produce legitimate plaintext.

- For example, suppose that only one bit pattern in 106 is legitimate plaintext. Then the probability that any randomly chosen bit pattern, treated as ciphertext, will produce a legitimate plaintext message is only 10-6. For a number of applications and encryption schemes, the desired conditions prevail as a matter of course.
- For example, suppose that we are transmitting Englishlanguage messages using a Caesar cipher with a shift of one (K = 1). A sends the following legitimate ciphertext:
  - `nbsftfbupbutboeepftfbupbutboemjuumfmbnctfbujwz`

  B decrypts to produce the following plaintext:
  `mareseatoatsanddoeseatoatsandlittlelambseativy`

- A simple frequency analysis confirms that this message has the profile of ordinary English. On the other hand, if an opponent generates the following random sequence of letters:

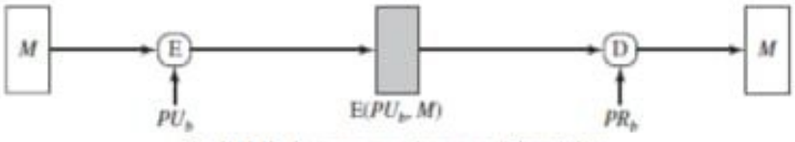
  `zuvrsoevgqxlzwigamdvnmhpmccxiuureosfbcebtqxsxq`

  this decrypts to

  `ytuqrndufpwkyvhfzlcumlgolbbwhttqdnreabdaspwrwp`

Public-Key Encryption

- The straightforward use of public-key encryption (Figure 12.1b) provides confidentiality but not authentication. The source (A) uses the public key PU_b of the destination (B) to encrypt M. Because only B has the corresponding private key PR_b, 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 and claim to be A.



(b) Public-key encryption: confidentiality

- To provide authentication, A uses its private key to encrypt the message, and B uses A's public key to decrypt (Figure 12.1c). 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 PR_a 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.

6. It is computationally infeasible to find any pair (x, y) such that H(x) = H(y). This is sometimes referred to as **strong collision resistance**.

- ✓ The first three properties are requirements for the practical application of a hash function to message authentication.
- ✓ The fourth property, the one-way property, states that it is easy to generate a code given a message but virtually impossible to generate a message given a code.
- ✓ The fifth property guarantees that an alternative message hashing to the same value as a given message cannot be found.
- ✓ This prevents forgery when an encrypted hash code is used (Figures b and c).
- ✓ The sixth property refers to how resistant the hash function is to a type of attack known as the birthday attack, which we examine shortly.

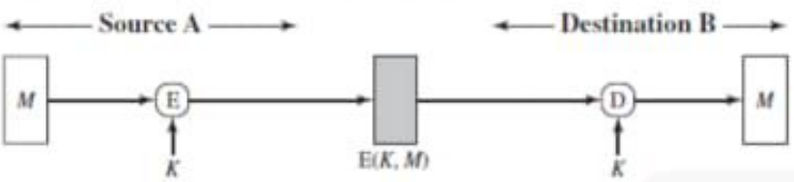
Message Encryption

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

Symmetric Encryption

- Consider the straightforward use of symmetric encryption (Figure 12.1a). 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.
- In addition, 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 the desired changes in the plaintext.

Figure 12.1 Basic Uses of Message Encryption



(a) Symmetric encryption: confidentiality and authentication

- Thus, in general, we require that only a small subset of all possible bit patterns be considered legitimate plaintext. In that case, any spurious ciphertext is unlikely to produce legitimate plaintext.

- For example, suppose that only one bit pattern in 106 is legitimate plaintext. Then the probability that any randomly chosen bit pattern, treated as ciphertext, will produce a legitimate plaintext message is only 10-6. For a number of applications and encryption schemes, the desired conditions prevail as a matter of course.
- For example, suppose that we are transmitting Englishlanguage messages using a Caesar cipher with a shift of one (K = 1). A sends the following legitimate ciphertext:
  - `nbsftfbupbutboeepftfbupbutboemjuumfmbnctfbujwz`

B decrypts to produce the following plaintext:

`mareseatoatsanddoeseatoatsandlittlelambseativy`

Sender:

- The sender creates a message using SHA to generate a 160 bit hashcode.
- The message and hashcode is concatenated and the result is encrypted using symmetric Encryption Algorithm.

Receiver:

- The receiver uses RSA (or) DSA algorithm to decrypt the message and recover hashcode.
- The receiver generates a new hashcode for the message and compare it with decrypted hashcode
- If two hashcodes are match ,the message is accepted,else it is rejected.

Fig .b. Encrypt hash code shared secret key

Sender:

- The sender creates a message using SHA to generate a 160 bit hashcode.
- Only the hash code is encrypted using Symmetric Encryption.
- The message and hash code encrypted and result is concatented

Receiver:

- The receiver uses RSA (or) DSA algorithm to decrypt the message and recover hashcode.
- The receiver generates a new hashcode for the message and compare it with decrypted hashcode
- If two hashcodes are match ,the message is accepted,else it is rejected.

Fig .c. Encrypt hash code sender's private key

Sender:

- The sender creates a message using SHA to generate a 160 bit hashcode.
- Only the hash code is encrypted using public key encryption and using the sender's private key.
- The message and hash code encrypted and result is concatented

Receiver:

- The receiver uses RSA (or) DSA algorithm to decrypt the message and recover hashcode.
- The receiver generates a new hashcode for the message and compare it with decrypted hashcode
- If two hashcodes are match ,the message is accepted,else it is rejected.

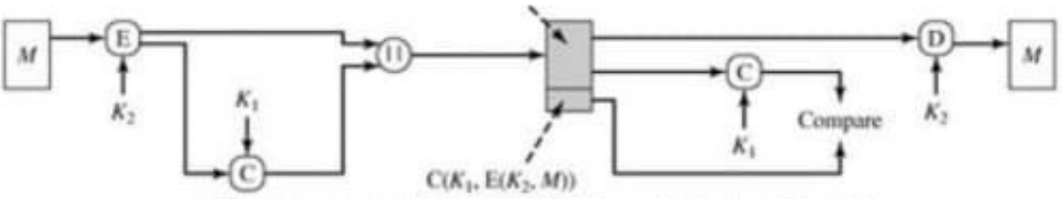
D) **Sender:**

- The sender creates a message using SHA to generate a 160 bit hashcode.
- Only the hash code is encrypted using public key encryption and using the sender's private key.
- The message and hash code encrypted and result is concatented

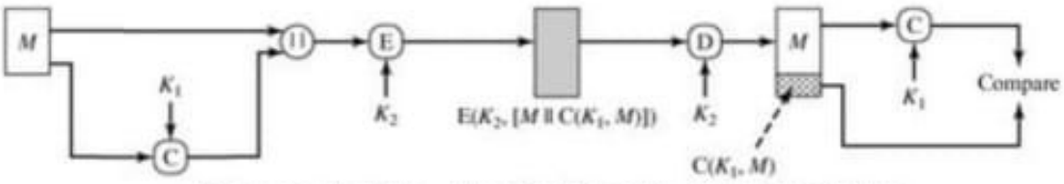
Receiver:

- The receiver uses RSA (or) DSA algorithm to decrypt the message and

recover hashcode.



(c) Message authentication and confidentiality; authentication tied to ciphertext



(b) Message authentication and confidentiality; authentication tied to plaintext

$E(K_2, M)$

- The process depicted in Figure a provides authentication but not confidentiality, because the message as a whole is transmitted in the clear.
- Confidentiality can be provided by performing message encryption either after (Figure .b) or before (Figure c) the MAC algorithm. In both these cases, two separate keys are needed, each of which is shared by the sender and the receiver.
- In the first case, the MAC is calculated with the message as input and is then concatenated to the message. The entire block is then encrypted. In the second case, the message is encrypted first.
- Then the MAC is calculated using the resulting ciphertext and is concatenated to the ciphertext to form the transmitted block. Typically, it is preferable to tie the authentication directly to the plaintext, so the method of Figure b is used.

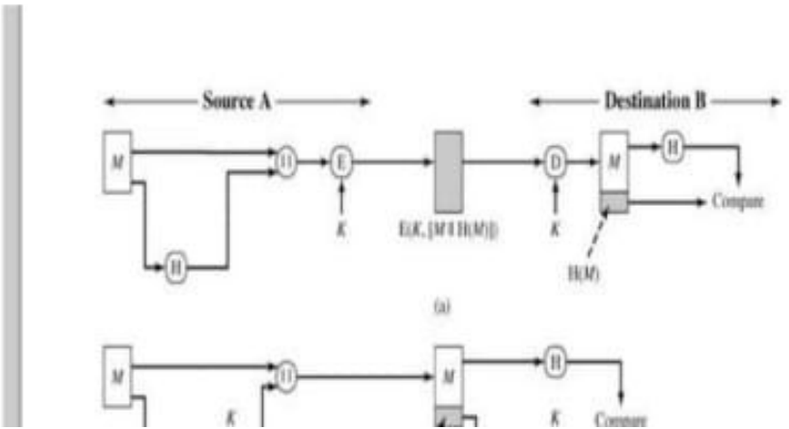
Application of MAC

- Application in message is broadcast to a number of destinations.
- Authentication of a computer program in plain text is an attractive service

4.4. HASH FUNCTION

- A variation on the message authentication code is the one way hash function. As with MAC, a hash function accepts a variable size message M as input and produces affixed-size output, referred to as hash code h=**H(M).**

- Unlike a MAC, a hash code does not use a key but is a function only of the input message. The hash code is also referred to as a **message digest or hash value.**



4.2. AUTHENTICATION FUNCTION

- 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.
- These may be grouped into three classes
  - ***Hash function***
  - ***Message Encryption***

  - ***Message Authentication Code***

• **Hash function:** A function that maps a message of any length into a fixedlength hash value, which serves as the authenticator
• **Message encryption:** The ciphertext of the entire message serves as its authenticator
• **Message authentication code (MAC):** A function of the message and a secret key that produces a fixed-length value that serves as the authenticator

4.3. MAC - MESSAGE AUTHENTICATION CODE

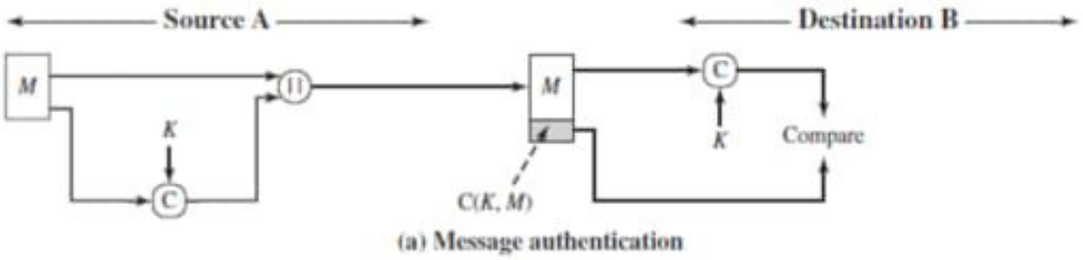
- An alternative authentication technique involves the use of a secret key to generate a small fixed-size block of data, known as a cryptographic checksum or MAC, that is appended to the message. This technique assumes that two communicating parties, say A and B, share a common secret key K.
- When A has a message to send to B, it calculates the MAC as a function of the message and the key:

$$MAC = C(K, M)$$

where

M = input message
C = MAC function
K = shared secret key
MAC = message authentication code

- If we assume that only the receiver and the sender know the identity of the secret key, and if the received MAC matches the calculated MAC, then
  1. The receiver is assured that the message has not been altered.
  2. The receiver is assured that the message is from the alleged sender.
  3. If the message includes a sequence number (such as is used with HDLC, X.25, and TCP), then the receiver can be assured of the proper sequence because an attacker cannot successfully alter the sequence number.



(a) Message authentication

recover hashcode.

- The receiver generates a new hashcode for the message and compare it with decrypted hashcode which uses the public key Encryption Algorithm of public key of sender.
- If two hashcodes are match ,the message is accepted,else it is rejected.

E) Sender:

- The sender creates a message M using SHA to generate a 160 bit hashcode.\
- This technique uses a hash fuction,but no encryption for message authentication
- This technique assumes that the two communicating parties share a common secret value 'S'.
- The source computes the hash value over the concatenation of M and S and appends the resulting hashvalue to M.

Receiver:

- The receiver uses RSA (or) DSA algorithm to decrypt the message and recover hashcode.
- The receiver generates a new hashcode for the message and compare it with decrypted hashcode which uses the public key Encryption Algorithm of public key of sender.
- The Message concatenated with hash value and 'S' is compared with receiver 's hash value.
- If two hashcodes are match ,the message is accepted,else it is rejected.

f) Confidentiality can be added to the previous approach by encrypting the entire message plus the hash code.

Requirements for a Hash Function

1. H can be applied to a block of data of any size.

2.H produces a fixed-length output.

3. H(x) is relatively easy to compute for any given x, making both hardware and Software implementations practical.

4. For any* given value h, it is computationally infeasible to find x such that H(x) =h. This is sometimes referred to in the literature as the one-way property.

5. For any given block x, it is computationally infeasible to find y x such that H(y) = H(x). This is
sometimes referred to as weak hash function.

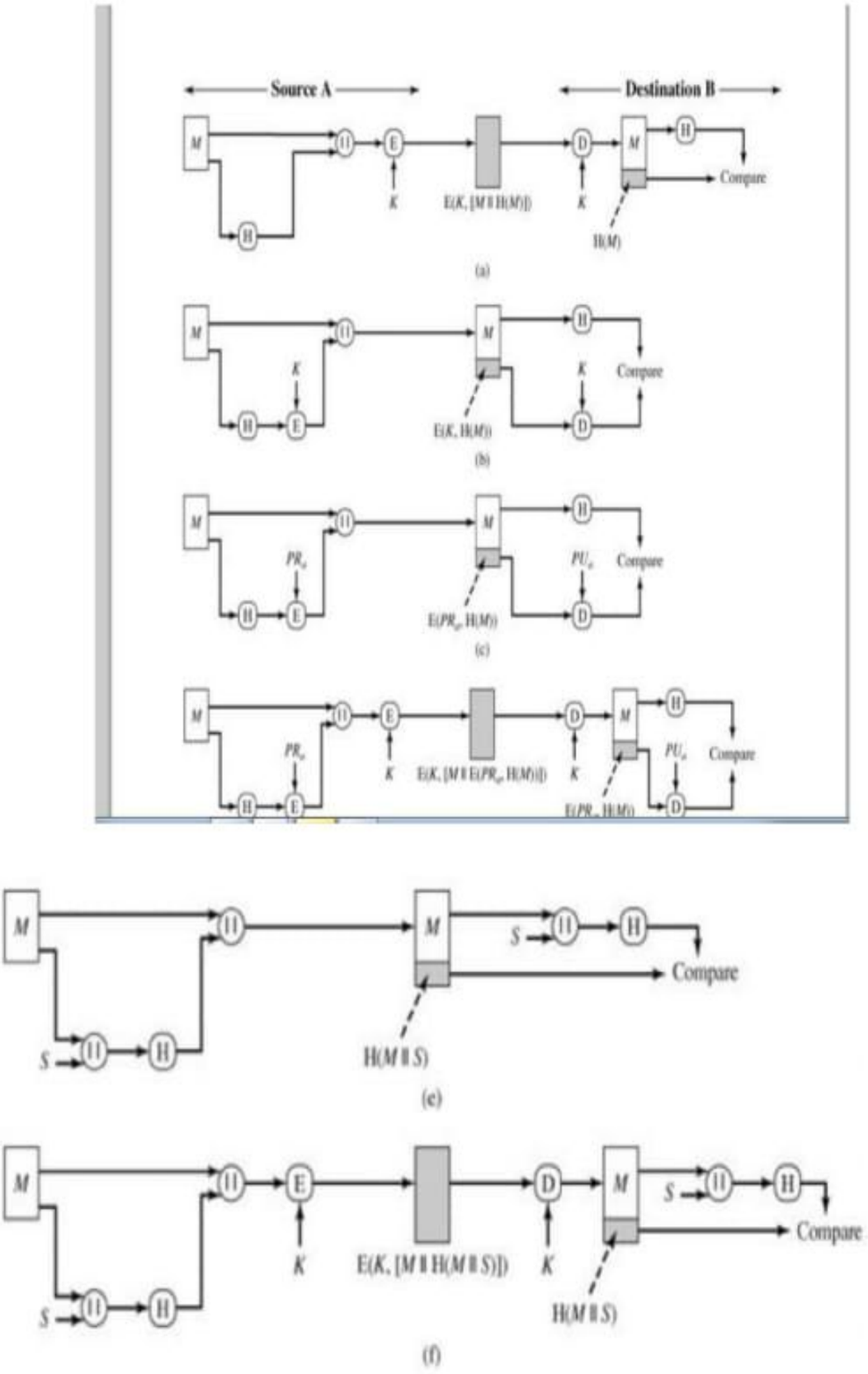


Fig .a. Encrypt message plus hash code