

# CHAPTER 3

## Public Key Cryptography and Message Authentication

*Every Egyptian received two names, which were known respectively as the true name and the good name, or the great name and the little name; and while the good or little name was made public, the true or great name appears to have been carefully concealed.*

—*The Golden Bough, Sir James George Frazer*

*To guard against the baneful influence exerted by strangers is therefore an elementary dictate of savage prudence. Hence before strangers are allowed to enter a district, or at least before they are permitted to mingle freely with the inhabitants, certain ceremonies are often performed by the natives of the country for the purpose of disarming the strangers of their magical powers, or of disinfecting, so to speak, the tainted atmosphere by which they are supposed to be surrounded.*

—*The Golden Bough, Sir James George Frazer*

# APPROACHES TO MESSAGE AUTHENTICATION

## Using conventional encryption

- Symmetric encryption alone is not a suitable tool for data authentication
  - We assume that only the sender and receiver share a key, so only the genuine sender would be able to encrypt a message successfully
  - The receiver assumes that no alterations have been made and that sequencing is proper if the message includes an error detection code and a sequence number
  - If the message includes a timestamp, the receiver assumes that the message has not been delayed beyond that normally expected for network transit

## Without message encryption

- An authentication tag is generated and appended to each message for transmission
- The message itself is not encrypted and can be read at the destination independent of the authentication function at the destination
- Because the message is not encrypted, message confidentiality is not provided

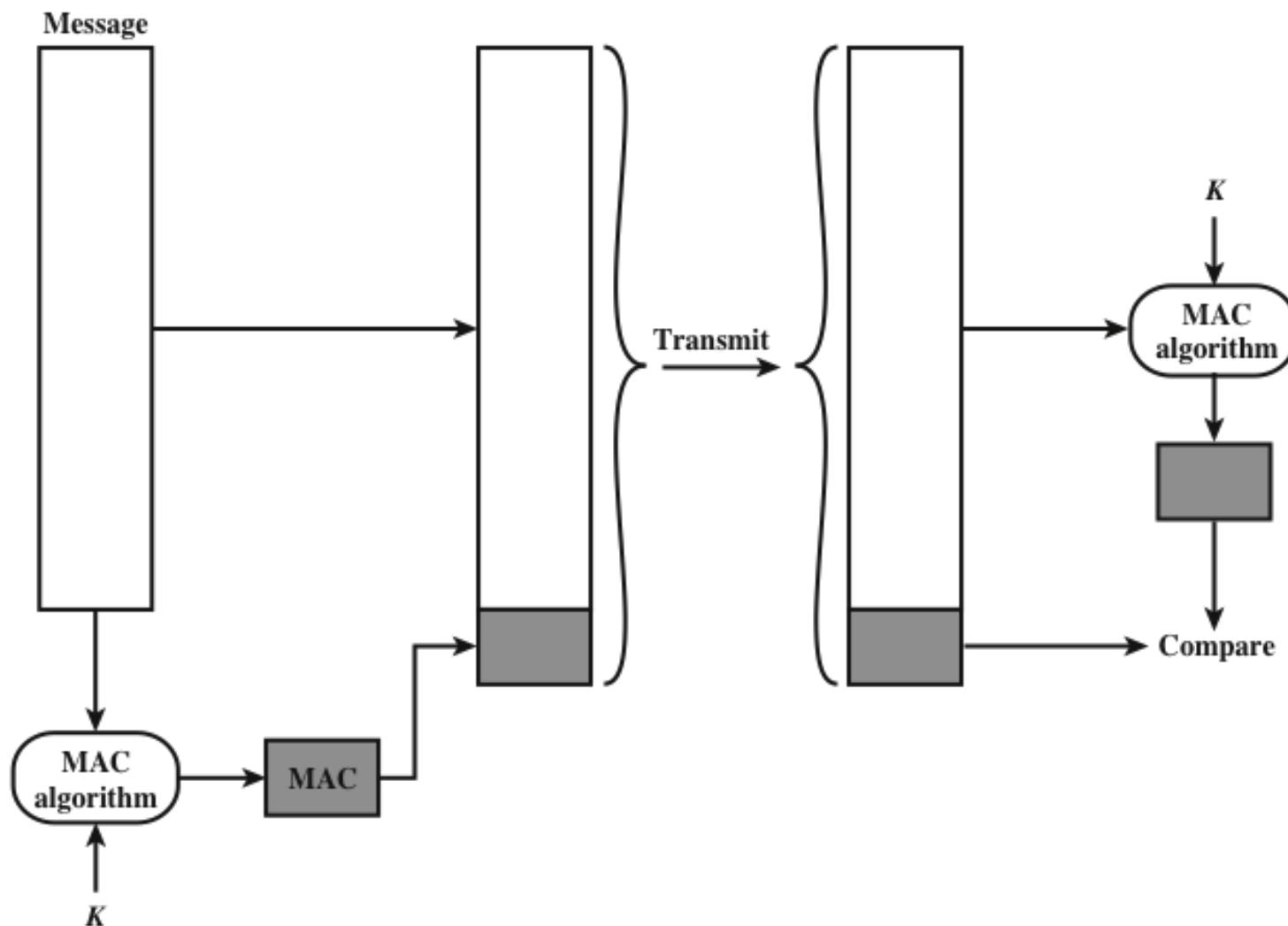
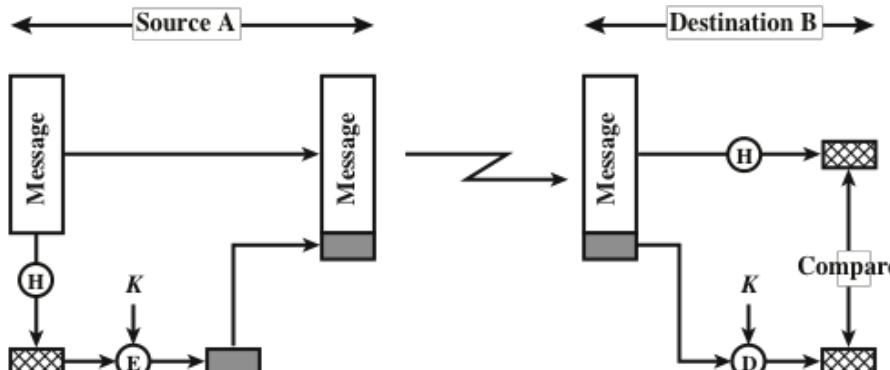


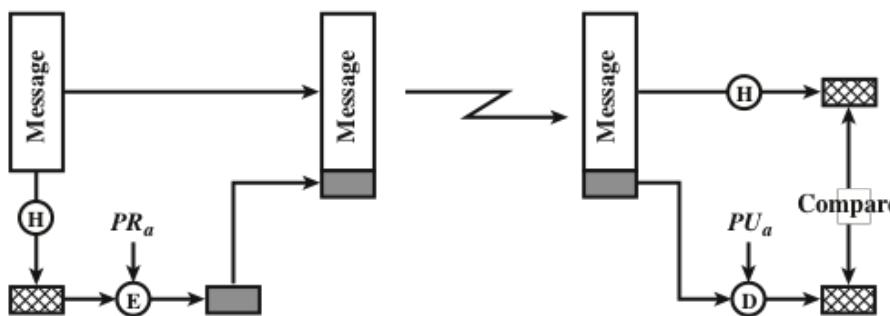
Figure 3.1 Message Authentication Using a Message Authentication Code (MAC)

# ONE-WAY HASH FUNCTIONS

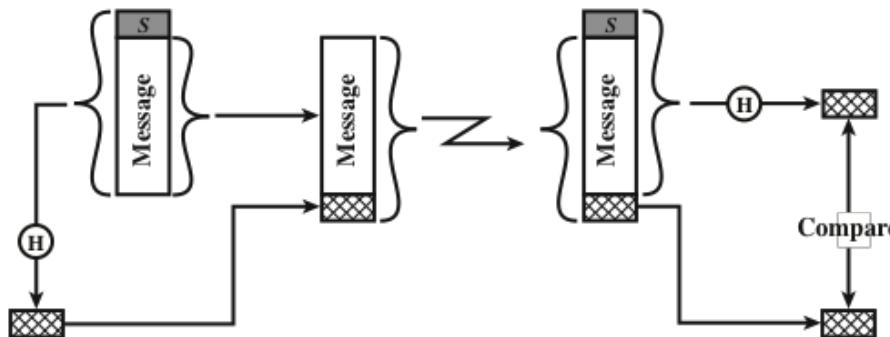
- Accepts a variable-size message  $M$  as input and produces a fixed-size message digest  $H(M)$  as output
- Does not take a secret key as input
- To authenticate a message, the message digest is sent with the message in such a way that the message digest is authentic



(a) Using conventional encryption



(b) Using public-key encryption



(c) Using secret value

Figure 3.2 Message Authentication Using a One-Way Hash Function

# SECURE HASH FUNCTIONS

- Is important not only in message authentication but in digital signatures
- Purpose is to produce a “fingerprint” of a file, message, or other block of data
- To be useful for message authentication, a hash function  $H$  must have the following properties:
  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 code  $h$ , it is computationally infeasible to find  $x$  such that  $H(x) = h$ . A hash function with this property is referred to as one-way or preimage resistant.
  5.
    - For any given block  $x$ , it is computationally infeasible to find  $y \neq x$  with  $H(y) = H(x)$ . A hash function with this property is referred to as second preimage resistant. This is sometimes referred to as weak collision resistant.
  6.
    - It is computationally infeasible to find any pair  $(x, y)$  such that  $H(x) = H(y)$ .
    - A hash function with this property is referred to as collision resistant. This is sometimes referred to as strong collision resistant.

# SECURITY OF HASH FUNCTIONS

- There are two approaches to attacking a secure hash function:
  - Cryptanalysis
    - Involves exploiting logical weaknesses in the algorithm
  - Brute-force attack
    - The strength of a hash function against this attack depends solely on the length of the hash code produced by the algorithm



	bit 1	bit 2	• • •	bit n
block 1	$b_{11}$	$b_{21}$		$b_{n1}$
block 2	$b_{12}$	$b_{22}$		$b_{n2}$
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
block m	$b_{1m}$	$b_{2m}$		$b_{nm}$
hash code	$C_1$	$C_2$		$C_n$

**Figure 3.3 Simple Hash Function Using Bitwise XOR**

# THE SHA SECURE HASH FUNCTION

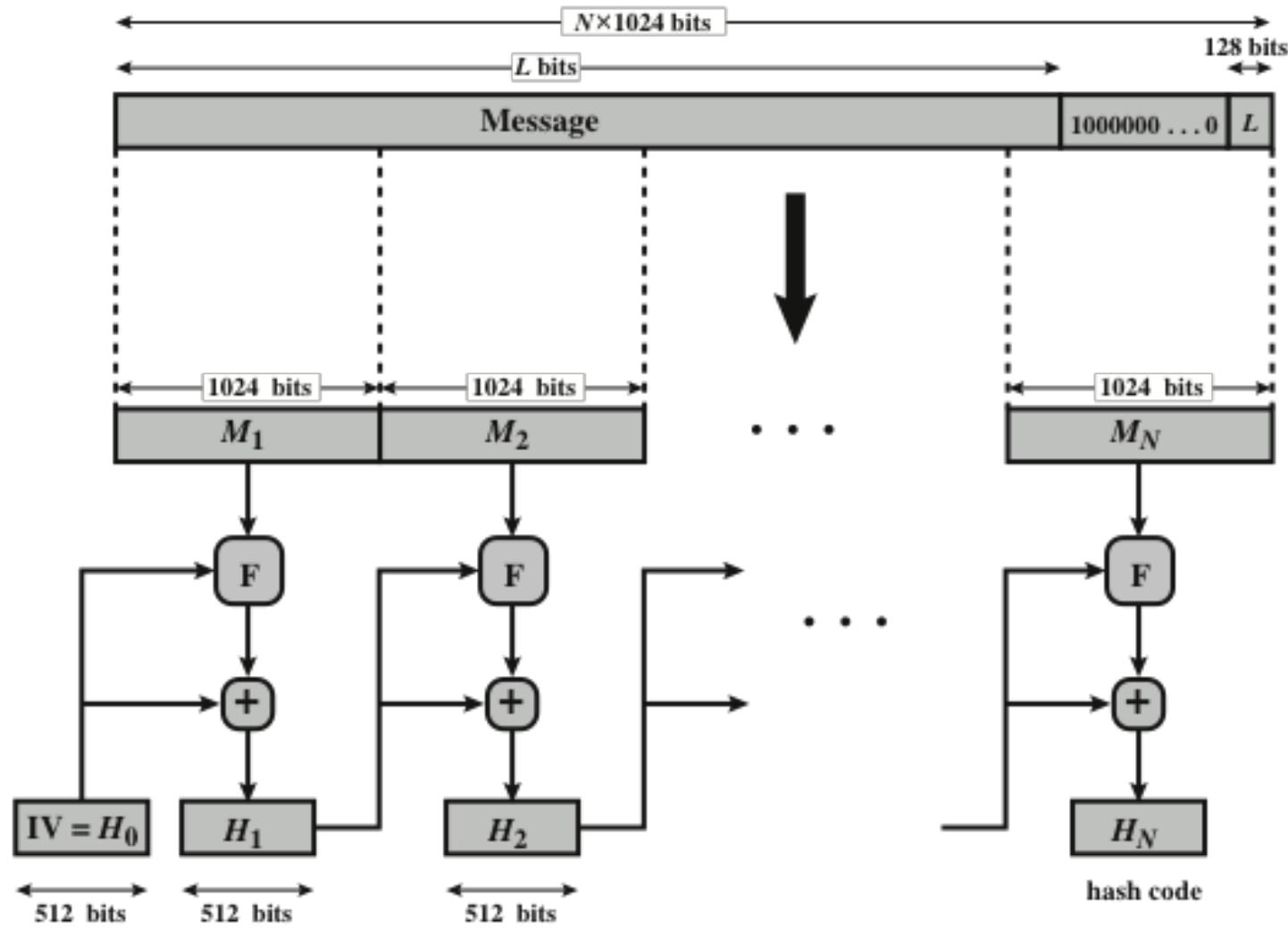
- SHA was developed by NIST and published as a federal information processing standard (FIPS 180) in 1993
- Was revised in 1995 as SHA-1 and published as FIPS 180-1
  - The actual standards document is entitled “Secure Hash Standard”
- Based on the hash function MD4 and its design closely models MD4
- Produces 160-bit hash values
- In 2005 NIST announced the intention to phase out approval of SHA-1 and move to a reliance on SHA-2 by 2010

# TABLE 3.1

## COMPARISON OF SHA PARAMETERS

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
<b>Message Digest Size</b>	160	224	256	384	512
<b>Message Size</b>	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
<b>Block Size</b>	512	512	512	1024	1024
<b>Word Size</b>	32	32	32	64	64
<b>Number of Steps</b>	80	64	64	80	80

Note: All sizes are measured in bits.



$+$  = word-by-word addition mod  $2^{64}$

**Figure 3.4 Message Digest Generation Using SHA-512**

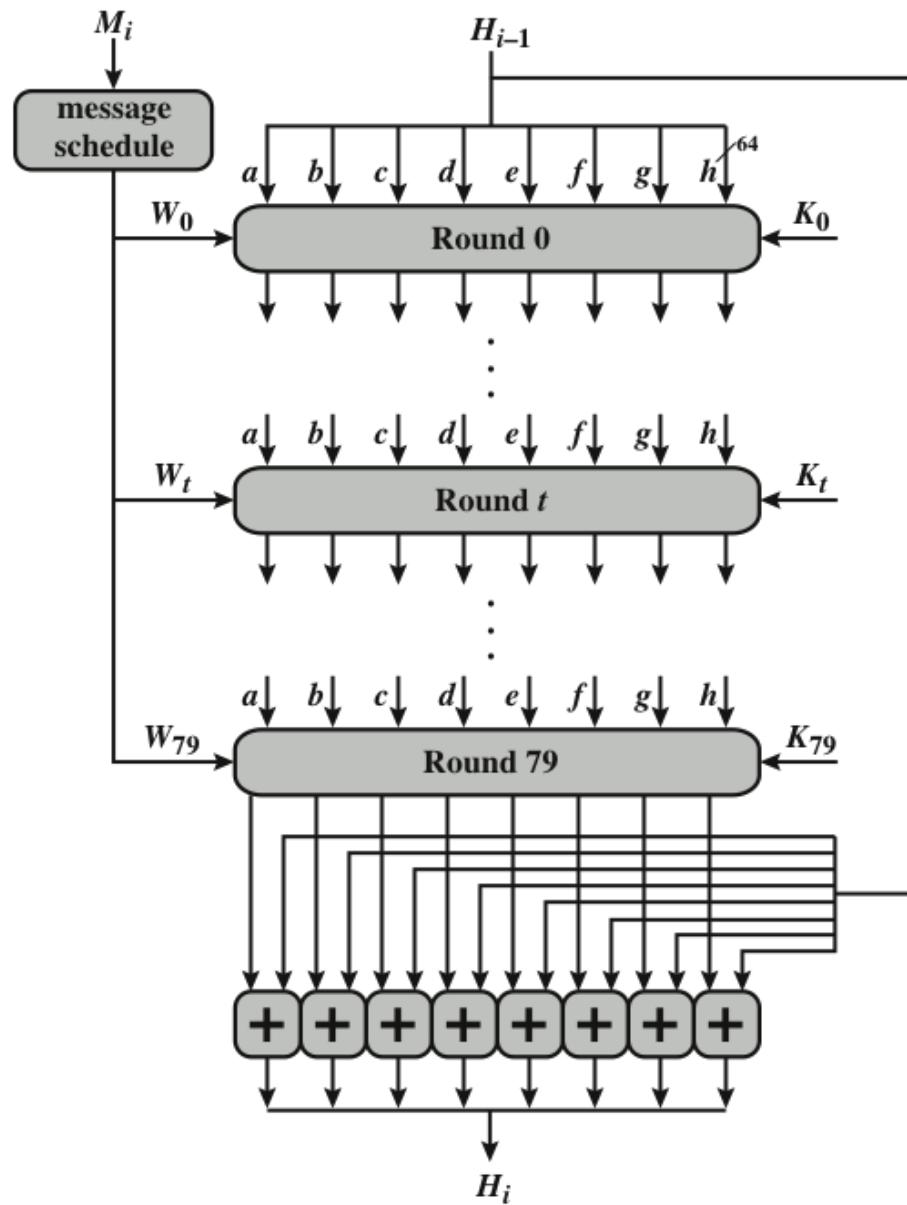
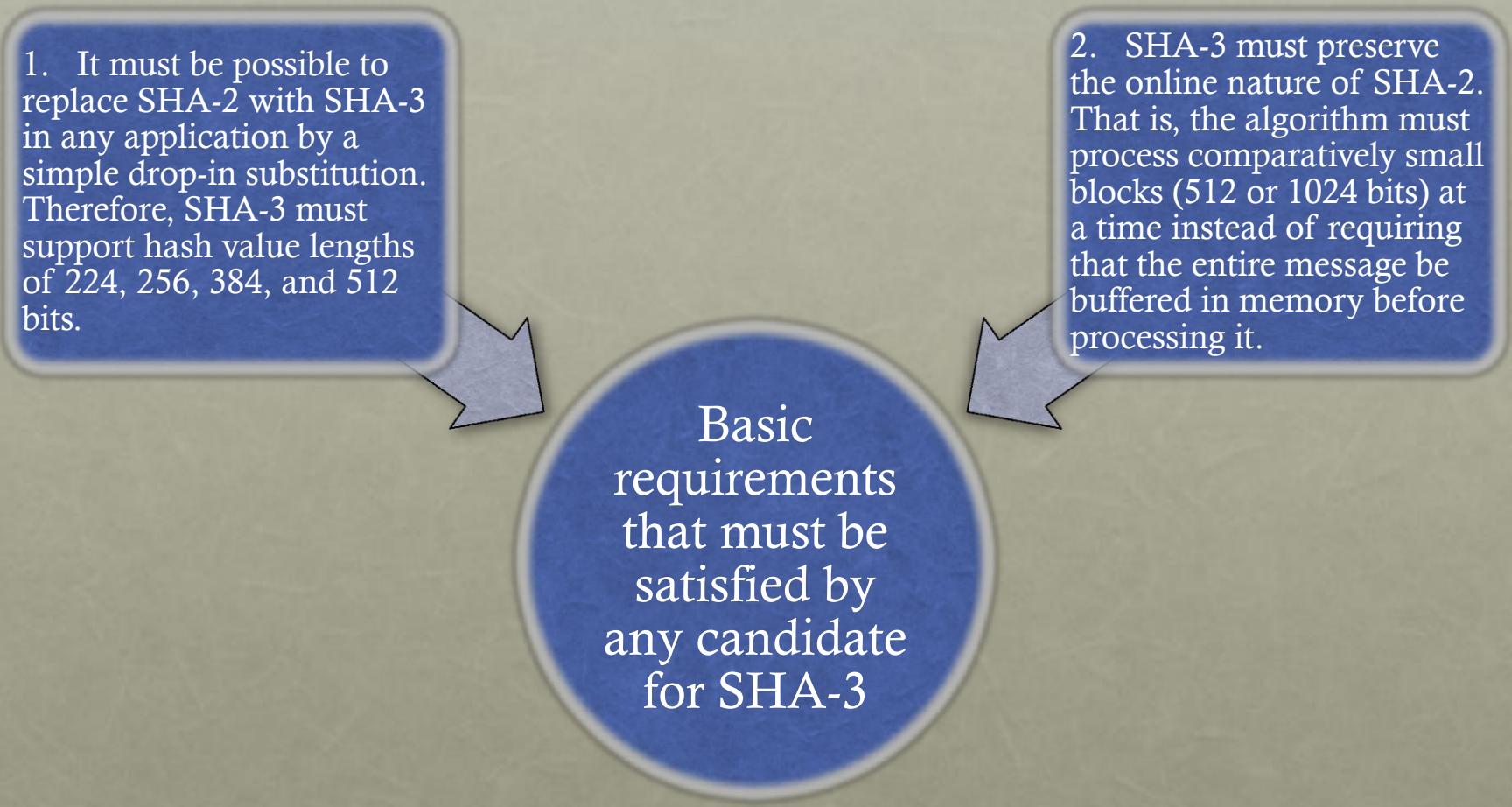


Figure 3.5 SHA-512 Processing of a Single 1024-Bit Block

# SHA-3

- 
1. It must be possible to replace SHA-2 with SHA-3 in any application by a simple drop-in substitution. Therefore, SHA-3 must support hash value lengths of 224, 256, 384, and 512 bits.
2. SHA-3 must preserve the online nature of SHA-2. That is, the algorithm must process comparatively small blocks (512 or 1024 bits) at a time instead of requiring that the entire message be buffered in memory before processing it.
- Basic requirements that must be satisfied by any candidate for SHA-3

# HMAC

- There has been an increased interest in **developing** a MAC derived from a cryptographic hash code, such as SHA-1
  - Cryptographic hash functions generally execute faster in software than conventional encryption algorithms such as DES
  - Library code for cryptographic hash functions is widely available
  - A hash function such as SHA-1 was not designed for use as a MAC and cannot be used directly for that purpose because it does not rely on a secret key
- There have been a number of proposals for the incorporation of a secret key into an existing hash algorithm
  - The approach that has received the most support is HMAC

## HMAC

- Has been issued as RFC 2104
- Has been chosen as the mandatory-to-implement MAC for IP Security
- Is used in other Internet protocols, such as Transport Layer Security (TLS) and Secure Electronic Transaction (SET)

# HMAC DESIGN OBJECTIVES

- To use, without modifications, available hash functions --- in particular, hash functions that perform well in software, and for which code is freely and widely available
- To allow for easy replaceability of the embedded hash function in case faster or more secure hash functions are found or required
- To preserve the original performance of the hash function without incurring a significant degradation
- To use and handle keys in a simple way
- To have a well understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions on the embedded hash function

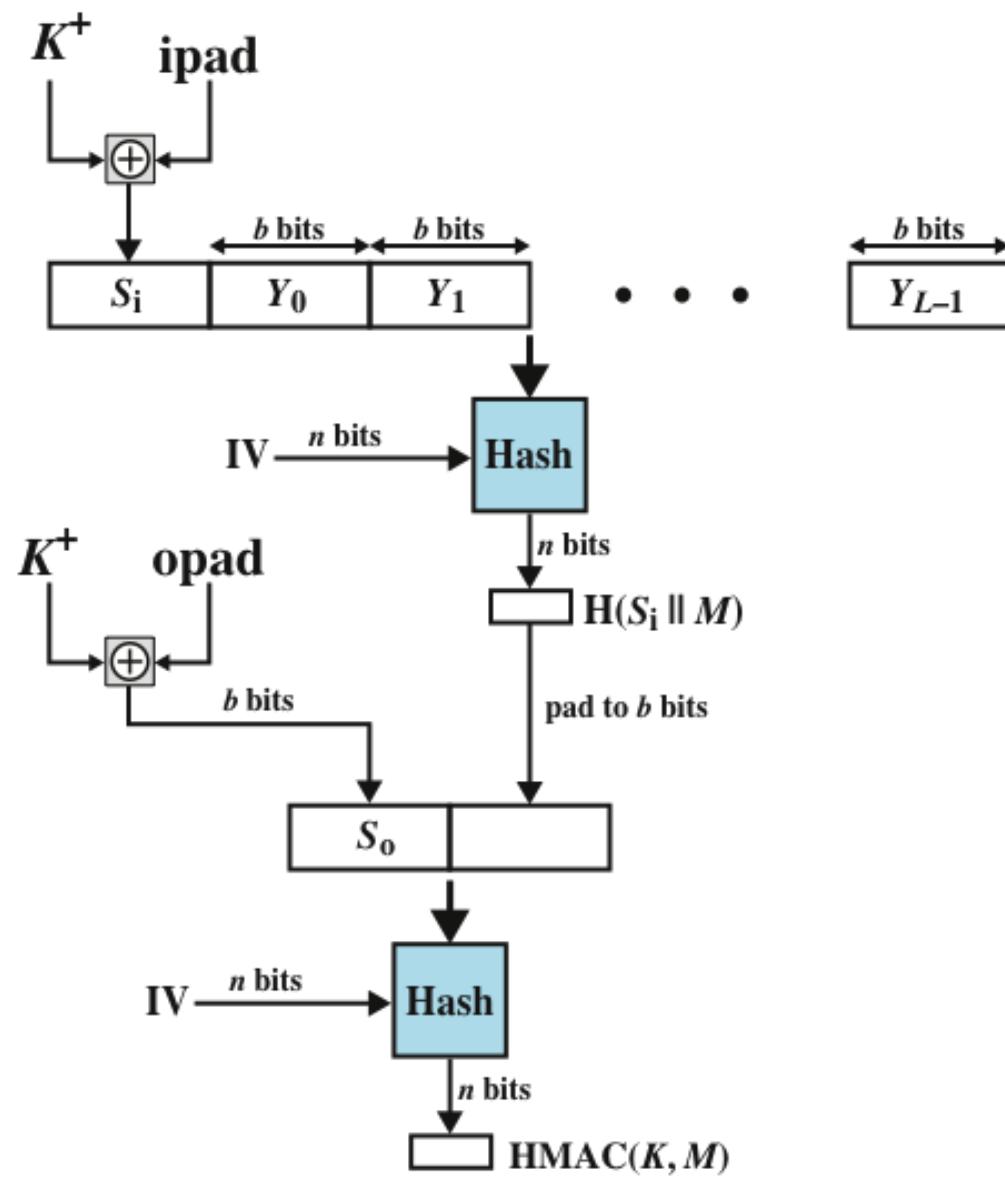
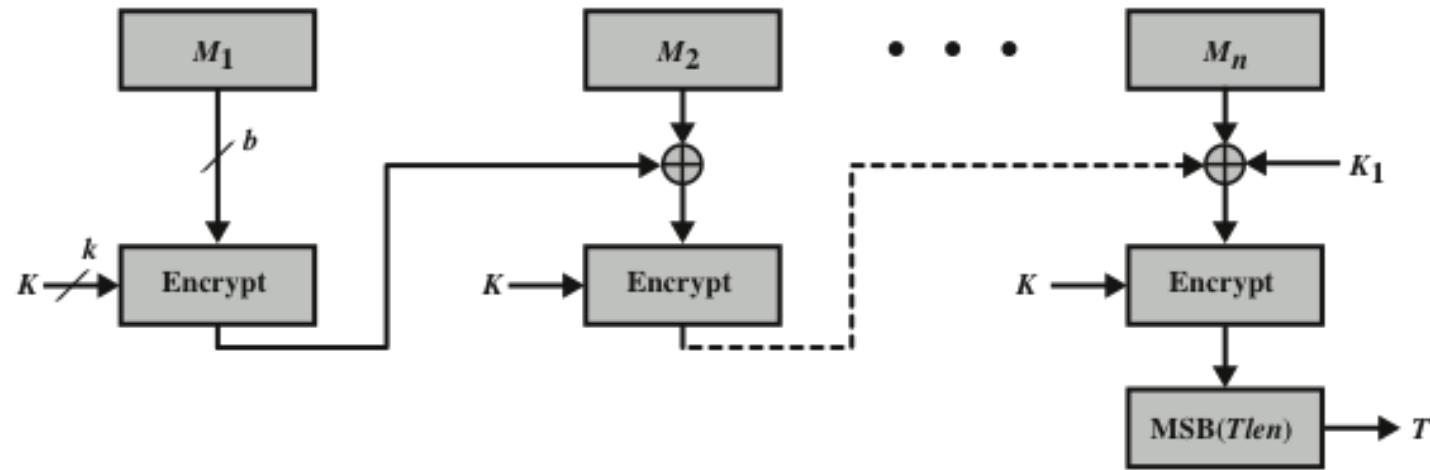
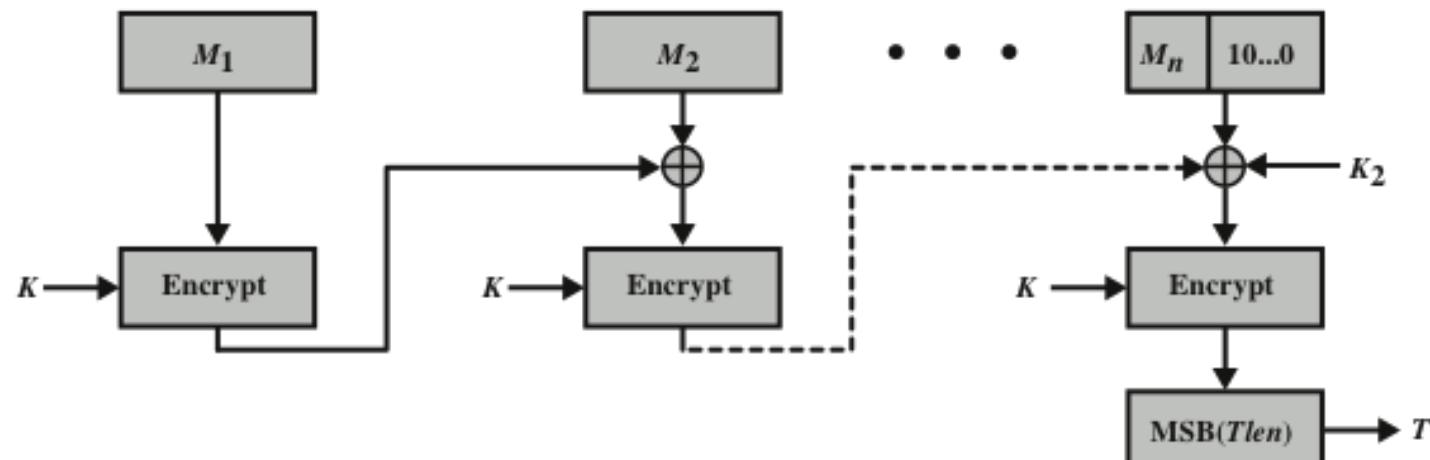


Figure 3.6 HMAC Structure



(a) Message length is integer multiple of block size

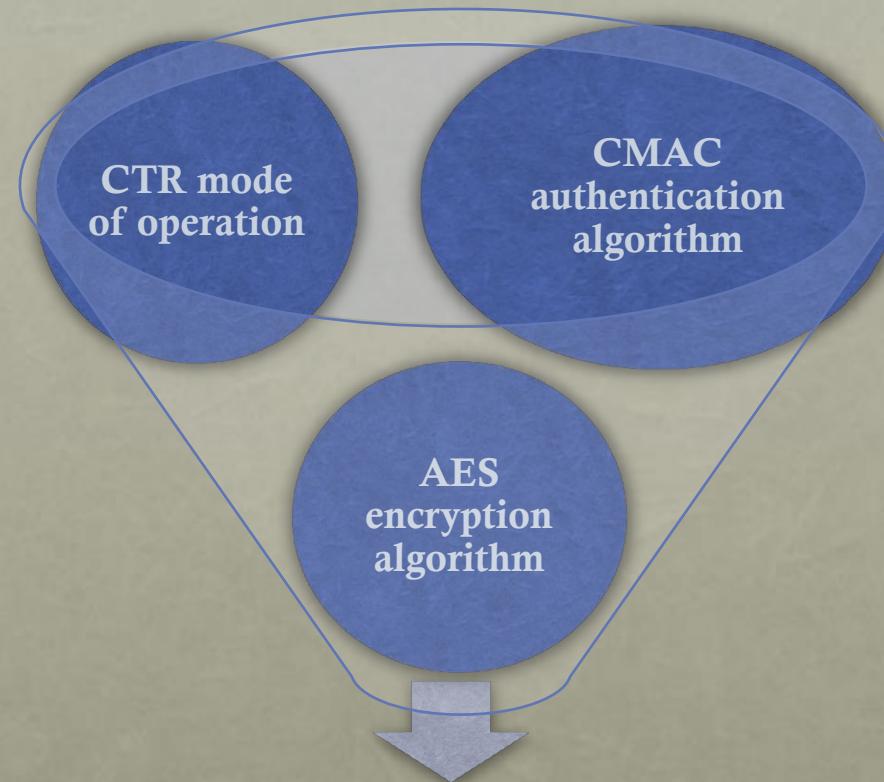


(b) Message length is not integer multiple of block size

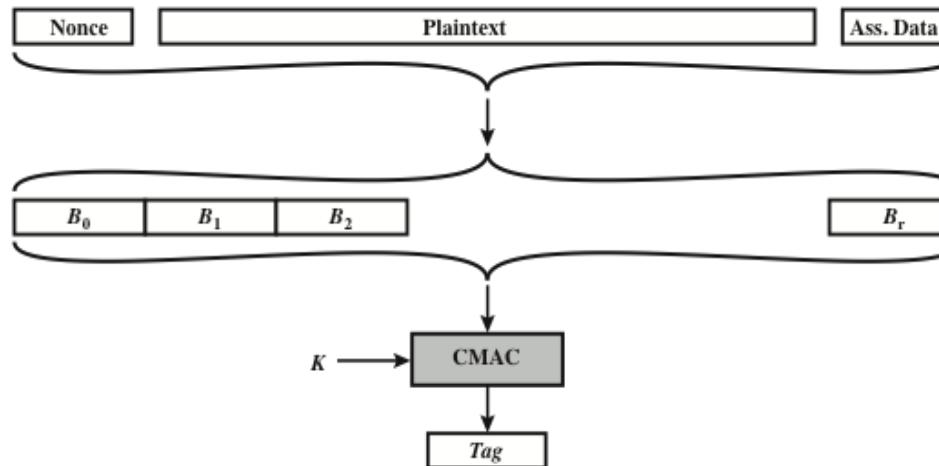
Figure 3.7 Cipher-Based Message Authentication Code (CMAC)

# COUNTER WITH CIPHER BLOCK CHAINING-MESSAGE AUTHENTICATION CODE (CCM)

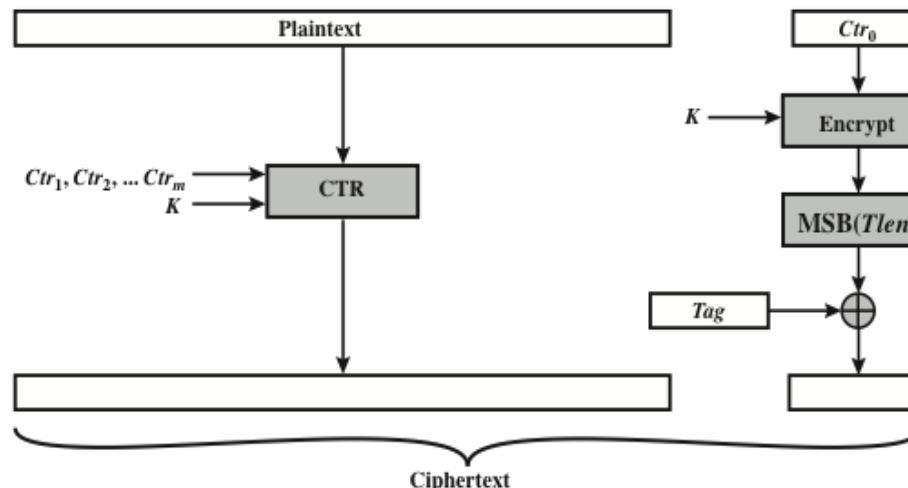
- NIST standard SP 800-38C
- Referred to as an *authenticated encryption* mode
  - “Authenticated encryption” is a term used to describe encryption systems that simultaneously protect confidentiality and authenticity of communications
- A single key is used for both encryption and MAC algorithms



Key algorithmic ingredients



(a) Authentication



(b) Encryption

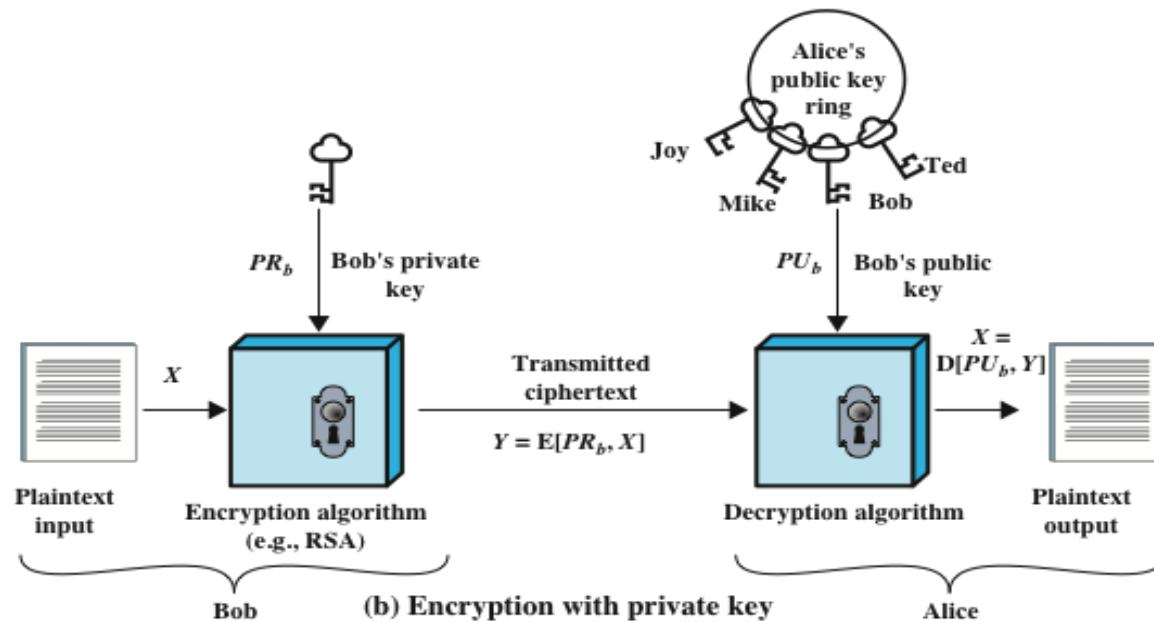
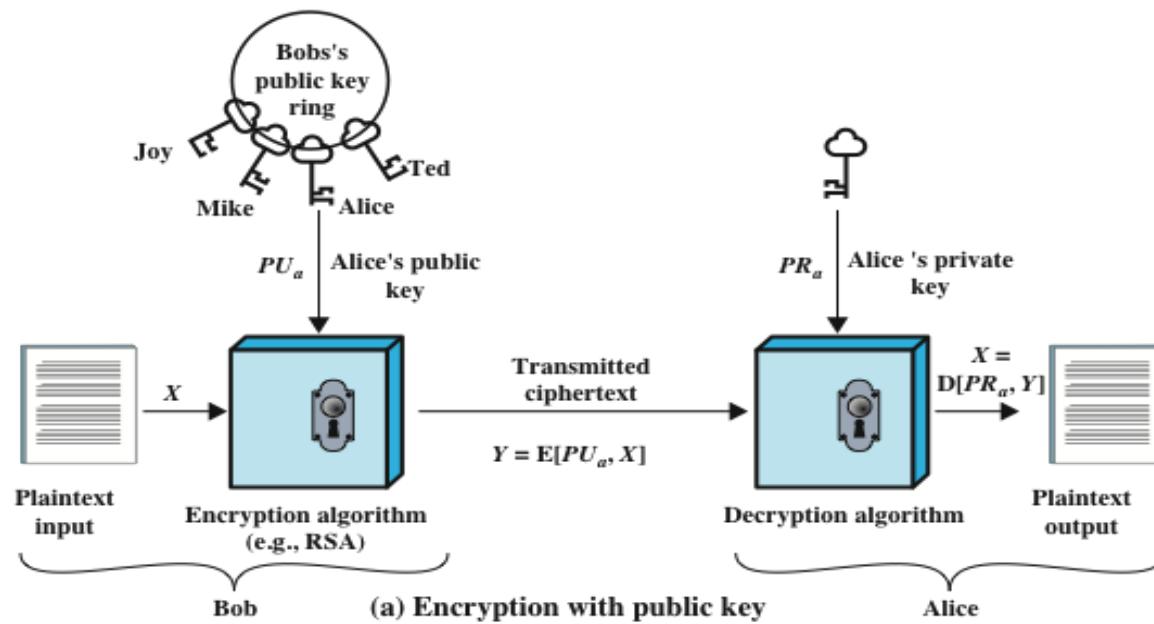
Figure 3.8 Counter with Cipher Block Chaining-Message Authentication Code (CCM)

# PUBLIC-KEY ENCRYPTION STRUCTURE

- First publicly proposed by Diffie and Hellman in 1976
- Based on mathematical functions rather than on simple operations on bit patterns
- Is asymmetric, involving the use of two separate keys

## Misconceptions:

- Public-key encryption is more secure from cryptanalysis than conventional encryption
- Public-key encryption is a general-purpose technique that has made conventional encryption obsolete
- There is a feeling that key distribution is trivial when using public-key encryption, compared to the rather cumbersome handshaking involved with key distribution centers for conventional encryption



**Figure 3.9 Public-Key Cryptography**