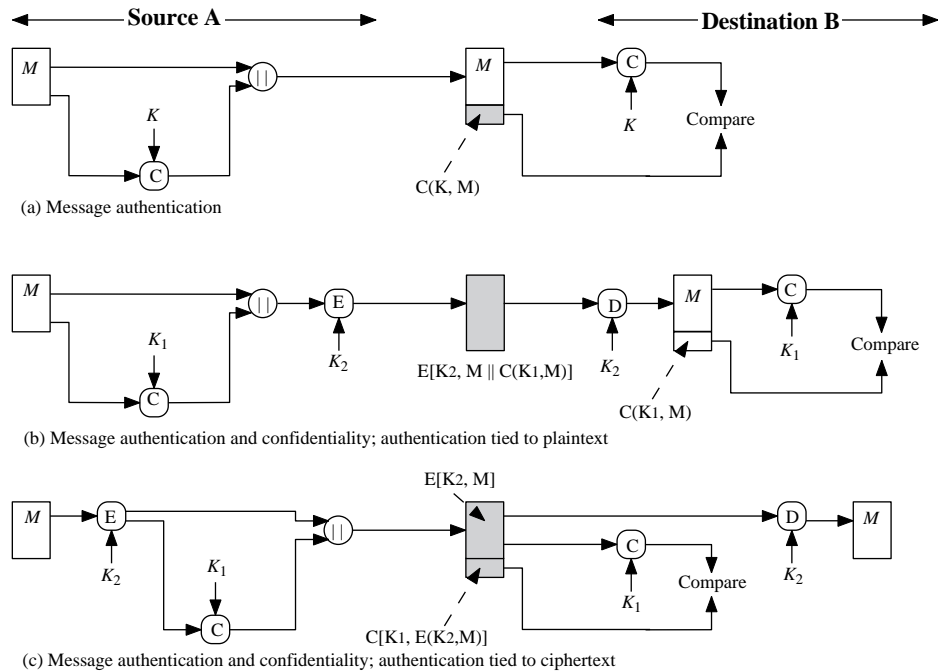


Fig. 2. Basic Usages of Message Encryption.

Fig. 2 (b) and (c) illustrate how to provide confidentiality and authentication simultaneously using MAC. Note that two different keys need to be used.

MAC provides an efficient way to message authentication. It also separates the authentication function from confidentiality. This is an attractive feature for many applications (such as software package distribution) where confidentiality is not necessary. Note that MAC can not be used as a digital signature to provide non-repudiation.

### C. Design of MAC

One of the most widely used MACs is referred to as the Data Authentication Algorithm. The algorithm is designed using the cipher block chaining (CBC) mode of operation of DES, as shown in Fig. 3. The MAC code (also referred to as DAC) consists of either the entire final block  $C_n$  or the leftmost  $m$  bits of the block with  $16 \leq m \leq 64$ .

We will introduce another approach to design MAC using Hash functions (HMAC) in the next lecture.

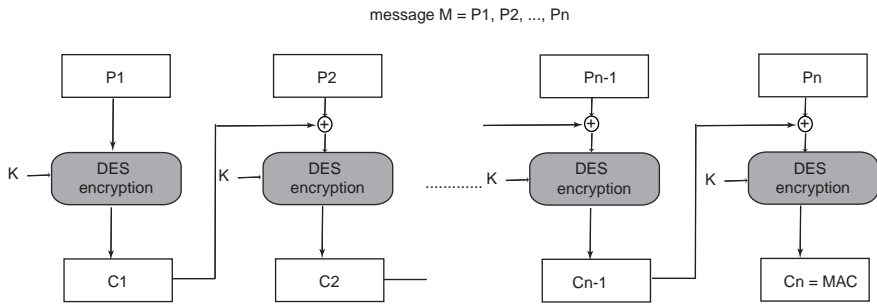


Fig. 3. MAC design using DES in CBC mode of operation.