Information & System Security

Lecture 31

>>Encrytion
>>Integrity
>>Identification
>>Authentication

VIT-AP UNIVERSITY

Message Integrity & Authentication

11-1 MESSAGE INTEGRITY

The cryptography systems that we have studied so far provide secrecy, or confidentiality, but not integrity.

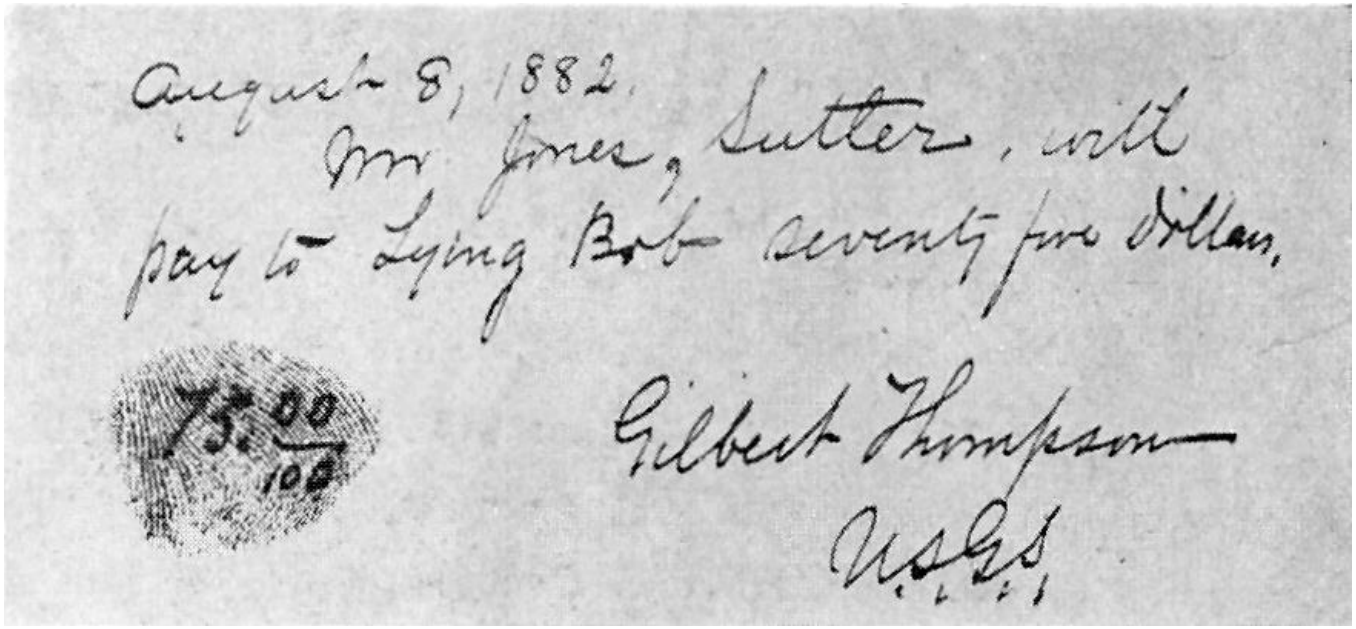
However, there are occasions where we may not even need secrecy but instead must have integrity.

Topics discussed in this section:

11.1.1 Document and Fingerprint

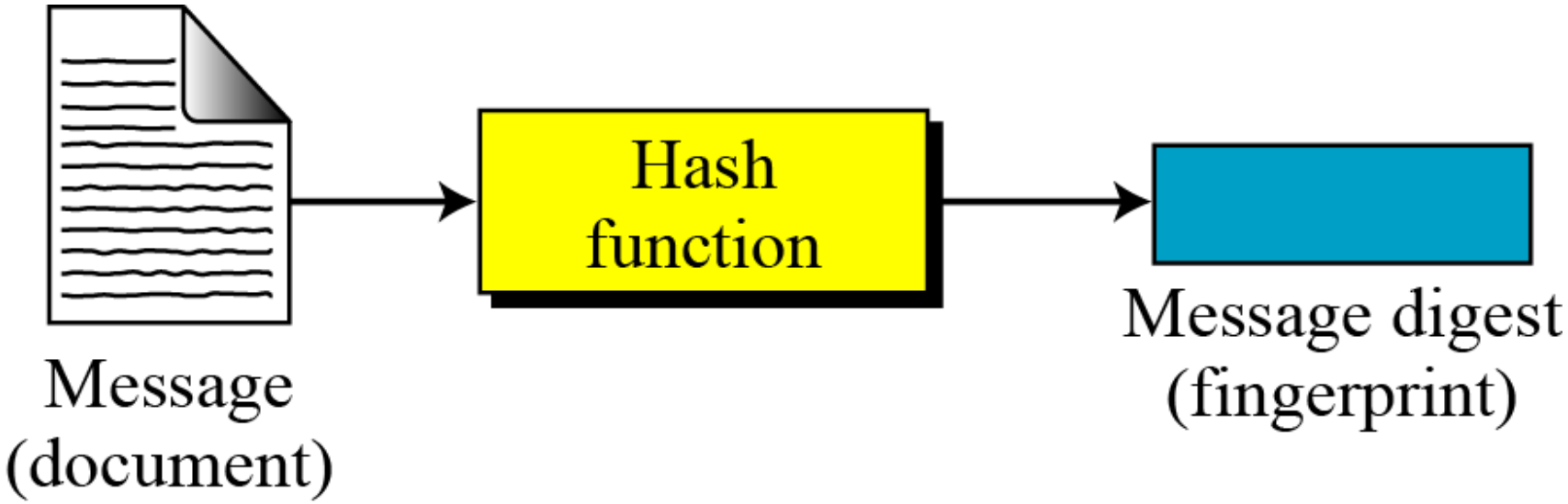
- ***One way to preserve the integrity of a document is through the use of a fingerprint.***

- ***If Alice needs to be sure that the contents of her document will not be changed, she can put her fingerprint at the bottom of the document.***



11.1.2 Message and Message Digest

The electronic equivalent of the document and fingerprint pair is the message and digest pair.



Message and digest

11.1.3 Difference

- *The two pairs (document/fingerprint) and (message/ message digest) are similar, with some differences.*

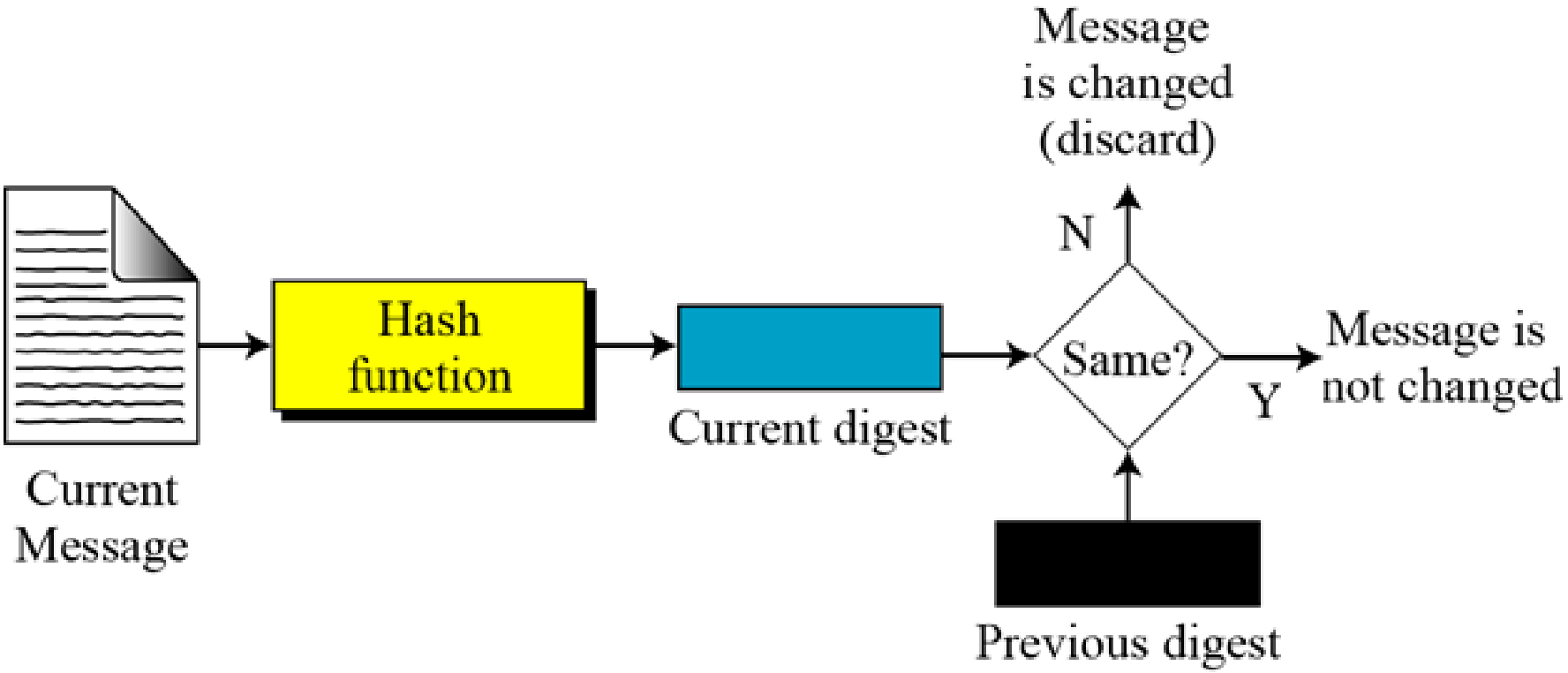
- *The document and fingerprint are physically linked together.*

- *The message and message digest can be unlinked separately, and, most importantly, the message digest needs to be safe from change.*

Note

The message digest needs to be safe from change.

11.1.4 Checking Integrity

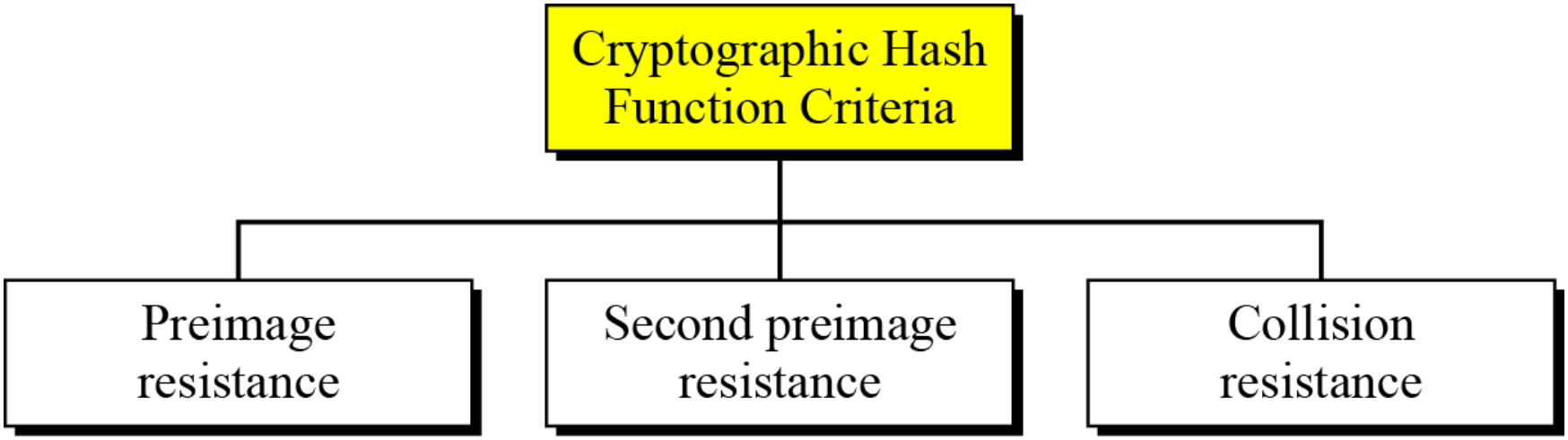


Checking integrity

11.1.5 Cryptographic Hash Function Criteria

A cryptographic hash function must satisfy three criteria:

- ***preimage resistance***
- ***second preimage resistance***
- ***collision resistance***



Criteria of a cryptographic hash function

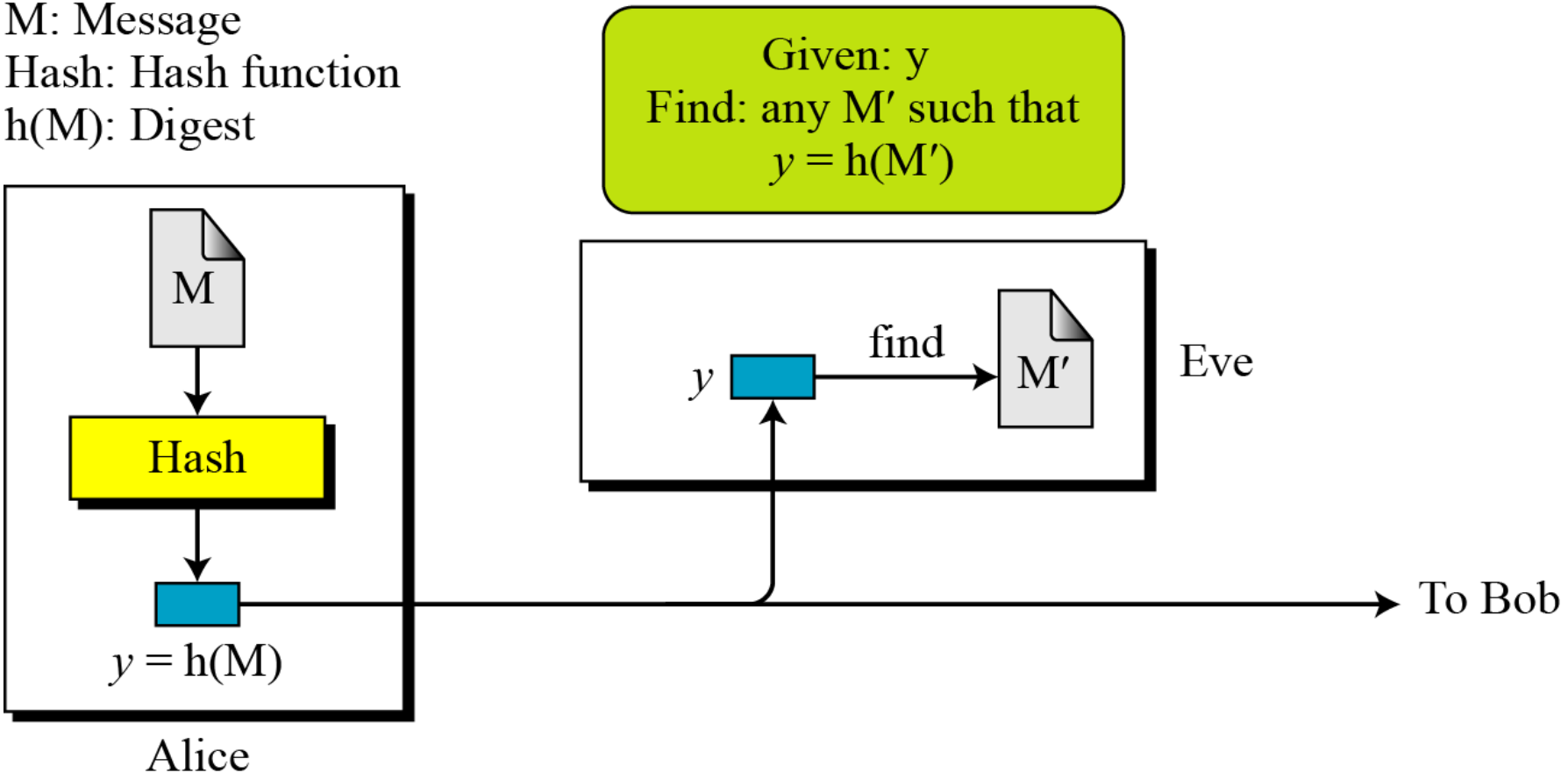
11.1.5 Continued

Preimage Resistance

Preimage Attack

Given: y = h(M) Find: M′ such that y = h(M′)

M: Message
Hash: Hash function
h(M): Digest



Example

Can we use a conventional lossless compression method such as *StuffIt* as a cryptographic hash function?

Solution

We cannot. A lossless compression method creates a compressed message that is reversible.

Example

Can we use a checksum function as a cryptographic hash function?

Solution

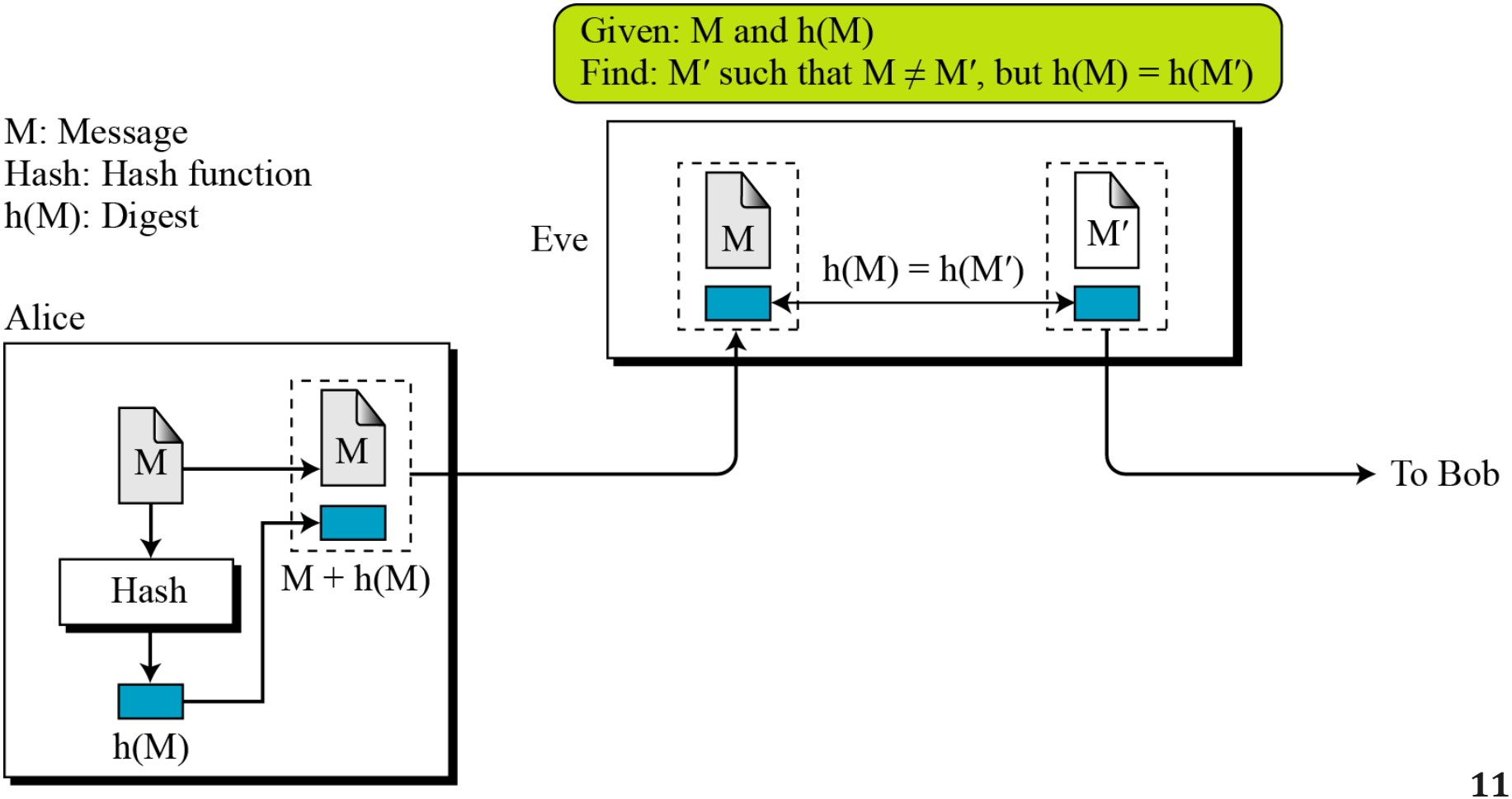
We cannot. A checksum function is not preimage resistant, Eve may find several messages whose checksum matches the given one.

11.1.5 Continued

Second Preimage Resistance

Second Preimage Attack

Given: M and h(M) Find: M′ ≠ M such that h(M) = h(M′)



11.1.5 Continued

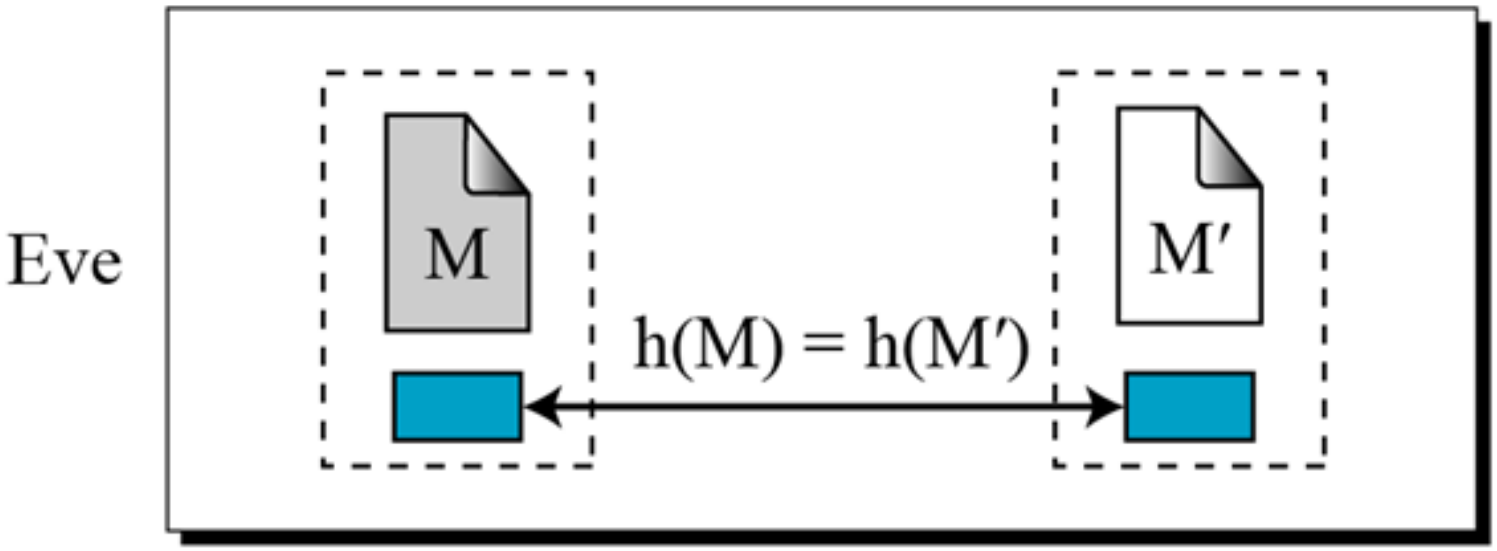
Collision Resistance

Collision Attack

Given: none **Find: $M' \neq M$ such that $h(M) = h(M')$**

M: Message Hash: Hash function h(M): Digest

Find: M and M′ such that $M \neq M'$, but $h(M) = h(M')$



11-2 MESSAGE AUTHENTICATION

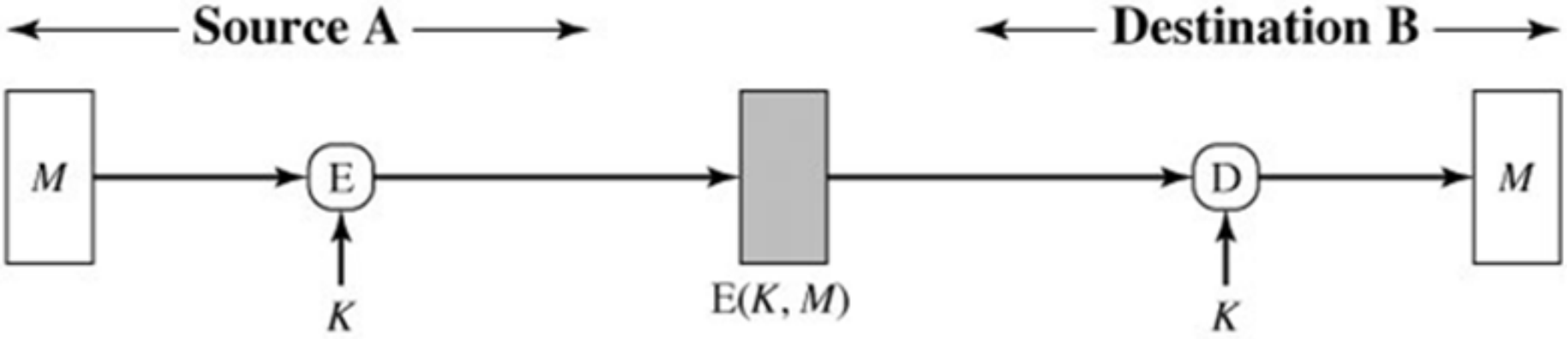
- *A message digest does not authenticate the sender of the message.*
- *To provide message authentication, Alice needs to provide proof that it is Alice sending the message and not an imposter.*
- *The digest created by a cryptographic hash function is normally called a modification detection code (MDC).*
- *What we need for message authentication is a message authentication code (MAC).*

Topics discussed in this section:

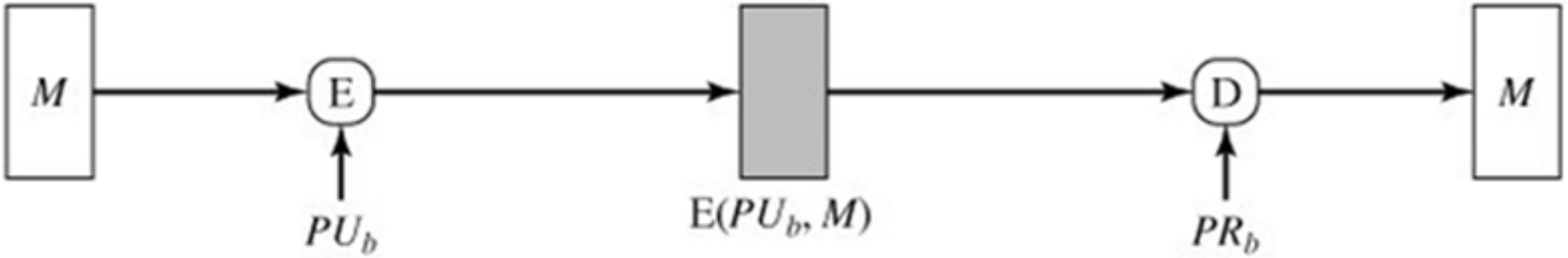
11.2.1 Modification Detection Code (MDC)

11.2.2 Message Authentication Code (MAC)

Basic Uses of Message Encryption

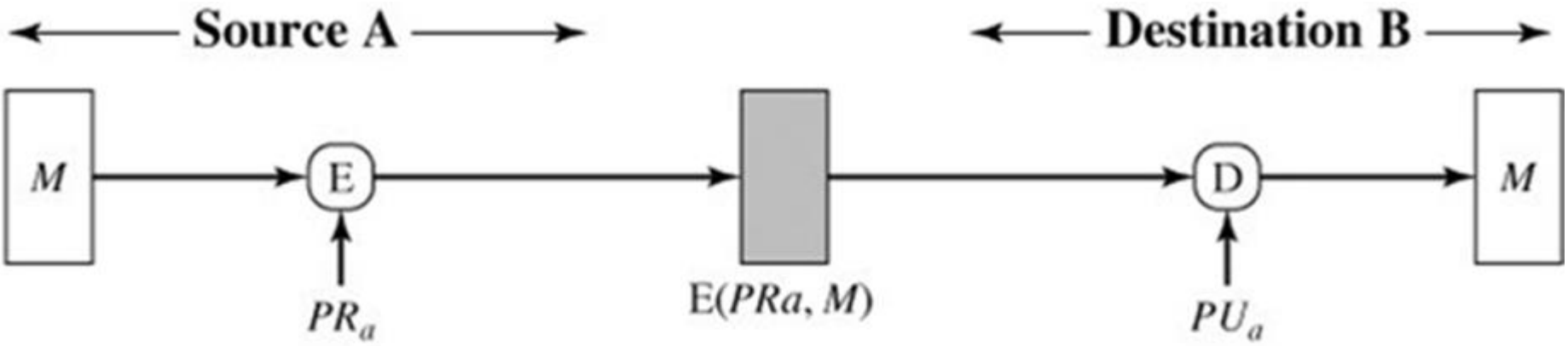


(a) Symmetric encryption: confidentiality and authentication



(b) Public-key encryption: confidentiality

Basic Uses of Message Encryption



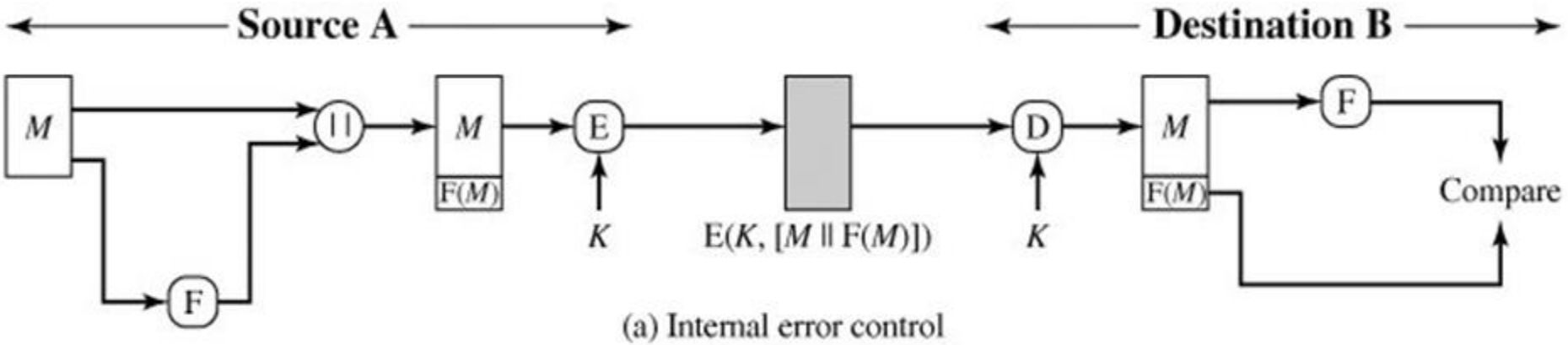
(c) Public-key encryption: authentication and signature

(d) Public-key encryption: confidentiality, authentication, and signature

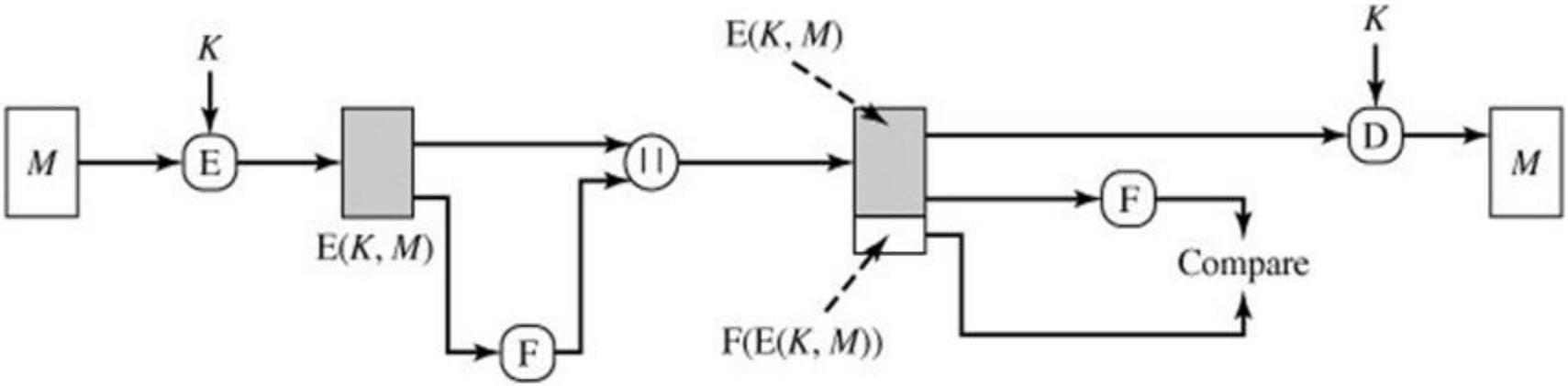
11.2 Continued

Internal and External Error Control

In internal error control, an error detecting code also known as checksum is used. In external error control, error detecting codes are appended after encryption.



(a) Internal error control



(b) External error control

11.2.1 Modification Detection Code (MDC)

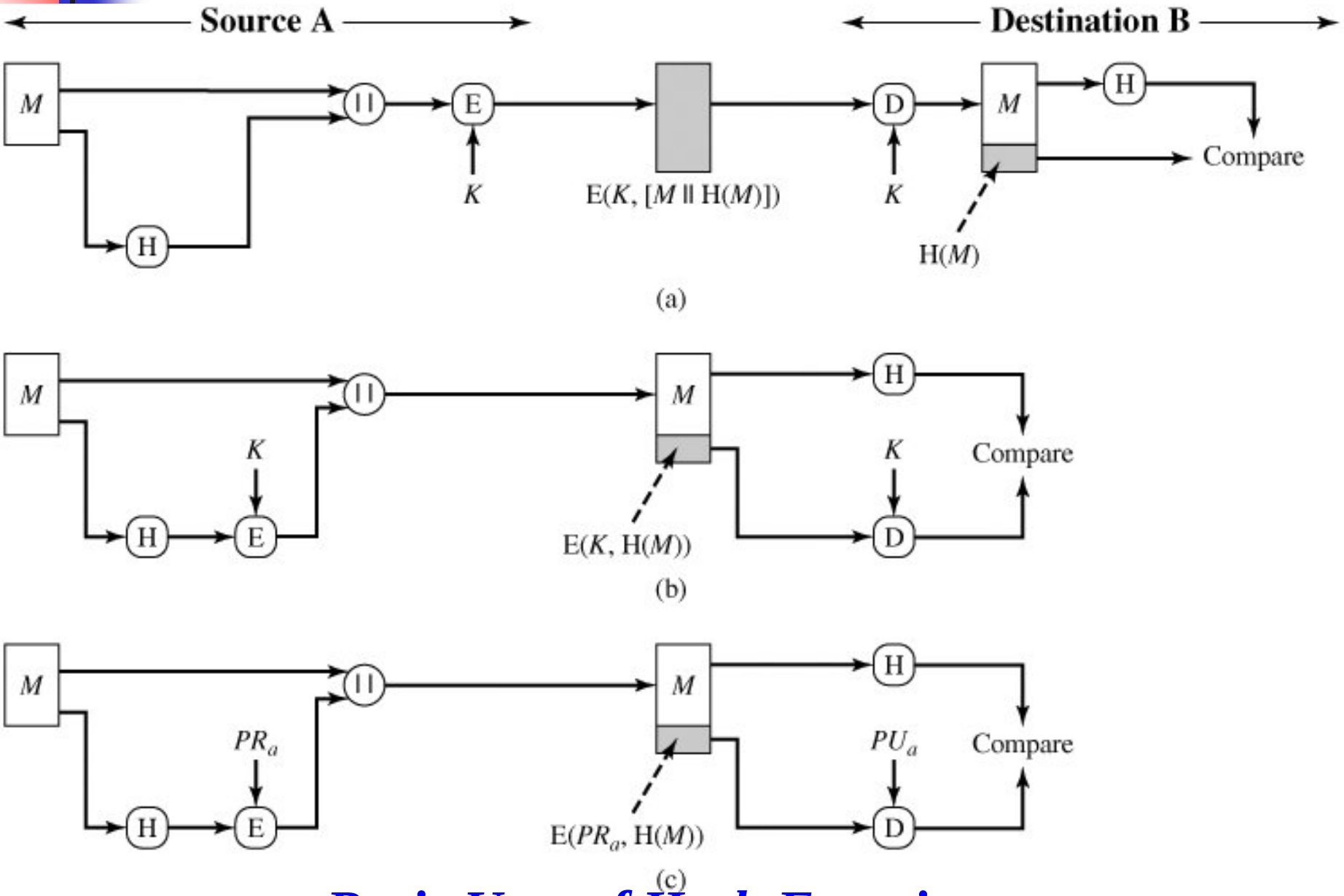
- *A modification detection code (MDC) is a message digest that can prove the integrity of the message: that message has not been changed.*

- *If Alice needs to send a message to Bob and be sure that the message will not change during transmission, Alice can create a message digest, MDC, and send both the message and the MDC to Bob.*

- *Bob can create a new MDC from the message and compare the received MDC and the new MDC.*

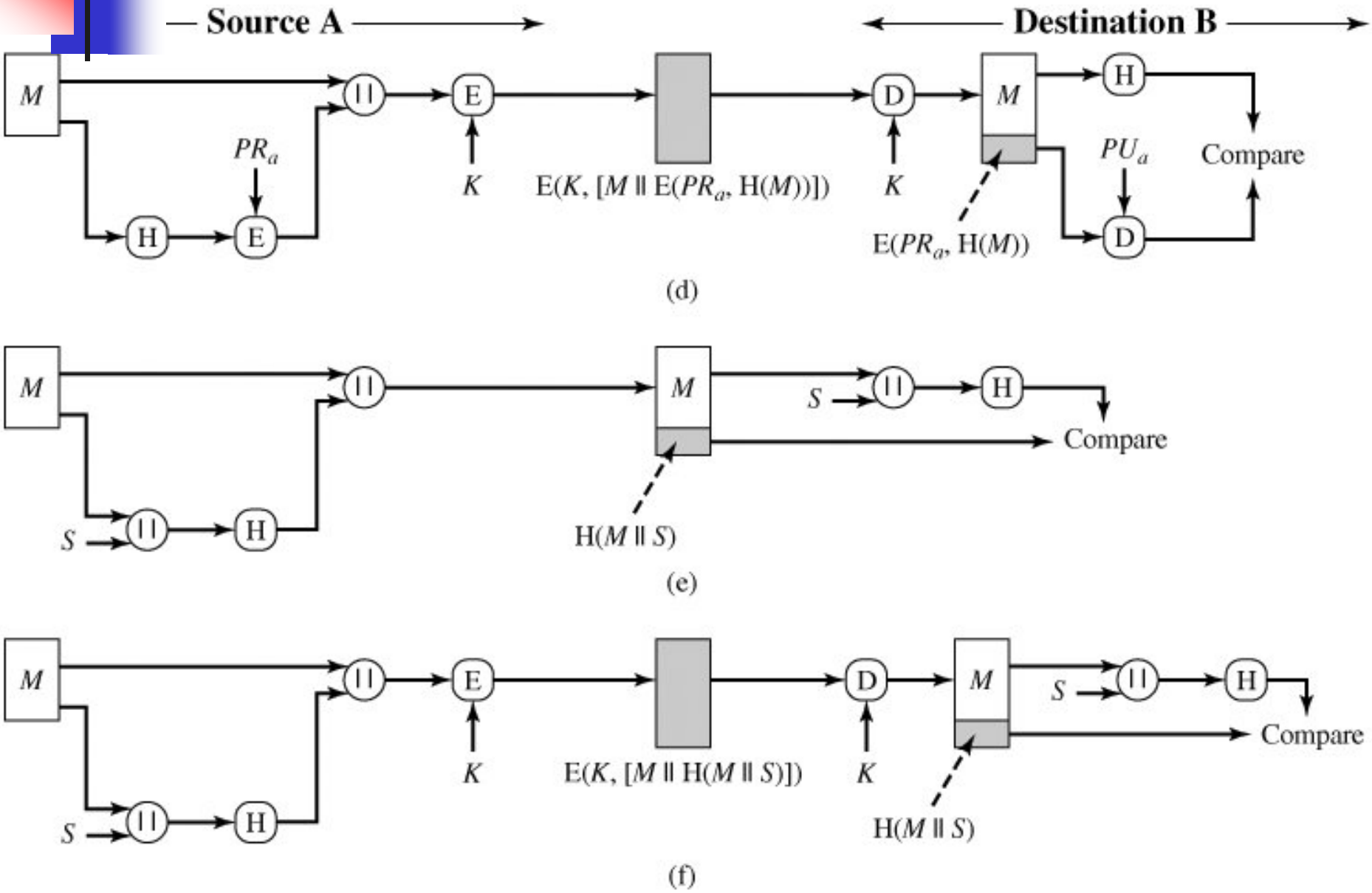
- *If they are the same, the message has not been changed.*

11.2.1 Continued



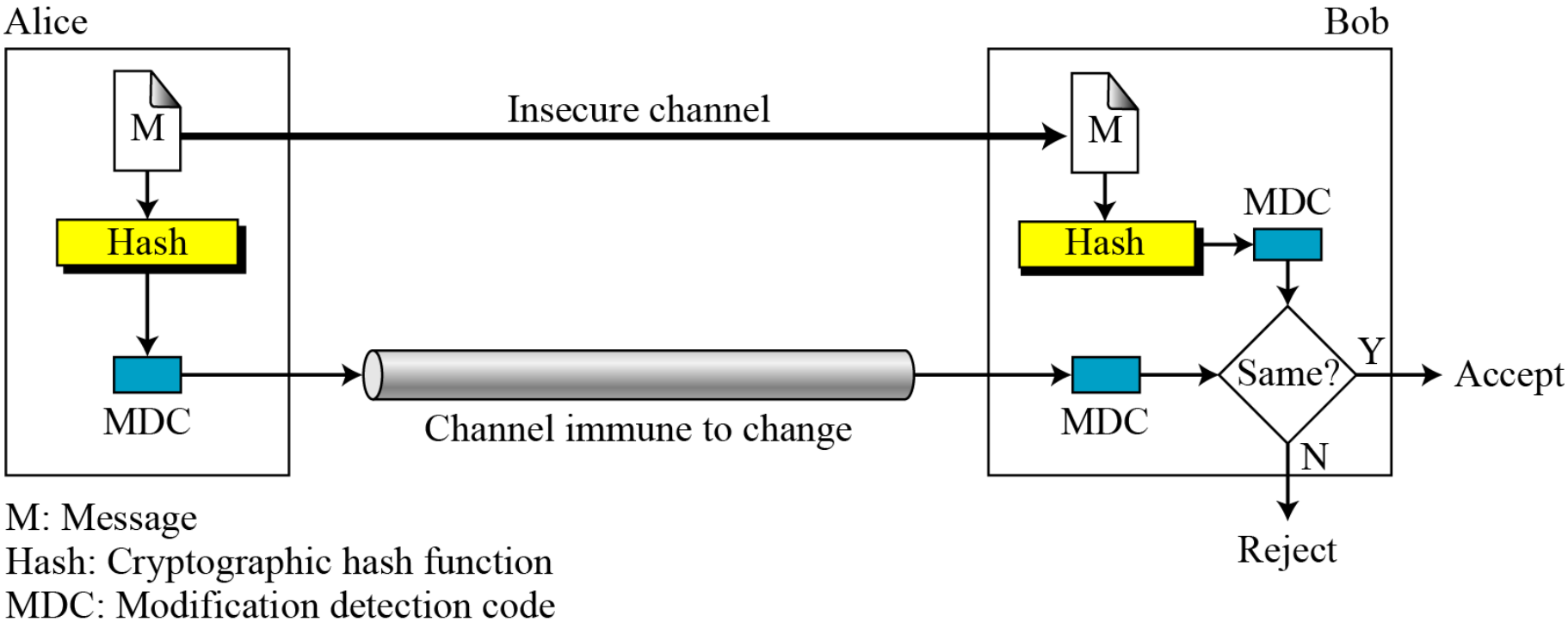
Basic Uses of Hash Function

11.2.1 Continued



Basic Uses of Hash Function

11.2.1 Continued



M: Message
Hash: Cryptographic hash function
MDC: Modification detection code

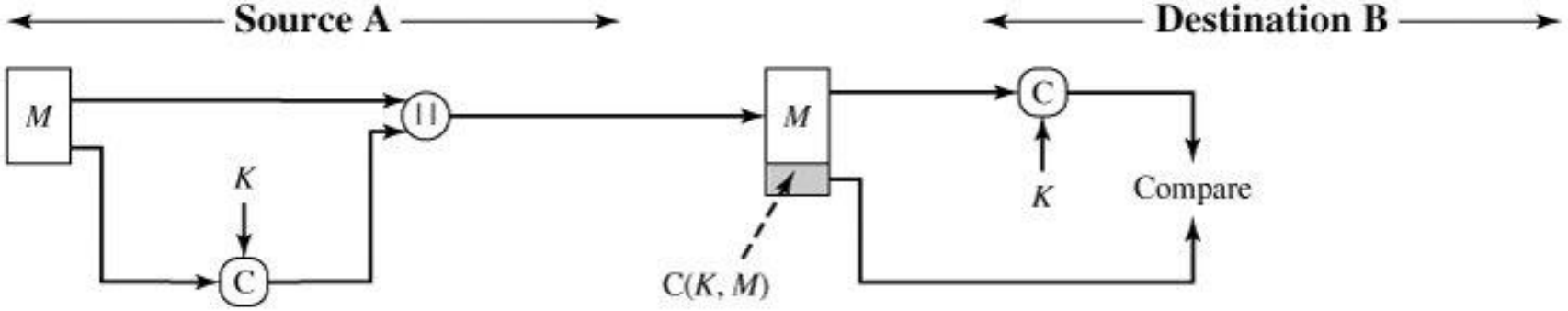
Modification detection code (MDC)

11.2.2 Message Authentication Code (MAC)

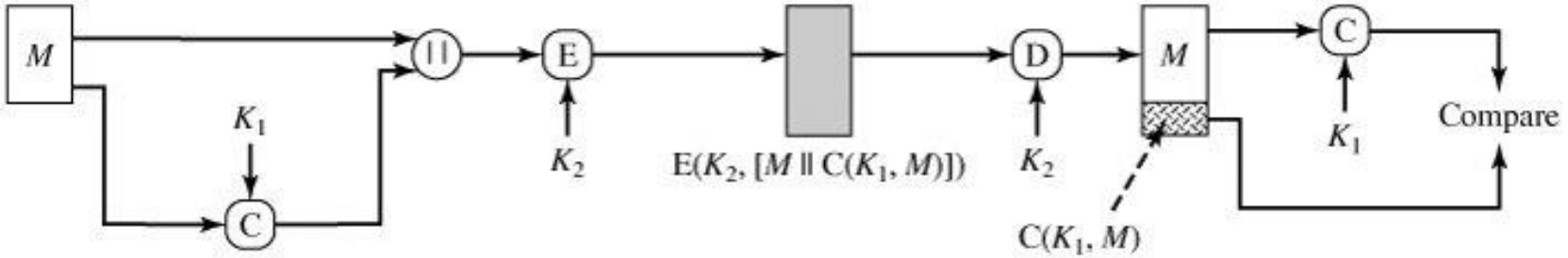
- *To ensure the integrity of the message and the data origin authentication—we need to change MDC to an MAC.*

- *MAC includes a secret (key) between the Sender and the Receiver.*

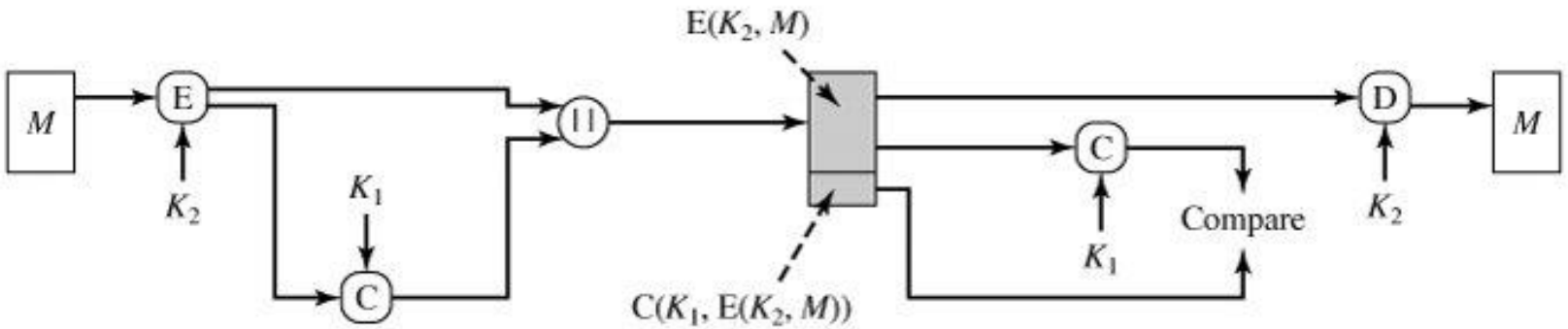
11.2.2 Continued



(a) Message authentication



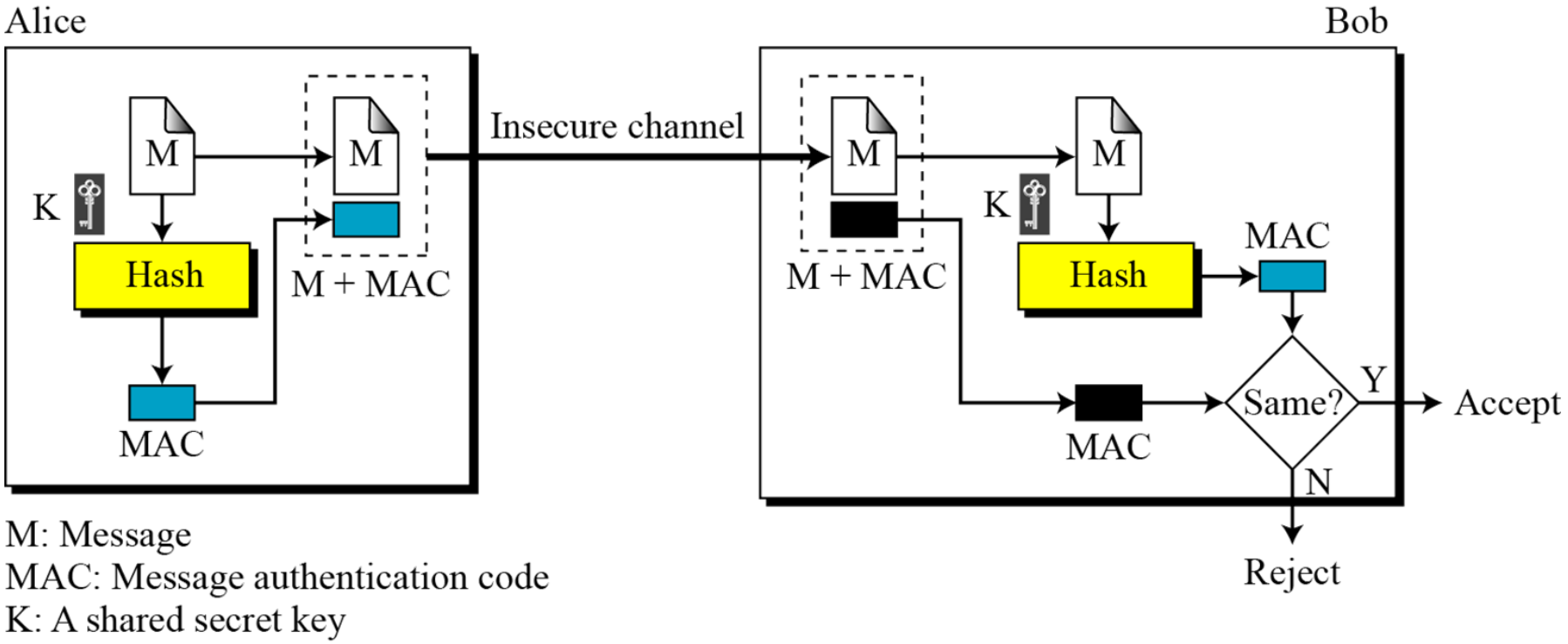
(b) Message authentication and confidentiality; authentication tied to plaintext



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

Basic Uses of Message Authentication Code

11.2.2 Continued



M: Message
MAC: Message authentication code
K: A shared secret key

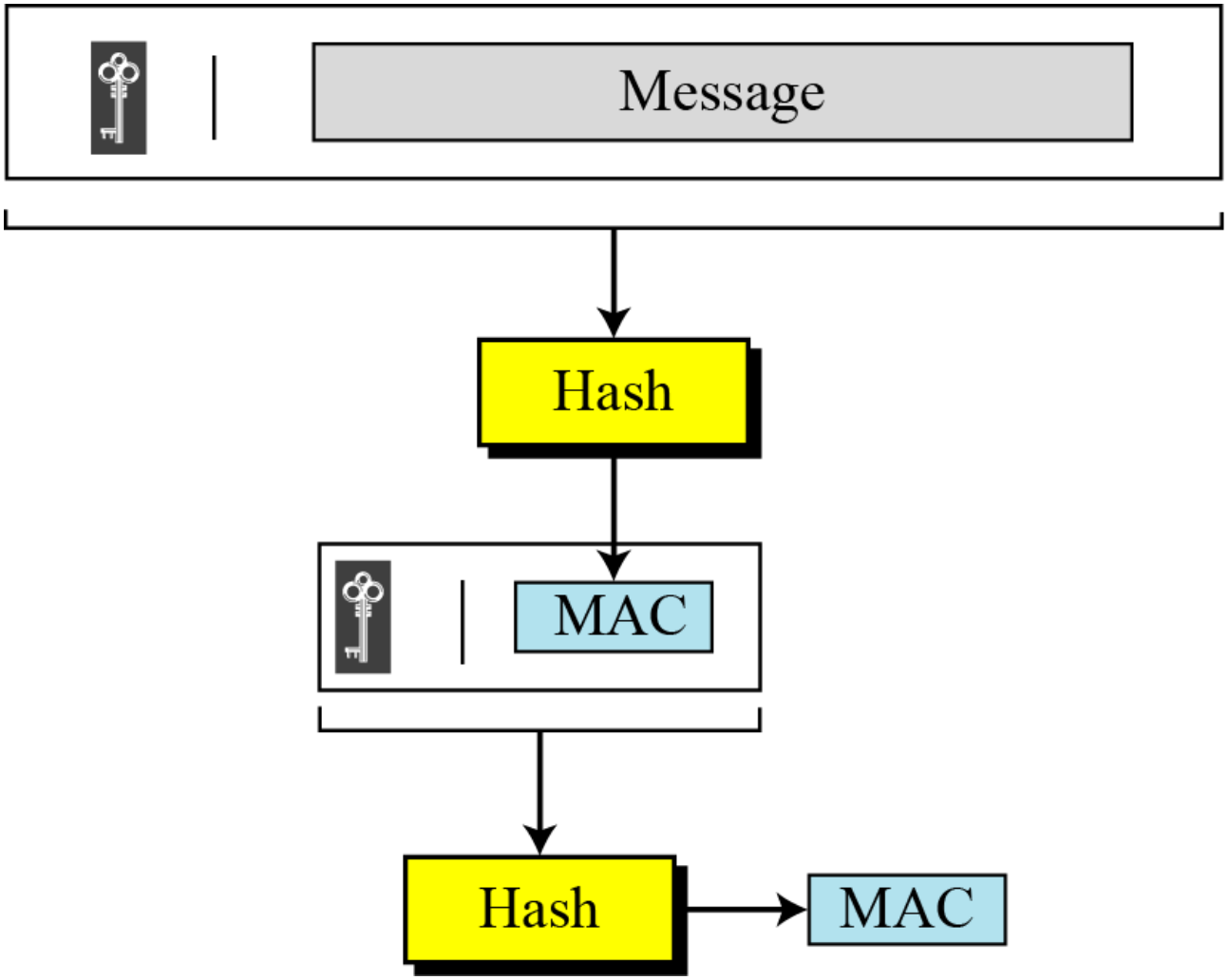
Message authentication code

Note

The security of a MAC depends on the security of the underlying hash algorithm.

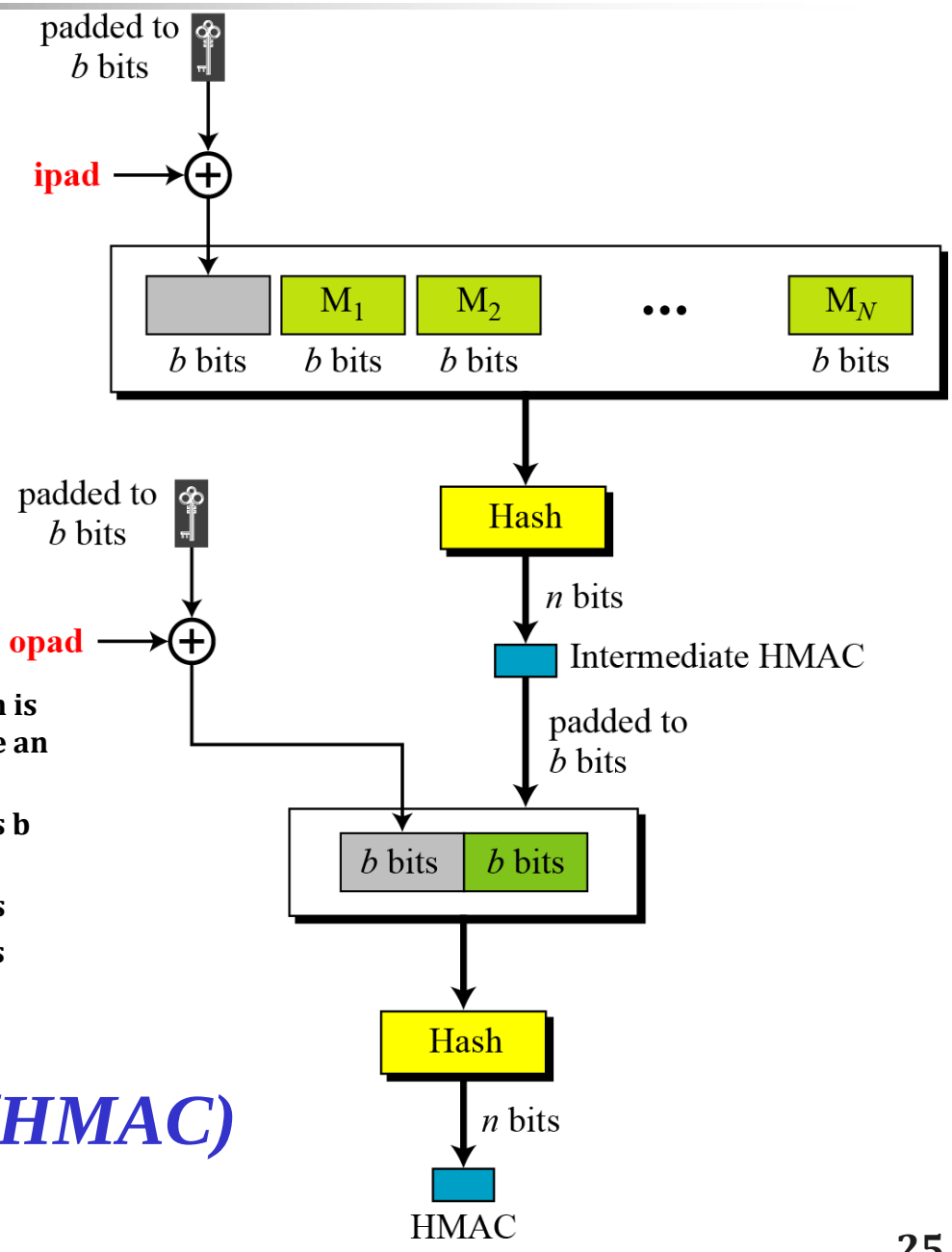
11.2.2 Continued

Nested MAC



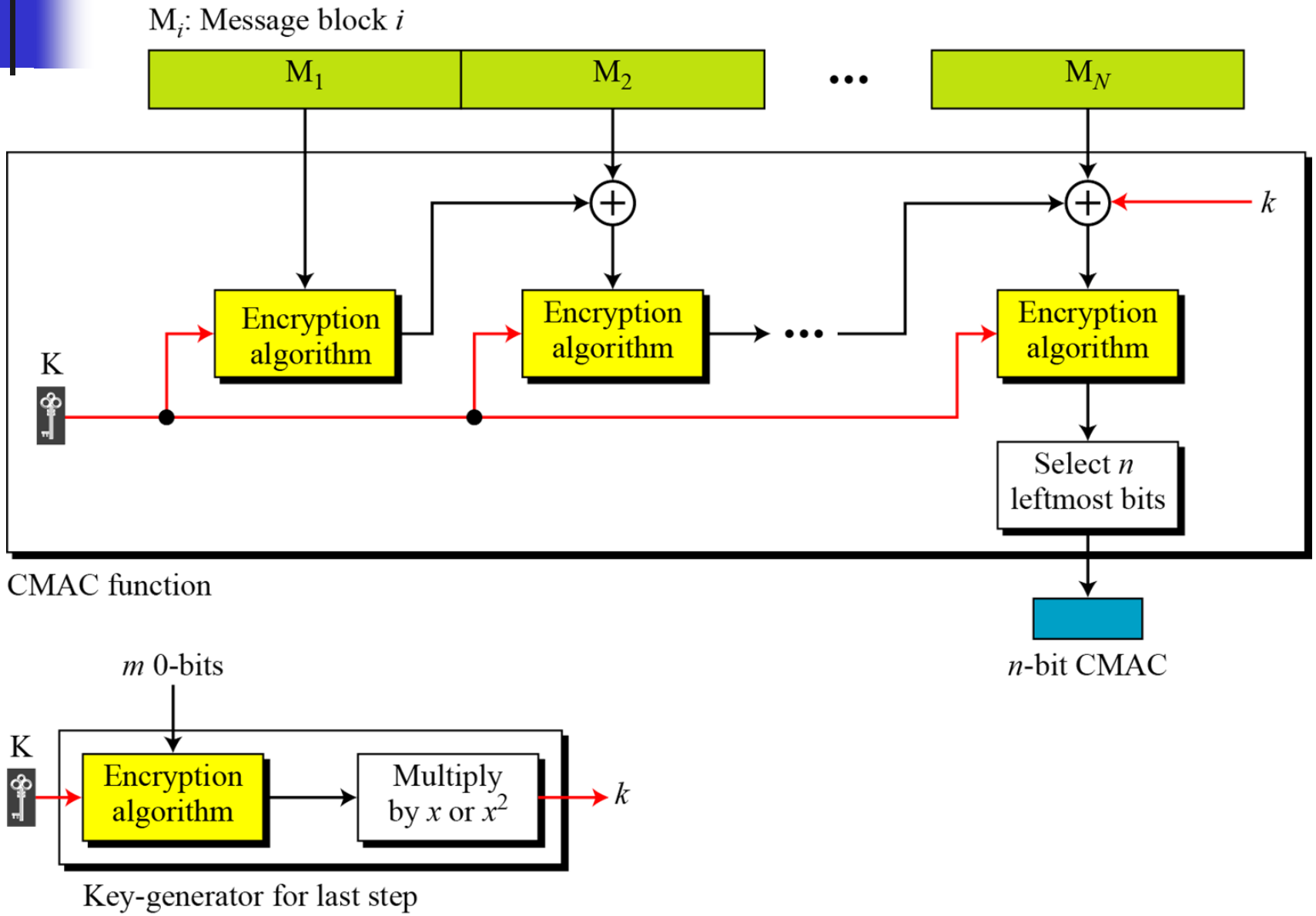
11.2.2 Continued

H	**embedded hash function (e.g., MD5, SHA-1)**
IV	**initial value input to hash function**
M	**message input to HMAC(including the padding specified in the embedded hash function)**
N	**number of blocks in M**
b	**number of bits in a block**
n	**length of hash code produced by hash function**
K	**secret key recommended length is ≥n; if key length is >b; the key is input to the hash function to produce an n-bit key**
K'	**K padded with zeros on the left so that the result is b bits in length**
ipad	**00110110 (36 in hexadecimal) repeated b/8 times**
opad	**01011100 (5C in hexadecimal) repeated b/8 times**



Hash-based MAC (HMAC)

11.2.2 Continued



Cipher-Based Message Authentication Code (CMAC)

References

- **Chapter 11 -** Behrouz A Forouzan, Debdeep Mukhopadhyay, Cryptography and Network Security, Mc Graw Hill, 3rd Edition, 2015.

- **Chapter 11 -** William Stallings, Cryptography and Network Security Principles and Practices, 7th Edition, Pearson Education, 2017.