Message authentication:

Message authentication is a mechanism or service used to verify the integrity of a message. Message authentication assures that data received are exactly as sent by (i.e., contain no modification, insertion, deletion, or replay) and that the purported identity of the sender is valid.

Symmetric encryption provides authentication among those who share the secret key. Encryption of a message by a sender's private key also provides a form of authentication.

The two most common cryptographic techniques for message authentication are a **message authentication code** (MAC) and a **secure hash function.**

A MAC is an algorithm that requires the use of a secret key. A MAC takes a variable length message and a secret key as input and produces an authentication code. A recipient in possession of the secret key can generate an authentication code to verify the integrity of the message.

A hash function maps a variable-length message into a fixed length hash value, or message digest. For message authentication, a secure hash function must be combined in some fashion with a secret key.

Authentication Requirements

In the context of communications across a network, the following attacks can be identified:

1- **Disclosure:** Release of message contents to any person or process not possessing the appropriate cryptographic key.

2- **Traffic analysis:** Discovery of the pattern of traffic between parties. In a connection-oriented application, the frequency and duration of connections could be determined. In either a connection-oriented or connectionless environment, the number and length of messages between parties could be determined.
3- **Masquerade:** Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or non-receipt by someone other than the message recipient.
4- **Content modification:** Changes to the contents of a message, including insertion, deletion, transposition, and modification.
5- **Sequence modification:** Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.
6- **Timing modification:** Delay or replay of messages. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed. In a connectionless application, an individual message (e.g., datagram) could be delayed or replayed.
7- **Source repudiation:** Denial of transmission of message by source.
8- **Destination repudiation:** Denial of receipt of message by destination.

Measures to deal with the first two attacks are in the realm of message confidentiality. Measures to deal with items 3 through 6 in the foregoing list are generally regarded as **message authentication**.

Mechanisms for dealing specifically with item 7 come under the heading of **digital signatures**.

In summary, message authentication is a procedure to verify that received messages come from the alleged source and have not been altered. Message authentication may also verify sequencing and timeliness.

Authentication Functions

Any message authentication or digital signature mechanism has two levels of functionality.

- At the lower level, there must be some sort of function that produces an authenticator: a value to be used to authenticate a message.
- This lower-level function is then used as a primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message.

This lecture is concerned with the types of functions that may be used to produce an authenticator. These may be grouped into three classes, as follows:

A- Message encryption: The ciphertext of the entire message serves as its authenticator.

B- Message authentication code (MAC): A function of the message and a secret key that produces a fixed-length value that serves as the authenticator.

C- Hash function: A function that maps a message of any length into a fixed-length hash value, which serves as the authenticator.

A- Message Encryption

Message encryption by itself can provide a measure of authentication. The analysis differs for symmetric and public-key encryption schemes.

1. Symmetric Encryption

Consider the straightforward use of symmetric encryption (Figure 13.1). A message *M* transmitted from source A to destination B is encrypted using a secret key *K* shared by A and B. If no other party knows the key, then confidentiality is provided: No other party can recover the plaintext of the message.

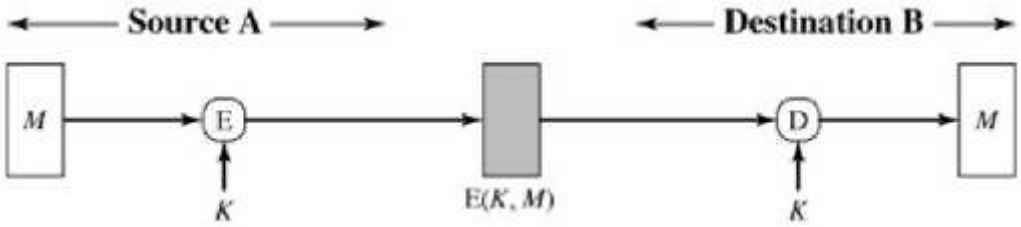

Figure (13.1) Symmetric encryption : Confidentiality and authentication

In addition, we may say that B is assured that the message was generated by A. Why?

The message must have come from A because A is the only other party that possesses *K* and therefore the only other party with the information necessary to construct ciphertext that can be decrypted with *K*. Furthermore, if *M* is recovered, B knows that none of the bits of *M* have been altered, because an opponent that does not know *K* would not know how to alter bits in the ciphertext to produce desired changes in the plaintext. So we may say that symmetric encryption provides authentication as well as confidentiality. However, this flat statement needs to be qualified. Consider exactly what is happening at B. Given a decryption function D and a secret key *K*, the destination will accept *any* input *X* and produce output $Y = D(K, X)$.

Append an error-detecting code, also known as a frame check sequence (FCS) or checksum, to each message before encryption, as

illustrated in Figure 13. a. A prepares a plaintext message *M* and then provides this as input to a function F that produces an FCS. The FCS is appended to *M* and the entire block is then encrypted. At the destination, B decrypts the incoming block and treats the results as a message with an appended FCS. B applies the same function F to attempt to reproduce the FCS. If the calculated FCS is equal to the incoming FCS, then the message is considered authentic.

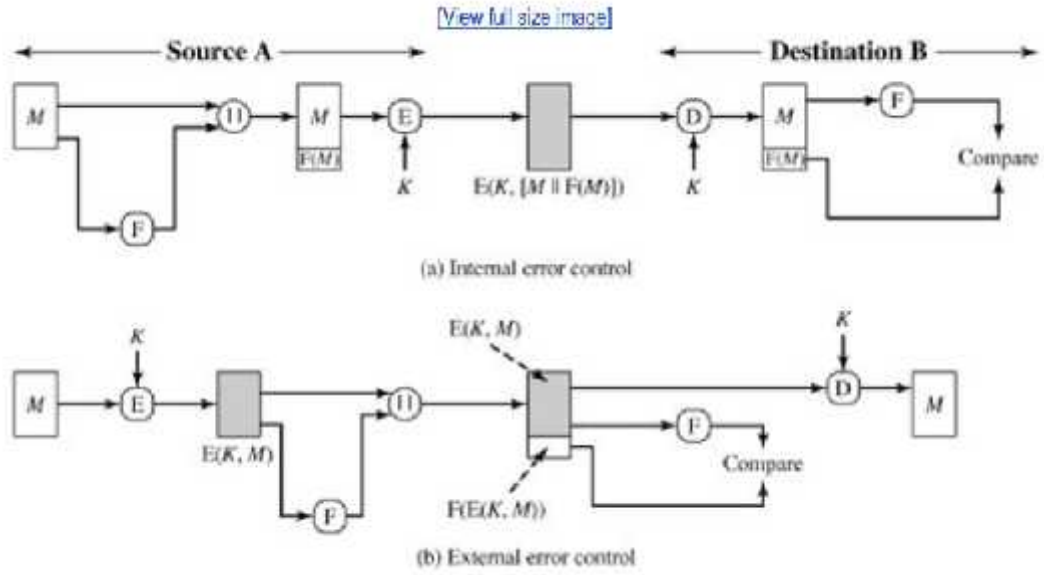

Figure 13.2. Internal and External Error Control

Note that the order in which the FCS and encryption functions are performed is critical. The sequence illustrated in Figure 13.2a is internal error control, which the authors contrast with external error control (Figure 13.2b). With internal error control, authentication is provided because an opponent would have difficulty generating ciphertext that, when decrypted, would have valid error control bits. If instead the FCS is the outer code, an opponent can construct messages with valid error control codes. Although the opponent cannot know what the decrypted plaintext will be, he or she can Still hope to create confusion and disrupt operations.

2- Public-Key Encryption

The straightforward use of public-key encryption (Figure 13.3a) provides confidentiality but not authentication. The source (A) uses the public key *PUb* of the destination (B) to encrypt *M*. Because only B has the corresponding private key *PRb* , only B can decrypt the message. This scheme provides no authentication because any opponent could also use B's public key to encrypt a message, claiming to be A.

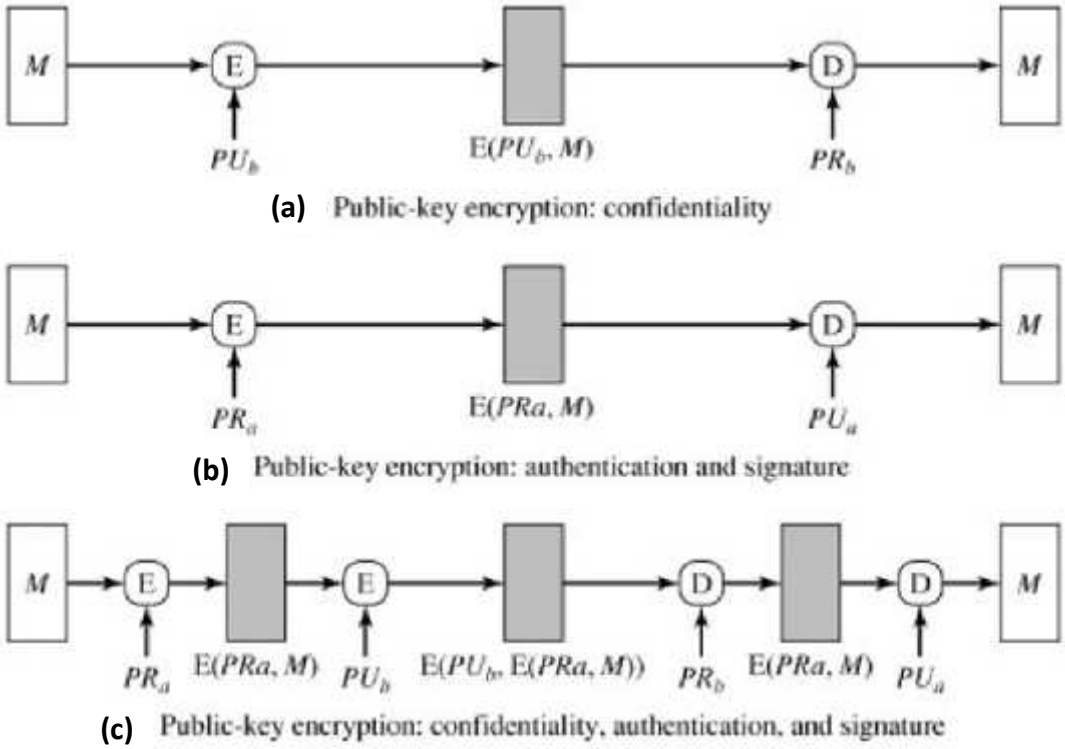

Figure (13.3) basic user of message encryption

To provide authentication, A uses its private key to encrypt the message, and B uses A's public key to decrypt (Figure 13.3b). This provides authentication using the same type of reasoning as in the symmetric encryption case: The message must have come from A because A is the only party that possesses *PRa* and therefore the only party with the information necessary to construct ciphertext that can be decrypted with *PU*a. Again, the same reasoning as before applies: There

must be some internal structure to the plaintext so that the receiver can distinguish between well-formed plaintext and random bits.

Assuming there is such structure, then the scheme of Figure 13.1b does provide authentication. It also provides what is known as digital signature.Only A could have constructed the ciphertext because only A possesses *PRa*. Not even B, the recipient, could have constructed the ciphertext. Therefore, if B is in possession of the ciphertext, B has the means to prove that the message must have come from A. In effect, A has "signed" the message by using its private key to encrypt. Note that this scheme does not provide confidentiality. Anyone in possession of A's public key can decrypt the ciphertext.

To provide both confidentiality and authentication, A can encrypt M first using its private key, which provides the digital signature, and then using B's public key, which provides confidentiality (Figure 13.3c). The disadvantage of this approach is that the public-key algorithm, which is complex, must be exercised four times rather than two in each communication.