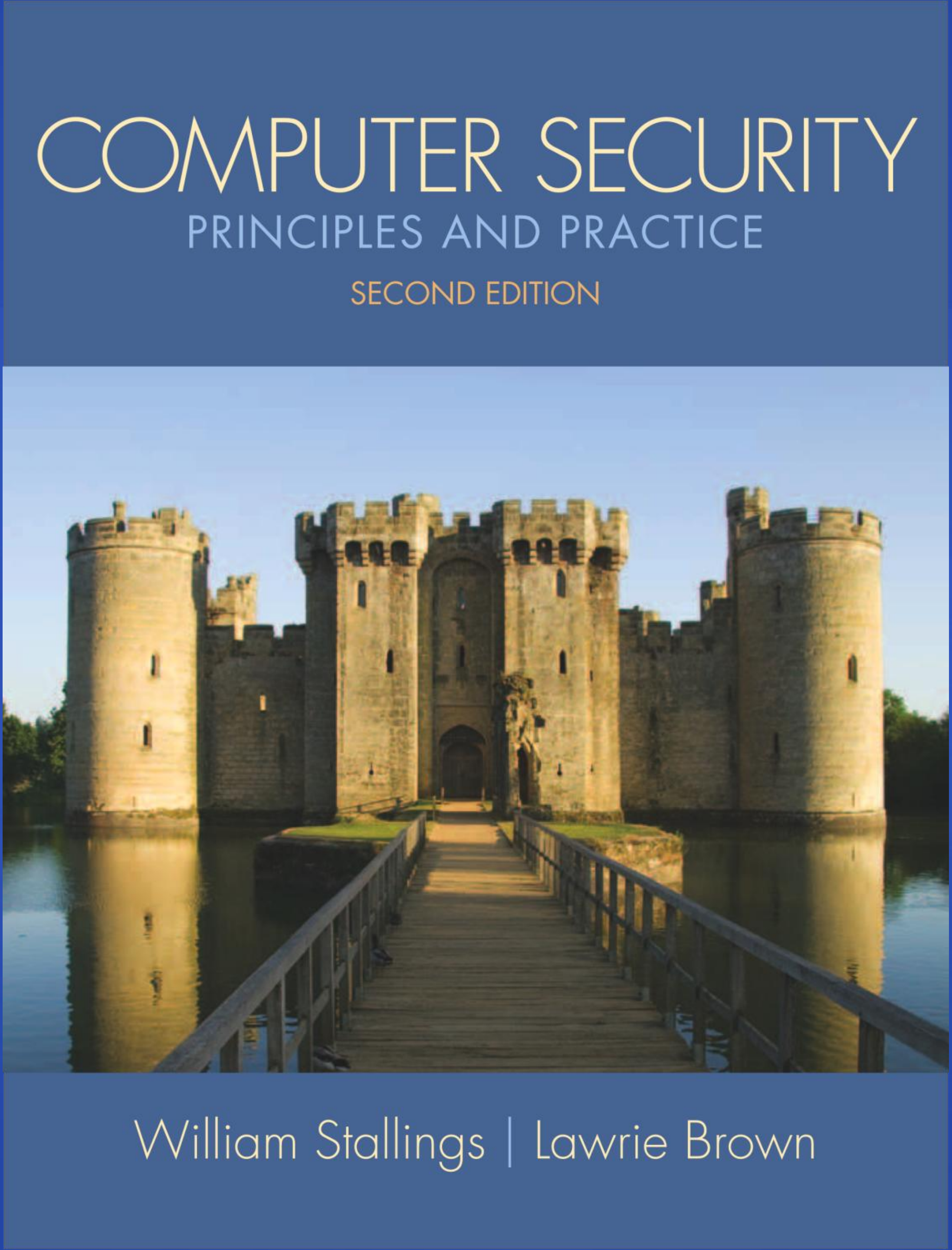


# COMPUTER SECURITY

## PRINCIPLES AND PRACTICE

SECOND EDITION

William Stallings | Lawrie Brown

# Chapter 2

## Cryptographic Tools

# Symmetric Encryption

- the universal technique for providing confidentiality for transmitted or stored data
- also referred to as conventional encryption or single-key encryption
- two requirements for secure use:
  - need a strong encryption algorithm
  - sender and receiver must have obtained copies of the secret key in a secure fashion and must keep the key secure

# Figure 2.1

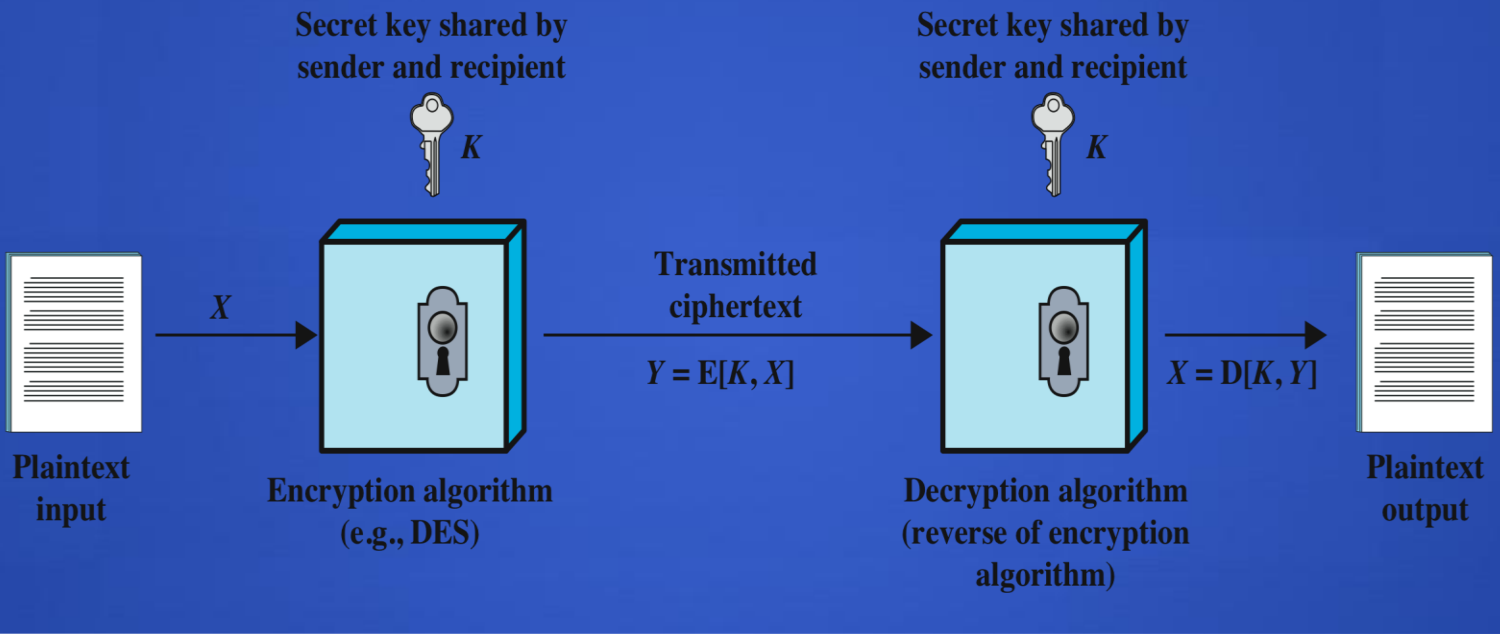

Secret key shared by sender and recipient
K
Secret key shared by sender and recipient
K
X
Transmitted ciphertext
Y = E[K, X]
X = D[K, Y]
Plaintext input
Encryption algorithm (e.g., DES)
Decryption algorithm (reverse of encryption algorithm)
Plaintext output

Figure 2.1 Simplified Model of Symmetric Encryption

# Attacking Symmetric Encryption

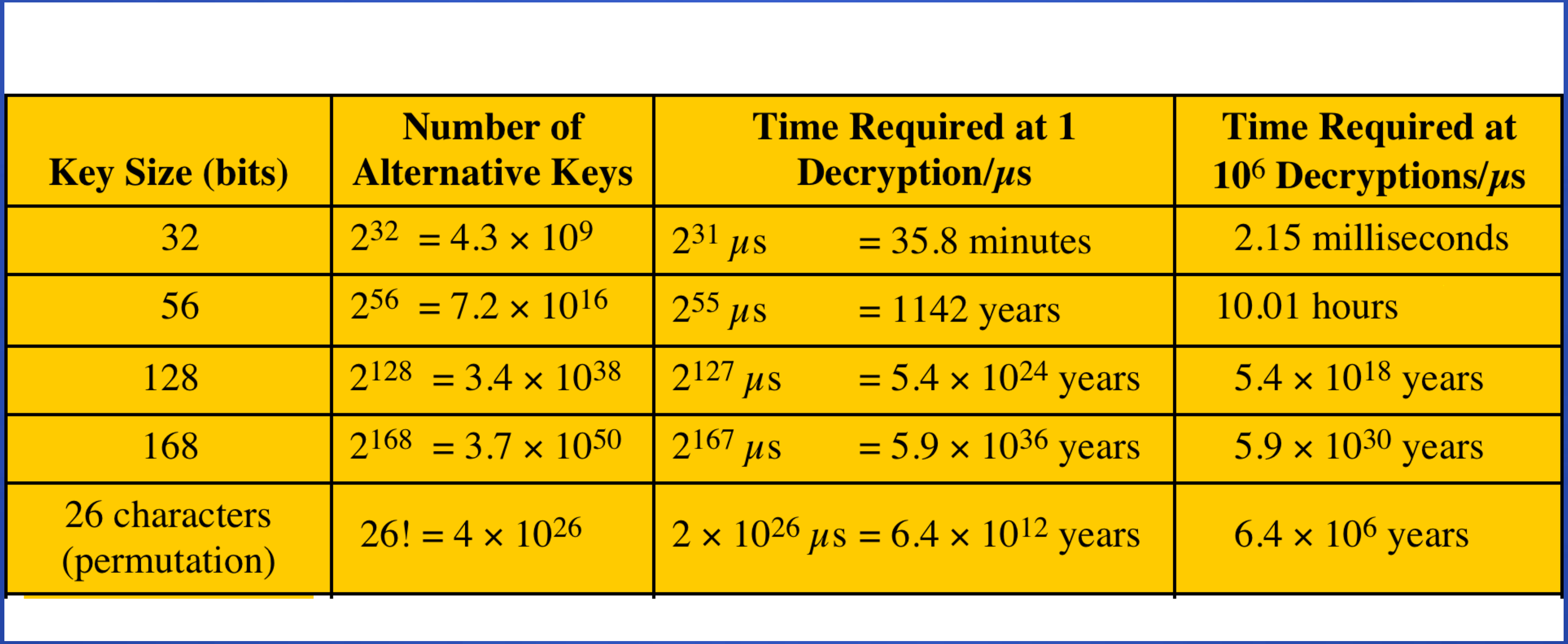
## Cryptanalytic Attacks

- rely on:
  - nature of the algorithm
  - some knowledge of the general characteristics of the plaintext
  - some sample plaintext-ciphertext pairs
- exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or the key being used
  - if successful all future and past messages encrypted with that key are compromised

## Brute-Force Attack

- try all possible keys on some ciphertext until an intelligible translation into plaintext is obtained
  - on average half of all possible keys must be tried to achieve success

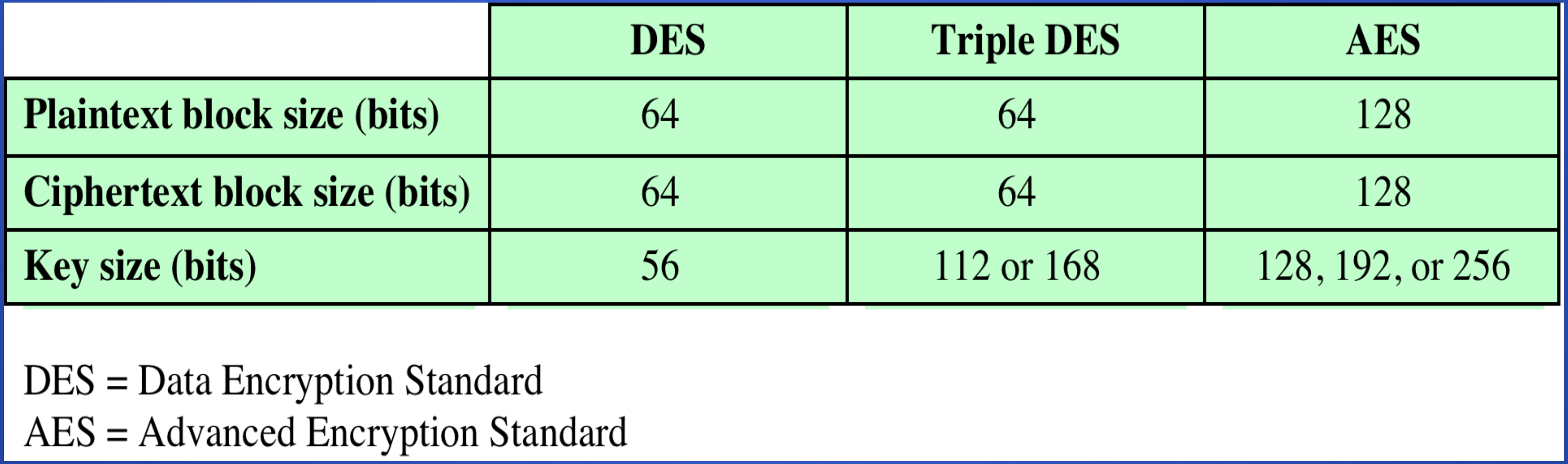
# Table 2.1



| Key Size (bits) | Number of Alternative Keys | Time Required at 1 Decryption/$\mu s$ | Time Required at $10^6$ Decryptions/$\mu s$ |
|---|---|---|---|
| 32 | $2^{32} = 4.3 \times 10^9$ | $2^{31}\ \mu s = 35.8$ minutes | 2.15 milliseconds |
| 56 | $2^{56} = 7.2 \times 10^{16}$ | $2^{55}\ \mu s = 1142$ years | 10.01 hours |
| 128 | $2^{128} = 3.4 \times 10^{38}$ | $2^{127}\ \mu s = 5.4 \times 10^{24}$ years | $5.4 \times 10^{18}$ years |
| 168 | $2^{168} = 3.7 \times 10^{50}$ | $2^{167}\ \mu s = 5.9 \times 10^{36}$ years | $5.9 \times 10^{30}$ years |
| 26 characters (permutation) | $26! = 4 \times 10^{26}$ | $2 \times 10^{26}\ \mu s = 6.4 \times 10^{12}$ years | $6.4 \times 10^6$ years |

## Average Time Required for Exhaustive Key Search

# Table 2.2



| | DES | Triple DES | AES |
|---|---|---|---|
| **Plaintext block size (bits)** | 64 | 64 | 128 |
| **Ciphertext block size (bits)** | 64 | 64 | 128 |
| **Key size (bits)** | 56 | 112 or 168 | 128, 192, or 256 |

DES = Data Encryption Standard
AES = Advanced Encryption Standard

## Comparison of Three Popular Symmetric Encryption Algorithms

# Data Encryption Standard (DES)

- the most widely used encryption scheme
  - FIPS PUB 46
  - referred to as the Data Encryption Algorithm (DEA)
  - uses 64 bit plaintext block and 56 bit key to produce a 64 bit ciphertext block

- strength concerns:
  - concerns about algorithm
    - DES is the most studied encryption algorithm in existence
  - use of 56-bit key
    - Electronic Frontier Foundation (EFF) announced in July 1998 that it had broken a DES encryption

# Figure 2.2

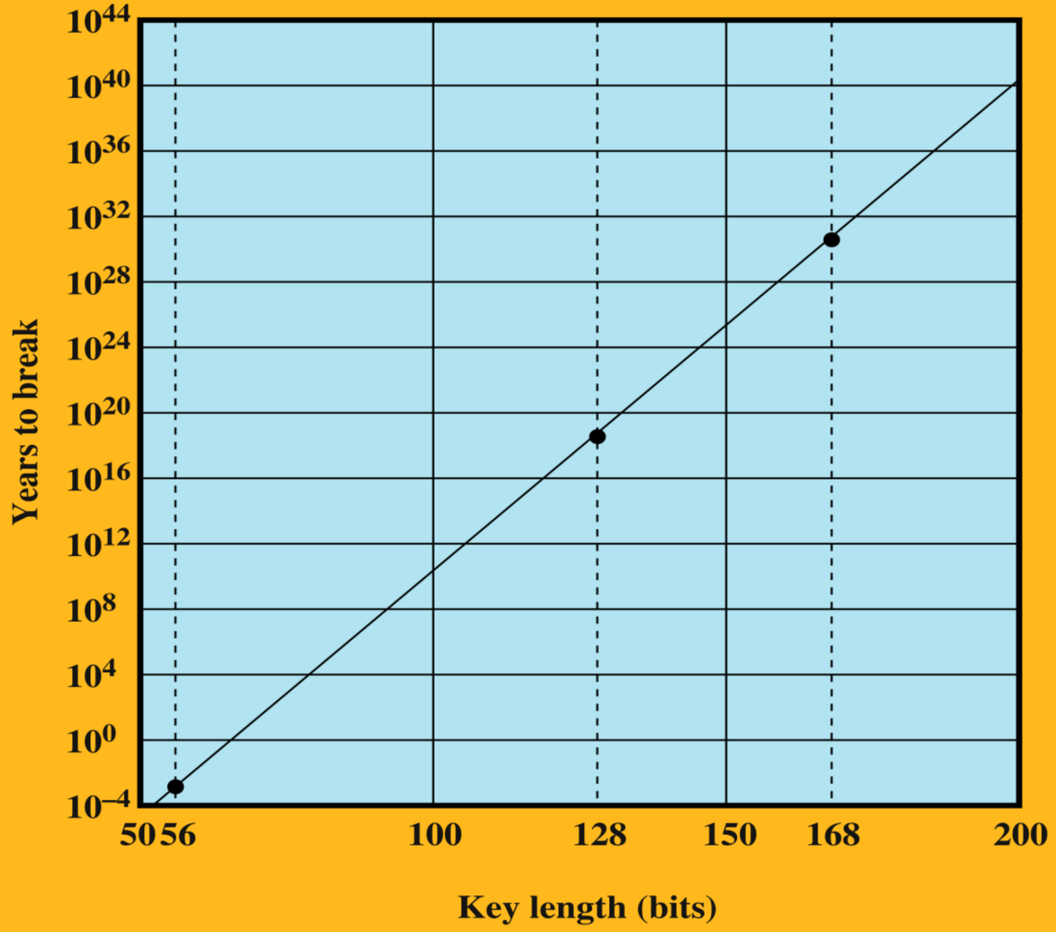


**Figure 2.2 Time to Break a Code (assuming 106 decryptions/ms)** The graph assumes that a symmetric encryption algorithm is attacked using a brute-force approach of trying all possible keys

# Triple DES (3DES)

- repeats basic DES algorithm three times using either two or three unique keys
- first standardized for use in financial applications in ANSI standard X9.17 in 1985
- attractions:
  - 168-bit key length overcomes the vulnerability to brute-force attack of DES
  - underlying encryption algorithm is the same as in DES
- drawbacks:
  - algorithm is sluggish in software
  - uses a 64-bit block size

# Advanced Encryption Standard (AES)

## needed a replacement for 3DES

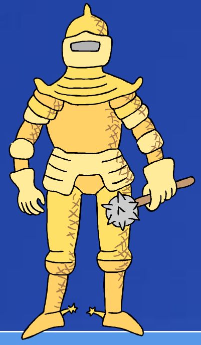
3DES was not reasonable for long term use

## NIST called for proposals for a new AES in 1997

- should have a security strength equal to or better than 3DES
- significantly improved efficiency
- symmetric block cipher
- 128 bit data and 128/192/256 bit keys

## selected Rijndael in November 2001

published as FIPS 197

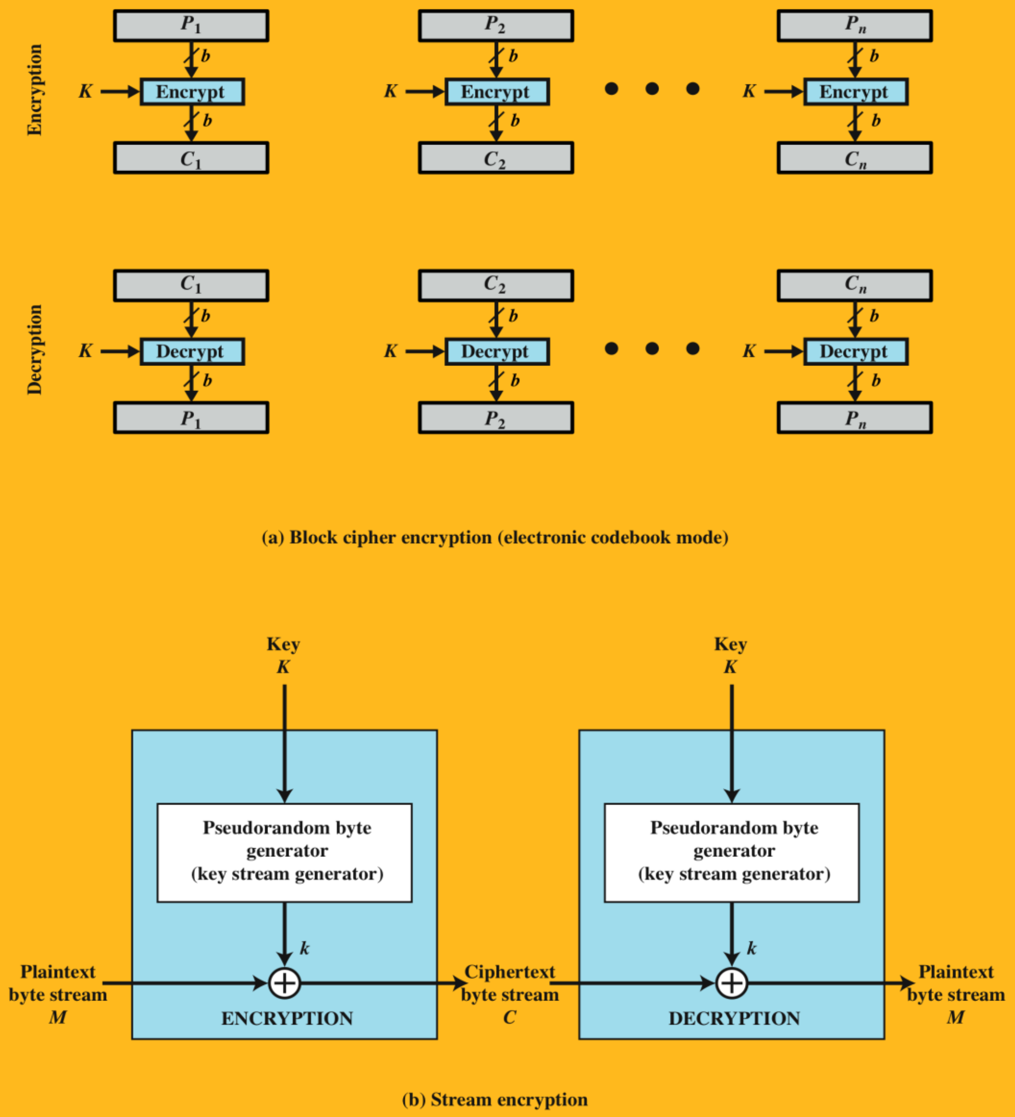


# Practical Security Issues

- typically symmetric encryption is applied to a unit of data larger than a single 64-bit or 128-bit block
- electronic codebook (ECB) mode is the simplest approach to multiple-block encryption
  - each block of plaintext is encrypted using the same key
  - cryptanalysts may be able to exploit regularities in the plaintext
- modes of operation
  - alternative techniques developed to increase the security of symmetric block encryption for large sequences
  - overcomes the weaknesses of ECB

# Block Cipher Encryption

# Stream Encryption



(a) Block cipher encryption (electronic codebook mode)

(b) Stream encryption

**Figure 2.3 Types of Symmetric Encryption**

# Block & Stream Ciphers

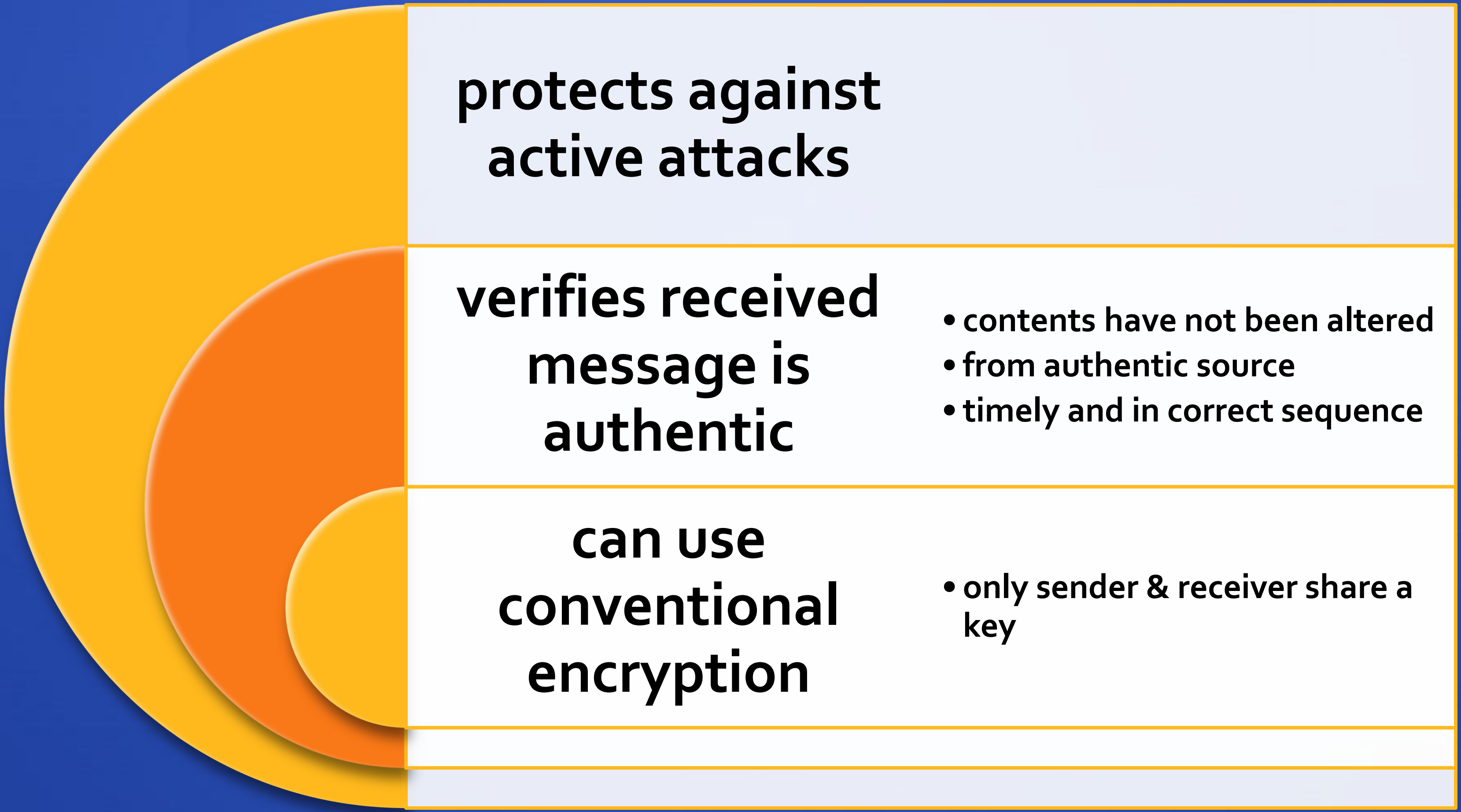
## Block Cipher

- processes the input one block of elements at a time
- produces an output block for each input block
- can reuse keys
- more common

## Stream Cipher

- processes the input elements continuously
- produces output one element at a time
- primary advantage is that they are almost always faster and use far less code
- encrypts plaintext one byte at a time
- pseudorandom stream is one that is unpredictable without knowledge of the input key

# Message Authentication



protects against active attacks

verifies received message is authentic

- contents have not been altered
- from authentic source
- timely and in correct sequence

can use conventional encryption

- only sender & receiver share a key

# Message Authentication Codes

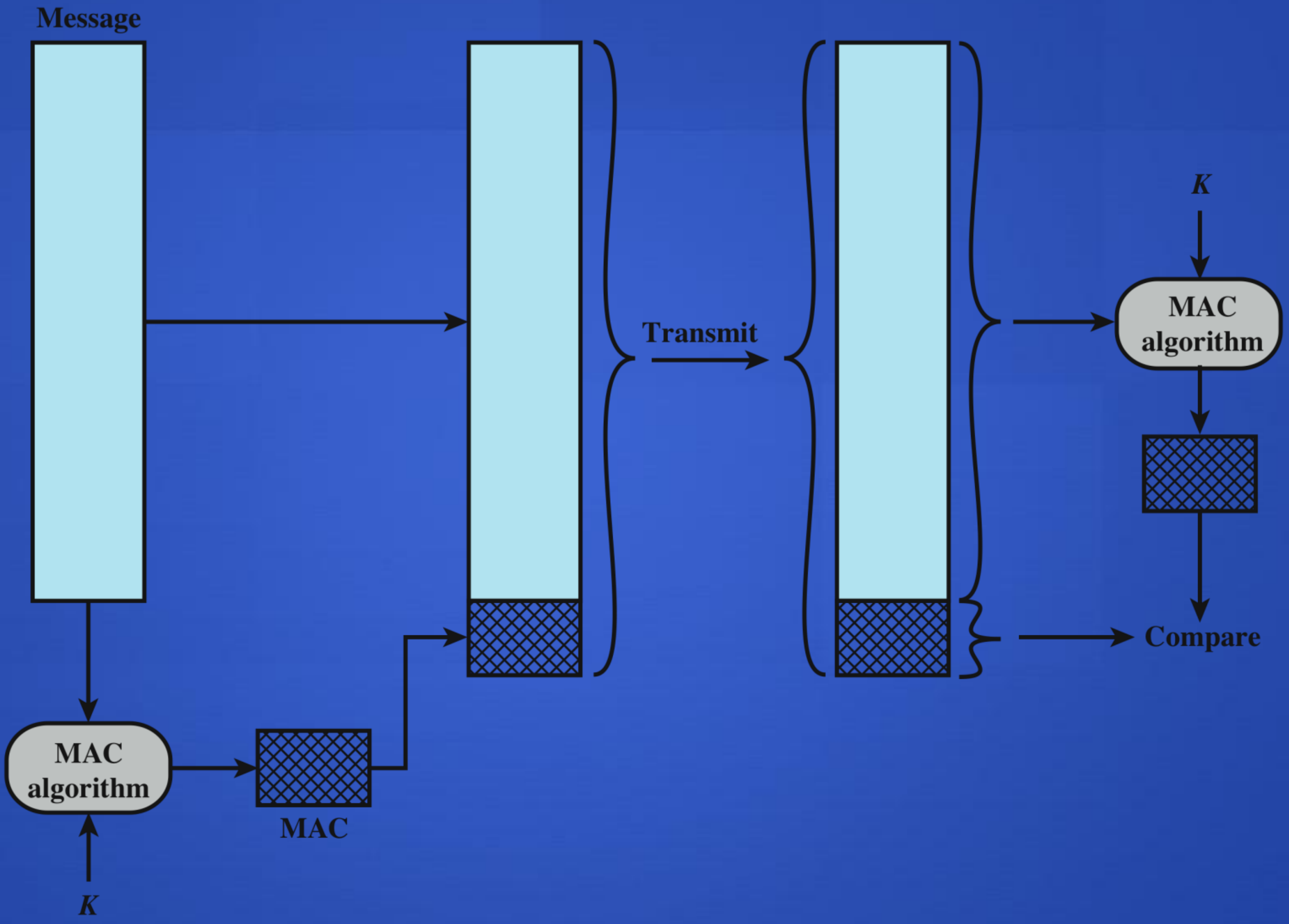


Figure 2.4 Message Authentication Using a Message Authentication Code (MAC). The MAC is a function of an input message and a secret key.

# Secure Hash Functions

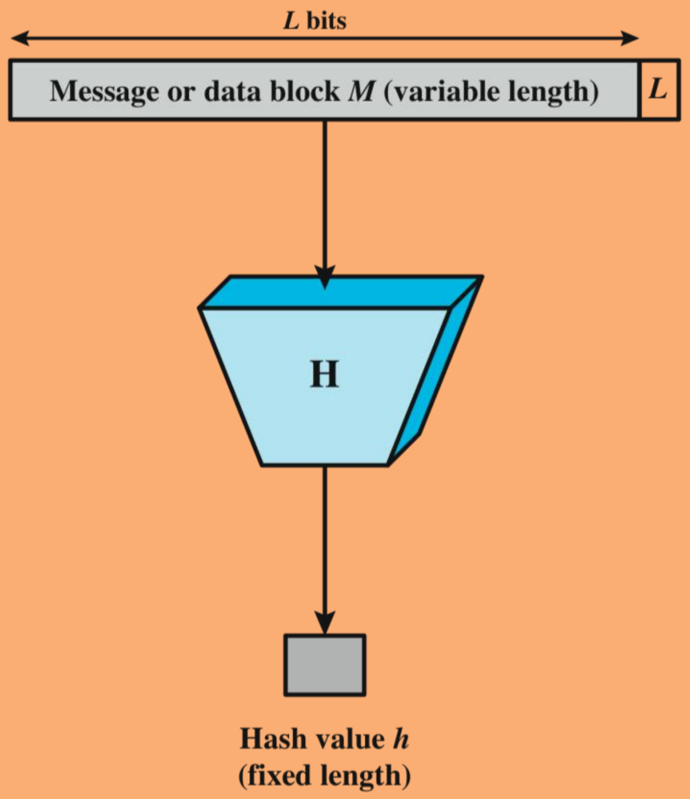


Figure 2.5 Block Diagram of Secure Hash Function; $h = H(M)$

# Figure 2.6

## Message Authentication Using a One-Way Hash Function

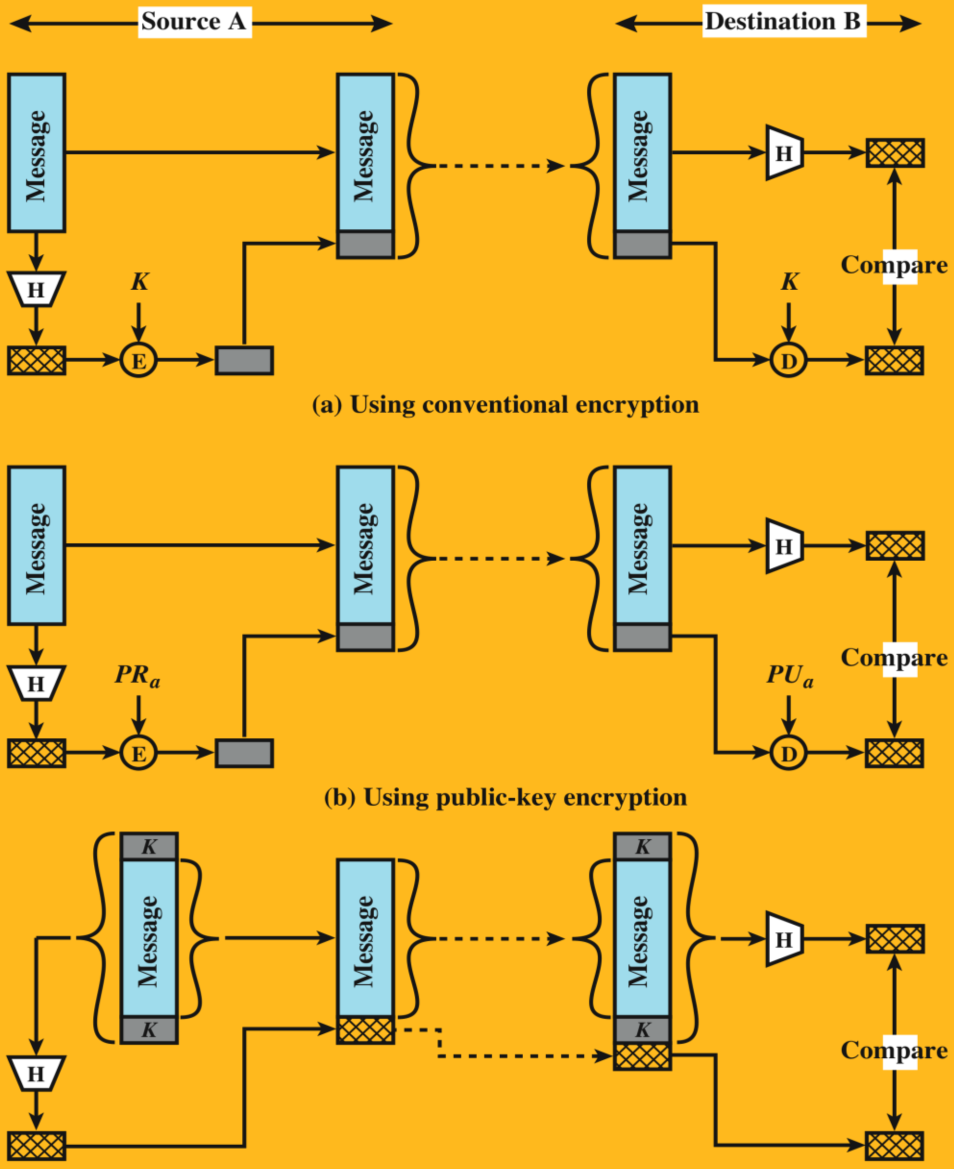


Figure 2.6 Message Authentication Using a One-Way Hash Function. The hash function maps a message into a relatively small, fixed-size block.

# Hash Function Requirements

- can be applied to a block of data of any size
- produces a fixed-length output
- H(*x*) is relatively easy to compute for any given *x*
- one-way or pre-image resistant
  - computationally infeasible to find *x* such that H(*x*) = *h*
- second pre-image resistant or weak collision resistant
  - computationally infeasible to find *y* ≠ *x* such that H(*y*) = H(*x*)
- collision resistant or strong collision resistance
  - computationally infeasible to find any pair (*x*, *y*) such that H(*x*) = H(*y*)

# Security of Hash Functions

- there are two approaches to attacking a secure hash function:
  - cryptanalysis
    - exploit logical weaknesses in the algorithm
  - brute-force attack
    - strength of hash function depends solely on the length of the hash code produced by the algorithm
- SHA most widely used hash algorithm
- additional secure hash function applications:
  - passwords
    - hash of a password is stored by an operating system
  - intrusion detection
    - store H(F) for each file on a system and secure the hash values

# Public-Key Encryption Structure

publicly proposed by Diffie and Hellman in 1976

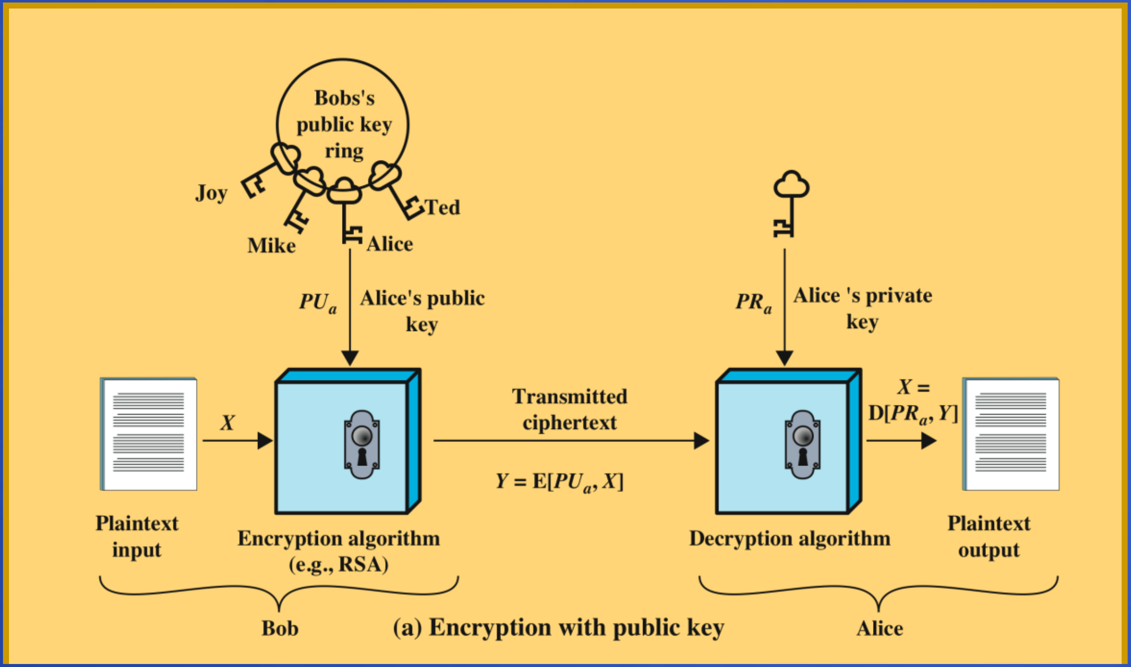
based on mathematical functions

asymmetric

- uses two separate keys
- public key and private key
- public key is made public for others to use

some form of protocol is needed for distribution

# Figure 2.7a
# Public-Key Encryption

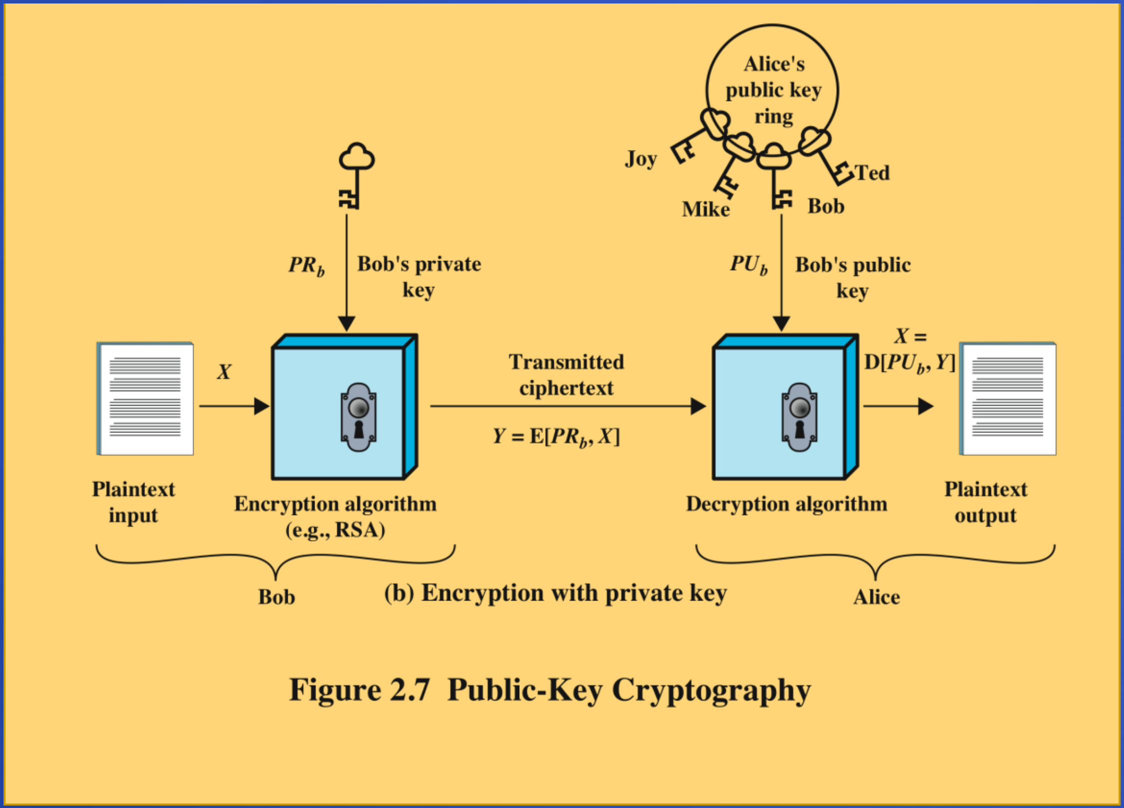


(a) Encryption with public key

***directed toward providing confidentiality

- plaintext
  - readable message or data that is fed into the algorithm as input
- encryption algorithm
  - performs transformations on the plaintext
- public and private key
  - pair of keys, one for encryption, one for decryption
- ciphertext
  - scrambled message produced as output
- decryption key
  - produces the original plaintext

# Figure 2.7b
# Private-Key Encryption



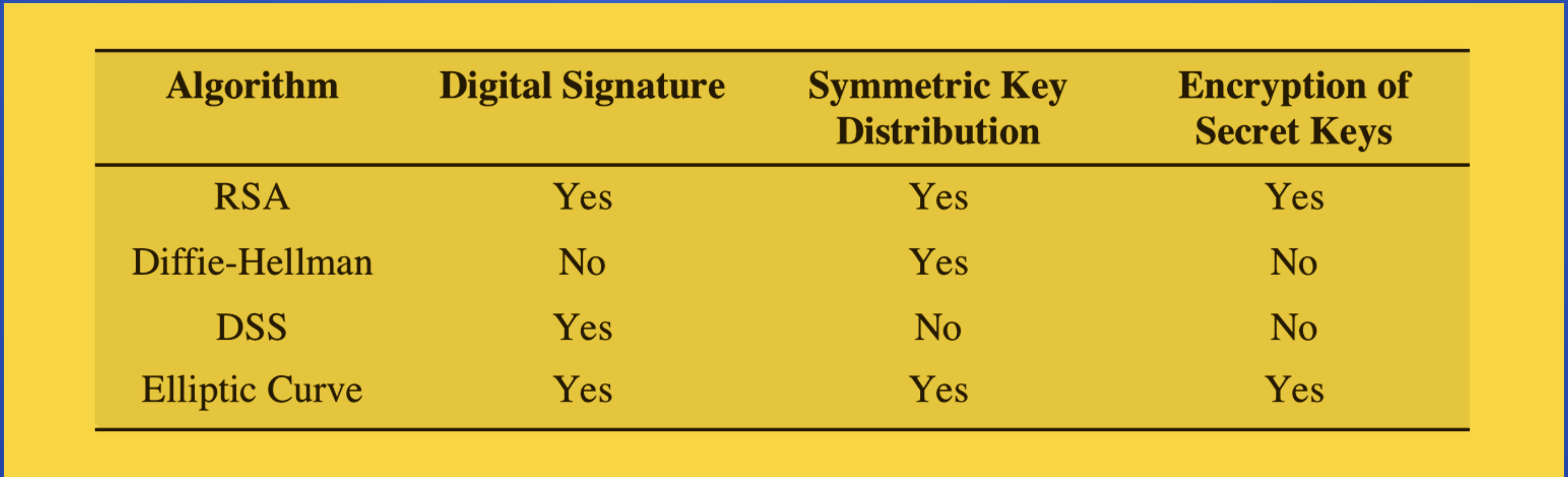
(b) Encryption with private key

Figure 2.7 Public-Key Cryptography

- user encrypts data using his or her own private key
- anyone who knows the corresponding public key will be able to decrypt the message

***directed toward providing authentication

# Table 2.3



| Algorithm | Digital Signature | Symmetric Key Distribution | Encryption of Secret Keys |
|---|---|---|---|
| RSA | Yes | Yes | Yes |
| Diffie-Hellman | No | Yes | No |
| DSS | Yes | No | No |
| Elliptic Curve | Yes | Yes | Yes |

Applications for Public-Key Cryptosystems

# Requirements for Public-Key Cryptosystems

computationally easy to create key pairs

useful if either key can be used for each role

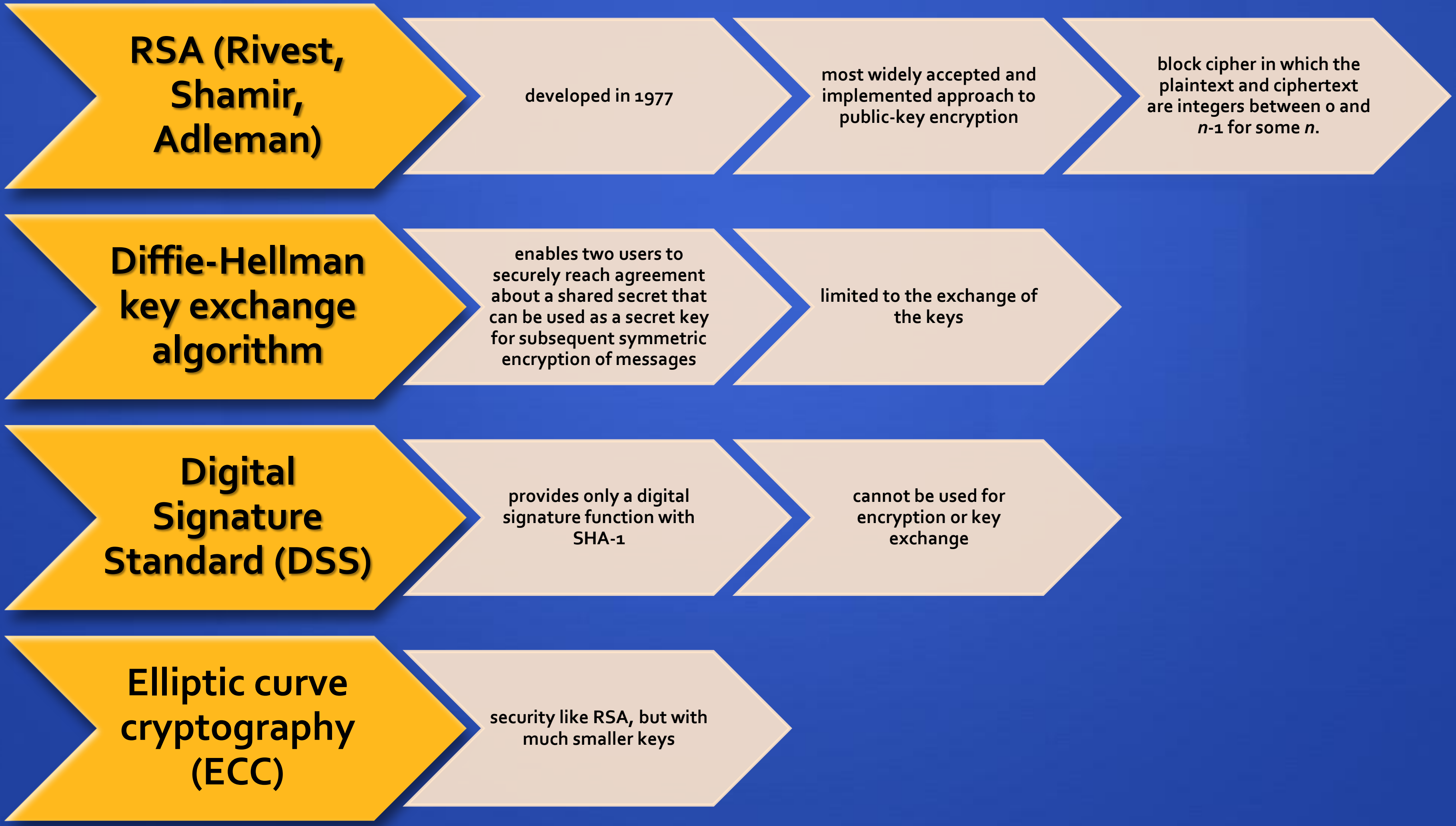
computationally easy for sender knowing public key to encrypt messages

computationally infeasible for opponent to otherwise recover original message

computationally easy for receiver knowing private key to decrypt ciphertext

computationally infeasible for opponent to determine private key from public key

# Asymmetric Encryption Algorithms



**RSA (Rivest, Shamir, Adleman)**

- developed in 1977
- most widely accepted and implemented approach to public-key encryption
- block cipher in which the plaintext and ciphertext are integers between 0 and *n*-1 for some *n*.

**Diffie-Hellman key exchange algorithm**

- enables two users to securely reach agreement about a shared secret that can be used as a secret key for subsequent symmetric encryption of messages
- limited to the exchange of the keys

**Digital Signature Standard (DSS)**

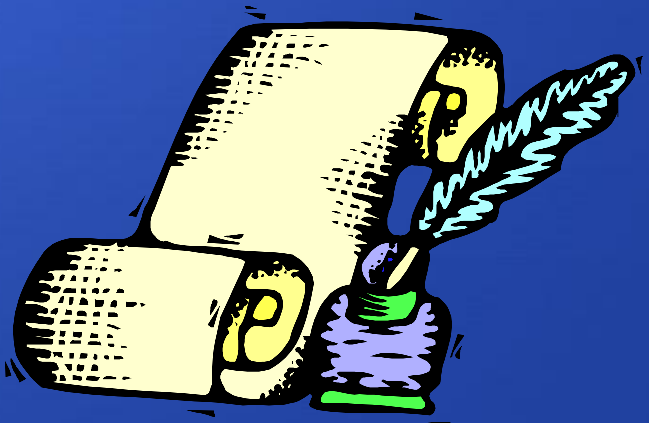
- provides only a digital signature function with SHA-1
- cannot be used for encryption or key exchange

**Elliptic curve cryptography (ECC)**

- security like RSA, but with much smaller keys

# Digital Signatures

- used for authenticating both source and data integrity
- created by encrypting hash code with private key
- does not provide confidentiality
  - even in the case of complete encryption
  - message is safe from alteration but not eavesdropping



# Public Key Certificates

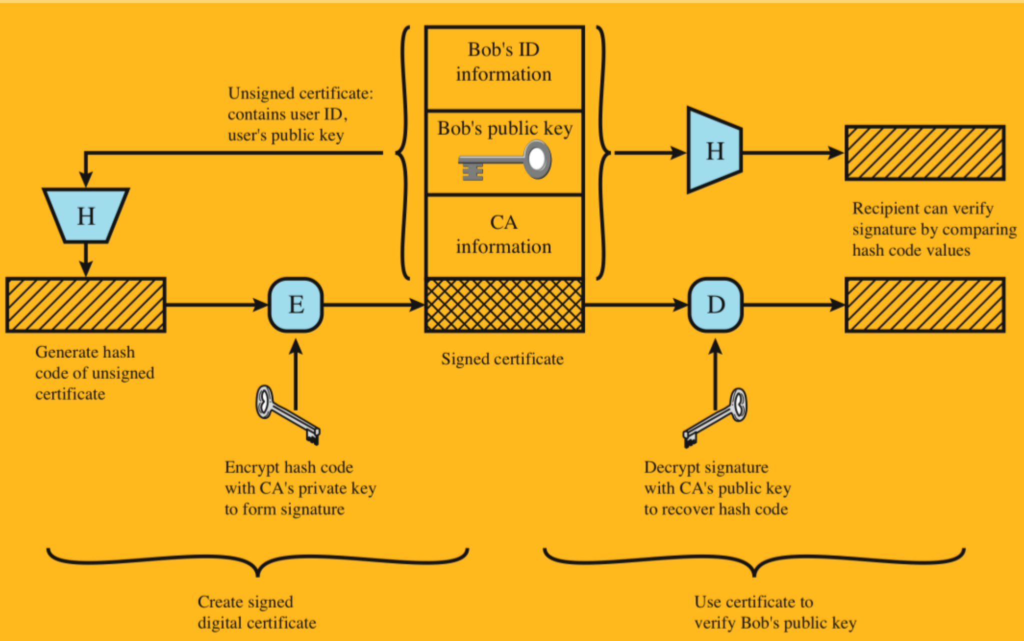
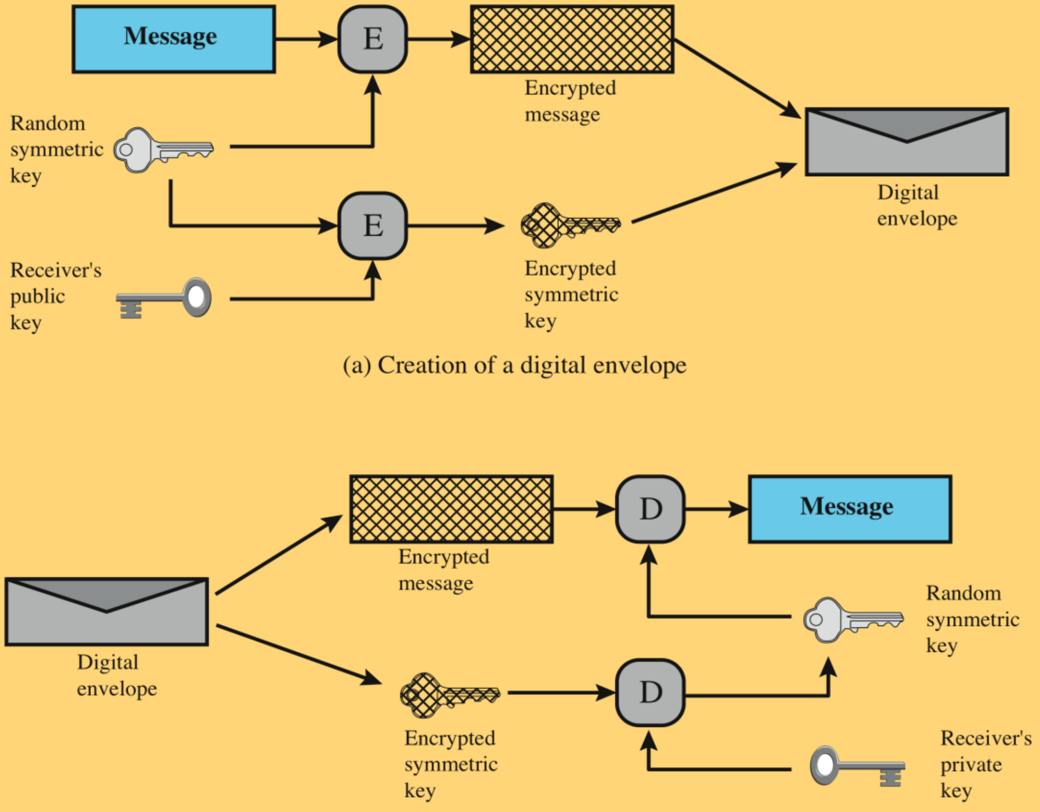

Bob's ID information
Bob's public key
CA information
Unsigned certificate: contains user ID, user's public key
H
H
Recipient can verify signature by comparing hash code values
E
D
Generate hash code of unsigned certificate
Signed certificate
Encrypt hash code with CA's private key to form signature
Decrypt signature with CA's public key to recover hash code
Create signed digital certificate
Use certificate to verify Bob's public key

**Figure 2.8 Public-Key Certificate Use**

# Digital Envelopes

- protects a message without needing to first arrange for sender and receiver to have the same secret key

***equates to the same thing as a sealed envelope containing an unsigned letter



(a) Creation of a digital envelope

(b) Opening a digital envelope

**Figure 2.9 Digital Envelopes**

# Random Numbers

Uses include generation of:

- keys for public-key algorithms
- stream key for symmetric stream cipher
- symmetric key for use as a temporary session key or in creating a digital envelope
- handshaking to prevent replay attacks
- session key

# Random Number Requirements

## Randomness

- criteria:
  - uniform distribution
    - frequency of occurrence of each of the numbers should be approximately the same
  - independence
    - no one value in the sequence can be inferred from the others

## Unpredictability

- each number is statistically independent of other numbers in the sequence
- opponent should not be able to predict future elements of the sequence on the basis of earlier elements

# Random versus Pseudorandom

- cryptographic applications typically make use of algorithmic techniques for random number generation
  - algorithms are deterministic and therefore produce sequences of numbers that are not statistically random
- pseudorandom numbers are:
  - sequences produced that satisfy statistical randomness tests
  - likely to be predictable
- true random number generator (TRNG):
  - uses a nondeterministic source to produce randomness
  - most operate by measuring unpredictable natural processes
    - e.g. radiation, gas discharge, leaky capacitors
  - increasingly provided on modern processors

# Practical Application: Encryption of Stored Data

common to encrypt transmitted data

much less common for stored data

- there is often little protection beyond domain authentication and operating system access controls
- data are archived for indefinite periods
- even though erased, until disk sectors are reused data are recoverable

approaches to encrypt stored data:

- use a commercially available encryption package
- back-end appliance
- library based tape encryption
- background laptop/PC data encryption

# Summary

- symmetric encryption
  - conventional or single-key only type used prior to public-key
  - five parts: plaintext, encryption algorithm, secret key, ciphertext, and decryption algorithm
  - two attacks: cryptanalysis and brute force
  - most commonly used algorithms are block ciphers (DES, triple DES, AES)
- hash functions
  - message authentication
  - creation of digital signatures
- public-key encryption
  - based on mathematical functions
  - asymmetric
  - six ingredients: plaintext, encryption algorithm, public and private key, ciphertext, and decryption algorithm
- digital signatures
  - hash code is encrypted with private key
- digital envelopes
  - protects a message without needing to first arrange for sender and receiver to have the same secret key
- random numbers
  - requirements: randomness and unpredictability
  - validation: uniform distribution, independence
  - pseudorandom numbers